Sorbonne Université - EDITE de Paris

Laboratoire d'Informatique de Sorbonne Université (LIP6) / Quantum Information

Study of Protocols Between Classical Clients and a Quantum Server

Par Léo Colisson

Thèse de Doctorat d'Informatique

Dirigée par Elham Kashefi et Antoine Joux

Présentée et soutenue publiquement le 28 Mars 2022, devant un jury composé de :

- Albrecht Martin,
  University of London (Royaume-Uni), Rapporteur
- Fawzi Omar,
  ENS de Lyon (France), Examinateur
- Jeffery Stacey,
  Centrum Wiskunde & Informatica (Pays-Bas), Rapporteuse
- Joux Antoine,
  CISPA Helmholtz Center for Information Security (Allemagne), Directeur de thèse
- Kashefi Elham,
  Sorbonne Université (France) et University of Edinburgh (Royaume-Uni), Directrice de thèse
- Magniez Frédéric,
  IRIF (France), Examinateur
- Naya-Plasencia María,
  INRIA Paris (France), Examinatrice
- Renner Renato,
  ETH Zürich (Suisse), Examinateur

Toute la splendeur respire là)[1] *dans če tričotaǧe*
(tout en čontrepoint ornemental, dont
l'oriǧine pour moi ne peut
pas faire de doute :
la nature !).

Les furtifs, Alain Damasio

[1] Bien entendu, ceci n'est pas une ode à cette thèse, mais à la beauté du monde et des mathématiques.

Abstract

Given the pace at which quantum hardwares evolve,
Society will surely face a problem to solve.

Imagine a few opulent quantum servers
Admired by the crowd of classical users.

Among them—can you see?—the poor Alice crying:
"This computer is too slow, what a bad timing!"

With a malicious sight, a rich quantum server
Kindly comes to Alice, offering to serve her.

But to the humble Alice, can we guarantee
That the price to pay, won't be her liberty?

Throughout this thesis, carefully, we will ensure
The computation, to the server stays obscure.

To lighten our walkway, we will craft a candle
That classically counterfeits a quantum channel.

To strike the fatal blow against the terror
We require the help of "Learning With Errors".

Sadly we uncover an abrupt disclosure:
No such protocols are composably secure.

In our quest of equity, we got a surprise:
Multiple clients can together socialize

And on quantum states we can prove non-trivial facts
With a single message, keeping the state intact.

Detailed Abstract

Quantum computers are the Holy Grail of many scientists: they promise surprising powers of computation by exploiting the stunning physical properties of infinitesimally small particles. The first quantum computers will certainly be extremely expensive; consequently, their services will (probably) be made available online, similar to popular classical "cloud" providers.

However, a company owning a secret high-value algorithm will undoubtedly try to protect it against a dubious cloud provider: it is thus of utmost importance to provide to the users guarantees on the privacy of their data. Multiple protocols already allow a weakly quantum client to delegate a computation on a remote quantum server by making sure that the server is *blind*, meaning that they cannot learn the input, the output, and the algorithm used by the client. But the client needs to be able to send quantum states to the server, which is a strong requirement as quantum communication is extremely challenging to achieve. Removing this constraint is therefore of paramount importance.

In this thesis, we will see how a purely classical client can use the computational resources of a quantum server, so that the performed computation is never revealed to the server. To this end, we develop the first protocol that can generate on a remote server a quantum state that is only known to the client. This modular tool allows us to classically fake a quantum channel, and can be used in particular to do classical-client blind quantum computing. We also provide constructions to more efficiently craft multi-qubit states.

In our protocol, we need to design a specific cryptographic family of functions having several properties. We propose a construction based on the Learning With Error problem, and provide a careful analysis of the parameters.

We also prove that there is no protocol secure in a generally composable model that can classically fake a quantum channel. A similar impossibility result also applies to the classical-client blind quantum protocols based on the "UBQC" protocol.

In addition to delegated computation, we show that our module also turns out to be useful to perform a task that might seem impossible to achieve at first sight: proving advanced properties on a sent quantum state in a non-interactive and non-destructive way, including when this state is generated collaboratively by several participants. This can be seen as a quantum analogue of classical Non-Interactive Zero-Knowledge proofs. The set of properties that are verifiable is highly non trivial—we can prove any property on the set of entangled qubits—and this could have numerous applications, for instance to filter participants in a protocol without revealing their identity.

Acknowledgement

In these past years, I had the chance to meet many extremely generous and talented people, from whom I learned so much. Given my endless enthusiasm for discussions, debates and questions, I am grateful to many more people than can reasonably fit in this acknowledgement. I therefore apologize in advance to those I could not mention here.

First and foremost, I would like to express my deepest gratitude to my supervisors, Elham Kashefi and Antoine Joux, who guided me throughout this PhD. Elham, words cannot express enough how grateful I am for the trust you have placed in me and for everything I have learned from you. Your everlasting optimism and happiness has been an inexhaustible source of motivation and your endless curiosity will forever be a model for me. I wish I had one percent of your talent to bring life to a group. Antoine, your extremely efficient, methodical and Cartesian reasoning is a constant source of inspiration for me. I particularly enjoyed our discussions, which have all been extremely enlightening.

I also warmly thank my co-authors, Alex(andru), Atul, Christian, Dominik, Elham, Frédéric and Petros, not only for the numerous insightful and lively discussions we had (I loved them!), but also for these deadline nights (there is an abstract that I will remember forever...). I must also express my gratitude to Air France, who offered to some of us a whole night ~~blocked at the airport~~ dedicated to science. A special thanks to Alex: you have been a great support to me from the very beginning, and your profound kindness is matched only by your talent in extended-table tennis.

I want to sincerely thank my reporters, Martin Albrecht and Stacey Jeffery, who kindly accepted to read my manuscript and provided very pertinent comments: given my contradictory passion for conciseness and explanation and the fact that I have clearly failed to meet at least one of these goals in this thesis, I am forever indebted to you. I also express my profound gratitude to the rest of the jury, Frédéric Magniez, María Naya-Plasencia, Omar Fawzi and Renato Renner: I am truly honored to present my work to such leading experts, whose expertise ranges from classical cryptography to quantum information theory. I also thank Iordanis Kerenidis and María Naya-Plasencia for having accepted to serve on the jury of my monitoring comity.

I extend my gratitude to Céline Chevalier and Omar Fawzi for being such wonderful internship supervisors. Omar, thank you so much for helping me to take my first steps

into the world of quantum computing and information theory: my life changed the day I discovered SDPs and I cannot wait to further explore the realm of information theory. Céline, I could not dream of a better place to explore classical and post-quantum cryptography, and of course composable security. During my journey, I got the chance to get many insightful discussions or advises, thanks in particular to Alain Passelègue, Alex Grilo, (Alex)andru Gheorghiu, Chris Peikert, Daniel Jost, Dominique Unruh, Florian Bourse, Geoffroy Couteau, John van de Wetering, Miriam Backens, Niel De Beaudrap, Romain Gay, Thomas Vidick, Thomas Zacharias, Vedran Dunjko, Vincent Danos and Yiannis Tselekounis. I am also grateful to all the teachers that transmitted me their passion: Samuel Bonnet, Clotilde Catté, Alain Thibaud, Michel Lepesant, Franz Ridde (thank you so much for the now famous "Pas de dessins, pas de points": you would love the ZX-calculus), Hubert Comon-Lundh, Jean Goubault-Larrecq, Gilles Dowek, David Pointcheval... And thanks to everybody that helped me online (LaTeX would have killed me otherwise).

I also want to express my gratitude to Khamsa Habouchi, Frédéric Grosshans and Nicolas Treps for your support in the QICS, as well as to Clément, Ganaël, Raja and Robert for the great time we spent while making OptiQraft! À toi de jouer maintenant Clément ;-)

Many thanks to the fantastic people I met in Edinburgh and in the QI team, with whom I shared countless fascinating discussions, breathtaking retreats, and priceless[2] beers (or bears). In particular, thanks Damian, Eleni, Elham, Frédéric, Alex, Shane, and Harold for the amazing dynamism and joy you create in the group (I couldn't dream of a better lab to do my thesis), Alex(andru) and (Alex)Andru (I tend to think that all Romanians are called "Alexandru" now, *Ce faci, șerifule?*), Clément (the anti-capitalist businessman who solved the issues I had with praline), Nathan (for the Karaokes, the proofreadings, and the "easy" hikes (say "Hi!" to the Canadian House)), Raja (for your big heart, and the crazy nights animated with your guitar), Robert (for the time I lost thanks to your comment on the style of my ZX diagrams), Luka (for your so characteristic laugh), PE (for the "tarte tatin" and the proof reading), Dominik (*Du weißt, was ich bin.*), Mina and Mahshid (*Farsim roubé!*), Andrea (because I like pineapple pizzas), Victor (*Soy naranja. Me gusta.*), Atul (for your "talent" at Pictionary), Ulysse (now I'm in the "cool guys office"), Federico (remember, one can open a bottle with a sheet of paper), Ellen (for the great retreat), Brian and Daniel (so british with the hand made coffee), Niraj and Rawad (with your communicative joy), Armando (and the Audi), Paul Hilaire (sorry for my cough) and the second Paul H., Yao, Anu, Mathieu, Tom, Matteo, Simon, Alisa, Francesco, Natansh, Cyril, Kaushik, Ieva, Rhea, Shraddha, Sasha, Theodoros, Shouvik, Yoann, Laura, Uta, Valentina, Verena, Majhid, Luis, Marine, Adriano, Adrien, Constantin, Slimane, Damien, Dimitrios, George, Ivan, James, Jonas, Kim, Pascal, Paolo, without forgetting the great "greek team" of Edinburgh with Aggelos, Thomas, Yiannis, Giorgos, Myrto, Aikaterini, Nikolaos and Orfeas: thanks you all for your friendship, the Friday pub and the extensive debates.

[2] Ok, this expression may not be appropriate for *Bières Cultes*.

I also immensely thank the marvelous crypto team at Ulm: Céline, David, Phong, Brice, Hoeteck, Michel and the open-space members Chloé, Mélissa, Balthazar, Antoine, Jérémy, Romain, Geoffroy, Michele (x2), Huy, Louiza, Anca, Ehsan, Azam and Lénaick. I had great time eating pizzas (chosen by the Italians in person) and playing thrilling board games with you all (thanks Antoine for the organization). Anyway, we all know who is the traitor. My only regret is that I never knew the shrimps.

And of course thanks to Alice and the malicious Bob, who accompanied me everyday during these last three years. And I'm not sure if I should thank the COVID 19 for starting in the middle of my PhD, as it blocked all my opportunities to visit other groups and go abroad for conferences... But on the positive side, let say that it helped our community to reduce our carbon footprint by favoring online events.

Je ne pourrais terminer ces remerciements sans un mot pour mes formidables amis à l'extérieur du labo, que je remercie infiniment. En particulier, merci Quentin pour ces inoubliables moments passés à crapahuter dans la forêt, pour nos discussions passionnées, et pour m'avoir fait découvrir un breuvage pour le moins... explosif. À Théo, mon mémorable ex-voisin tout aussi théâtral que bienveillant, et à notre livre toujours en cours d'écriture. À Thomas (et sa petite famille) qui a été à mes côtés pendant 7 ans et qui n'a eu de cesse de raccourcir sa coiffure et de prouver sa (parfois trop) grande générosité. À Magali (j'omettrai tes nombreux surnoms pour le bien de tous), pour ton brin de folie contagieux, tes îles flottantes, le perloscope et les innombrables trajets en Bordemobile. À Julien, notre mélomane favori, aux opéras et tes DVDs (j'en ai d'ailleurs quelques uns à te rendre). À la TS7 et à nos fameux débats du vendredi, avec Sylvain (la petite puce... accrétion !), Victor (et son VTT), Rémi, William, Renée, Yohann et toute la bande. Aux pianistes de St Do, et à Sarkis et Phillipe mes anciens professeurs passionnés de piano et saxophone.

À notre incroyable groupe HX3 : Clément (ce projet guirlande avec Alexis sera certainement la réalisation la plus ambitieuse de notre vie), Chloé (et sa joie communicative, notre amour de la vaisselle et I'm blue 🎵), Joachim (notre géant aux cœur tendre, mais aux mains usées par la batterie), Milan (et notre voyage en Europe), Benoit (faut vraiment que tu me montres les beaux coins vers Grenoble), Mathieu (Triboulet RPZ), Gauthier (et ses cannelloni à tomber), Isabelle (quel groupe de khôlle, sisi je connais mes DL !), Eli (à quand la présidentielle ?), Solène (notre poétesse), Audrey (poétesse, j'ai le droit ?), Turian et Paul-Erwan (fatigué ?), Agathe (et Kim Jong Un), nos JB (toujours un pour rattraper l'autre), Chef (et sa (réelle) modestie), sans oublier Arthur (partit trop tôt).

À Cachan, j'ai eu la chance de rencontrer nombre de personnes incroyables, tant par leurs talents que leur générosité. Un grand merci à Zobi et Zobinette (à quand le prochain escape game ?) et à tous les membres de la Kataclist (ainsi que l'édition 2.0) : pour ne pas exploser la taille de ces remerciements, je ne citerai que notre bien aimé président ~~chopeman~~ Bourbon (d'une rare humanité et ouverture d'esprit, meilleur DJ de l'ENS, et toujours là pour vous redonner le sourire), mais vous êtes tous fantastiques ! Aux PHYTEM (oui, je vous ai quitté, mais mon cœur est resté), un immense merci à Routhier (et nos haricots dans une coupelle à raclette), Pieter Walch (le maître incontesté des éléments), Bellezza (la radio, ça passe jusqu'à où ? Pauvre François...), Corentin, Bastien

et Souquet (pour leur chorées) et les membres du [Car]Naval (Souquet les artimuses !). Merci aux infos de m'avoir ensuite accueilli chaleureusement : Émilie et Aliaume (Salsa !), Damien, Gaetan, Rebecca, Laurent, Guinness et Shana. En parlant de danse, un grand merci à tous les membres du club Salsa (et à Francis), ça me faisait (et fait) tellement de bien de vous retrouver, en particulier Céline (merci dragonne, ce fût un plaisir de réaliser en ta compagnie nos merveilleuses chorées), mon cher Lev-Arcady (qui guide ? et qui critique mes librairies !), Émilie (la relève est assurée !), Jadoul (oui, tu es beau), Raphaël (et le déménagement à distance), Claire (ou Kloppy), Mélisenda (et Lilo et Stitch), sans oublier nos renforts tant appréciés d'Agro/du Parc, en particulier Marie (merci pour ta joie de vivre, pour mon placement original dans l'arbre généalogique et les cueillettes) et Quentin (et son aide pour équilibrer le ratio de danseuses).

Merci aussi à Ethan, ce fût un plaisir de t'accompagner pendant ces 4 années, félicitations pour ta réussite. Je te souhaite une excellente continuation. Merci aux Dijonnais pour leur bonne humeur musicale et à Augustin pour ce projet "cinématographique" plein de rebondissements! Un grand merci à la famille d'Alexia, pour leur support (entre autre sous la forme de bocaux) et leur merveilleux jardin.

Je souhaiterais remercier de tout mon cœur ma famille pour leur soutien indéfectible : mes parents pour toutes les valeurs qu'ils m'ont transmises et sans qui je n'en serais pas là aujourd'hui, Carl (notre voyageur) et Léna (Nana !) pour les heures passées à s'amuser (ou à se chamailler), mes grands-parents pour leur dynamisme et leur sagesse, mes oncles, tantes, parrain et marraine pour leur bonne humeur et leur bienveillance, et bien sûr mes nombreux cousins et cousines pour les innombrables souvenirs de vacances (« pas de relations sentimentales ! »).

Enfin, les mots ne suffiront pas pour remercier Alexia de sa présence à mes côtés pendant toutes ces années et de son soutien inébranlable (y compris avant les deadlines avec mes briques ^^). Merci pour tout ce que tu as déjà fait pour moi, pour ton écoute (tu connais « ma copine » ?) et pour ton amour. Et si j'ai adoré tous nos voyages, j'ai hâte de voir ce que le nôtre nous réserve !

How To Read This Thesis

I wanted this thesis to be as much self-contained as possible, so that it can be understandable (and verifiable) by readers having either a quantum or a classical background. At the same time, I wanted to allow the reader to quickly grasp the fundamental ideas behind our methods without being overwhelmed by details. As a result, each chapter starts with a short description of our approach, hopefully as self-contained as possible. We encourage the hurried reader to directly move to these overviews. The dependency relation between the chapters is pictured in Figure 1.1.

Do I need to learn the ZX-calculus? I recently discovered the joy of diagrammatic reasoning, with notably the ZX-calculus. Despite its frightening name, it allows to simplify significantly the computations and provides a greater intuition of what is happening. Unfortunately, it is not yet used extensively across the quantum community. Nevertheless, I chose to replace some dirty linear algebra computations with nicer diagrammatic computations: it turns out to be particularly profitable, notably to explain both the correctness and security of the UBQC protocol and to simplify our proofs. However, the reader really reluctant to learn the ZX-calculus does not fundamentally need it. All the proofs can be rewritten using standard linear algebra.

Do I need to understand UBQC? The UBQC protocol is used to achieve blind delegated quantum computing when a quantum channel is available. One of our main contribution is a protocol—called QFactory—that fakes a quantum channel using a purely classical channel: QFactory is then used modularly to replace the quantum channel in UBQC. Therefore, it is not necessary to understand UBQC to understand QFactory. But we also prove multiple results regarding classical versions of UBQC: in that case UBQC will be required as picture in Figure 1.1.

Table of Contents

Notations and Acronyms

Notations. We summarize in this table the notations used across the thesis.

Notation	Description
$\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$	Set of (respectively) reals, positive reals and non-negative reals.
$\mathbb{C}$, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Z}_q$	Set of (respectively) complex numbers, non-negative integers, integers and the quotient ring of integers modulo q.
$A \times B$, $A \otimes B$	Cartesian and Tensor product (respectively).
$\{k_i\}_{i \in \mathcal{K}}$ or $\{k_i\}$	Family indexed by elements i in $\mathcal{K}$.
$\{A_i \mid i \in X\}$	Set of elements A_i such that $i \in X$.
$[n]$	Set $\{1, \ldots, n\}$.
$\mathbb{Z}_{\frac{\pi}{4}}$	Set of angles $\{0, \frac{\pi}{4}, \ldots, 7\frac{\pi}{4}\}$.
I^c	Complementary of the set I: $I^c = \{x \mid x \notin I\}$.
$\mathcal{U}(X)$	Uniform distribution over X.
$x \leftarrow \chi$	x is sampled according to the distribution χ.
$x \xleftarrow{\$} X$	$x \in X$ is sampled uniformly at random in the set X.
$x \coloneqq v$	The notation x is defined as v. We also use $x \coloneqq v$ to say that x takes the value v.
$\lvert a \rvert$	Size of a: if a is a complex number, it is its norm, if a is a set, it is the number of elements in the set and if a is a bit string, it is its length.

Notation	Description
$a \propto b$	a is proportional to b.
$\delta_{i,j}$	Kronecker delta: $\delta_{i,j} = 1$ if and only if $i = j$, otherwise $\delta_{i,j} = 0$.
$\mathbf{u}\|\mathbf{v}$	Concatenation of the bit strings $\mathbf{u}$ and $\mathbf{v}$.
$\mathbf{v}_i$, $\mathbf{v}[i]$	ith element of the bit string $\mathbf{v}$. $\mathbf{v}[i]$ is used to avoid confusions when the name of a string may already contain a subscript.
$\mathbf{v}_{1:n}$	Sub-string of $\mathbf{v}$ composed of $\mathbf{v}_1\|\ldots\|\mathbf{v}_n$.
$\mathcal{H}$	Finite dimensional Hilbert space.
$\mathcal{H}_n$	Hilbert space of dimension n.
$\mathscr{L}(\mathcal{H})$	Linear operators from $\mathcal{H}$ to $\mathcal{H}$.
$\mathscr{L}_\circ(\mathcal{H})$	Linear operators from $\mathcal{H}$ to $\mathcal{H}$ with trace 1, corresponding to (normalized) quantum states represented as density matrices.
$[\rho]$	Classical description of the quantum state ρ.
$\overline{a}$	Complex conjugate of a (a may be a complex number or matrix).
$\mathbf{A}_{i,j}$	Entry at the ith line and jth column of the matrix $\mathbf{A}$.
$\mathbf{I}$, $\mathbf{I}_n$	Identity matrix (of dimension n).
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$.
$\mathbf{A}^\dagger$	Hermitian adjoint of the matrix $\mathbf{A}$, equal to its conjugate transpose: $\mathbf{A}^\dagger = \overline{\mathbf{A}}^T$.
$\|\psi\rangle, \langle\psi\|, \langle\psi\|\phi\rangle$	Braket notation: $\|\psi\rangle$ is a vector, $\langle\psi\| \coloneqq \|\psi\rangle^\dagger$ and $\langle\psi\|\phi\rangle \coloneqq \langle\psi\|\,\|\phi\rangle$. More details in Section 2.1.
$\|\mathbf{x}\|_2$	Euclidean norm of $\mathbf{x}$: $\|\mathbf{x}\|_2 = \sqrt{\overline{\mathbf{x}}^T\mathbf{x}}$
$a \wedge b$	Logical "and" operation between bits a and b.
$a \oplus b$	Logical "XOR" operation (addition modulo 2) between bits a and b. If a and b are bit strings, it corresponds to the bitwise XOR operation.
$\langle b, x\rangle$	If $(b, x) \in \{0,1\}^n$, $\langle b, x\rangle = \oplus_i b_i x_i$.

Acronyms. We summarize in this table the acronyms and abbreviations used across the thesis.

Acronyms	Description
BB84	Quantum states $\{\|0\rangle, \|1\rangle, \|+\rangle, \|-\rangle\}$ defined in [BB84].
CC	Constructive Cryptography (Section 6.2)
CPTP map	Completely Positive Trace-Preserving map (Section 2.1.3)
IND-CPA/...	Security games. You can click on the name to go to the definition.
GHZ	Greenberger–Horne–Zeilinger quantum states: $\frac{1}{\sqrt{2}}(\|0\ldots0\rangle + \|1\ldots1\rangle)$.
iff	If and only if.
LWE	Learning With Errors (Section 5.2).
MBQC	Measurement Based Quantum Computing (Section 2.3.6).
MPC	Multiparty computing (Section 7.2.2).
POVM	Positive Operator-Valued Measure (Section 2.1.2).
PPT/QPT	Probabilistic Polynomial Time and Quantum Polynomial Time.
RHS/LHS	Right-Hand Side and Left-Hand Side of an equation.
RSP	Remote State Preparation (Chapter 4).
$\mathsf{RSP}_{\mathsf{CC}}$	Classical-client Remote State Preparation (Chapter 4).
UBQC	Universal Blind Quantum Computing (Section 2.3.7).
$\mathsf{UBQC}_{\mathsf{CC}}$	Classical-client Universal Blind Quantum Computing (Section 2.3.7).
VBQC	Verifiable Blind Quantum Computing [FK17].
ZK/NIZK/NIZKoQS	Zero-Knowledge, Non-Interactive Zero-Knowledge and Non-Interactive Zero-Knowledge on Quantum States (Chapter 7).

List of Figures

List of Protocols

List of Games

Chapter 1

Introduction

"We never rob. We just sort of borrow a bit from those who can afford it."

— Disney, *Robin Hood*

Is Nature fair: benevolent towards the weak and an advocate of equity? While it is certainly ambitious to characterize Nature's philanthropy, we propose in this thesis to approach this issue indirectly by means of quantum cryptography:

Can weak (classical) clients compete against powerful (quantum) servers?
More specifically, can such weak clients use the computational resources of powerful servers without revealing them any information?

Classical Cryptography. Cryptology—the science of secret—is a surprising but powerful tool to distill the essence of our universe. In 1995, Impagliazzo defined five worlds [Imp95a]: Algorithmica, Heuristica, Pessiland, Minicrypt and Cryptomania, later extended with Obfustopia. Each of these worlds is characterized by the difficulty of specific problems.

- In the first three worlds, (classical) cryptography is very limited as basically no problem is sufficiently "hard"[1]. In particular, two parties can communicate securely—in the sense that no eavesdropper can decrypt an exchanged message—only if they can communicate beforehand via a completely trusted channel a large quantity

[1]The exact definition of "hard" is not important for now and will be formalized later. But informally, a problem is hard if no polynomially bounded computer can solve this task efficiently (more precisely in polynomial time, whatever that means).

of truly random information (scaling with the size of the message). History has shown that it is a challenging task to achieve [Cen, SE02].

- In the Minicrypt world, *one-way* functions exist (i.e. functions which can be evaluated efficiently but are sufficiently hard to invert): cryptography becomes interesting. For instance, symmetric cryptography is possible: as before parties wanting to communicate must agree on a secrete random key over a trusted channel, but the amount of exchanged information can be much smaller [HIL+88, LR85, GGM86]. More involved cryptographic applications are also imaginable: among others, it is possible to prove a statement without revealing anything beyond the fact that this statement is true (this is known as Zero-Knowledge [GMR85, BM88, GMW91]) and it is possible to digitally sign a document [NY89].
- In the Cryptomania world, *trapdoor* one-way functions exist (i.e. one-way functions which are easy to invert when the trapdoor in known). In this world, even more enthralling cryptographic tasks are possible as one can securely communicate without agreeing first on a secret key (better known as public-key cryptography [DH76, RSA78]). It is also possible to do much more advanced tasks, like multiparty computations: one can compute any joint function between multiple parties such that the input of each party is never revealed to others [GMW87].
- In the Obfustopia world, programs can also be *obfuscated*, meaning that it is possible to modify the code of a program in such a way that the code does not reveal any valuable information about the program besides what would be learnable by evaluating the program.

One of the most fundamental question in computer science is certainly to prove in which world we are actually living. In particular, answering this question would allow us to solve the Holy Grail of computer scientists, which is to determine if $\mathsf{P} = \mathsf{NP}$[2].

As far as we know, it seems that we live in Cryptomania: while we do not know how to obfuscate a program, we know constructions to obtain trapdoor functions—for instance based on the hardness of the Learning With Error problem [Reg05]—such that no known algorithm can invert these functions efficiently without the trapdoor. As a consequence, we can use public-key cryptography: we even use it every day. Without public-key cryptography, it would be impossible to securely connect to a bank website, to buy something on the internet or to send an encrypted message to a friend; only people having enough power to distribute secret keys—say, by physically sending an agent with

[2] P is the class of problems that are classically efficiently solvable and NP is the class of problems for which it is easy to verify whether a solution is correct. If $\mathsf{P} = \mathsf{NP}$, it means that we live in Algorithmica.

a suitcase full of random keys—would be able to securely communicate. So far, Nature seems to care about equity.

Quantum Computing. However, there exists a different approach to characterize the power of an individual beyond its ability to share keys: computational power. Nowadays, everyone has roughly the same computational power (most have access to a "classical" computer), notably though, a fundamentally new kind of device is slowly emerging: *quantum computers*. This idea started to appear around 1980 [Fey82, Ben80, Man80], pushed by the extraordinary discovery of quantum mechanics at the beginning of the 20th century [Pla01, Ein05, MR82]. Quantum mechanics describes the stunning physical properties of infinitesimally small particles. At that scale, physics behaves strangely: observing a quantum state disturbs it, making it impossible to copy quantum states and it is possible to manipulate an exponential amount of data using only a few operations.

By exploiting the unusual properties of quantum mechanics, quantum computers could outperform their classical counterpart. One of the most famous application of quantum computers is the ability to efficiently factorize very large numbers, otherwise known as Shor's algorithm [Sho94]. Classically, there are no known efficient algorithms which can solve this problem: the hardness of the factorization is a necessity for the security of the RSA public-key encryption scheme [RSA78], which is one of the pillars of contemporary cryptography. Thus, quantum computers are a threat to the security of Internet. Quantum algorithms can also be used to quadratically speed-up search using the well-known Grover's algorithm [Gro96]. Furthermore, new thrilling algorithms are developed on a daily basis [Mon16]. As a consequence, people having access to a quantum computer could have a significant advantage over those having only access to a classical computer.

Delegated Computing. However, building a full-fledged quantum computer is an extremely challenging task (which is yet to be achieved). Nonetheless, much progress has been done since the first experiments [MMK+95, JM98]. The first embryos of quantum computers are now starting to emerge, claiming to solve specific problems that are unsolvable on classical computers [AAB+19, ZWD+20].

The first full-fledged quantum computers are likely to be extremely expensive and will certainly be made available to the public over the Internet: multiple quantum companies already share their device "in the cloud", like [IBM, Ion, Rig, Hon, Xan, Goo]. Similarly, classical cloud services such as Amazon AWS, Dropbox, or Google Cloud are tremendously popular: countless users are interested in using remote servers to store

data, or to delegate massive computations on powerful servers. Chances are high that History will repeat for quantum devices.

Just as with classical delegated computing, it is of utmost importance to provide to the users guarantees on the privacy of their data. A company owning a secret high-value quantum algorithm will undoubtedly try to protect it against a dubious cloud provider, similarly, a medical lab surely needs to preserve the secrecy of the database of its patients on account of medical confidentiality.

A protocol known as *Universal Blind Quantum Computing* (UBQC) [BFK09, DFP+14] already allows a user to delegate a computation on a remote quantum server by making sure that the server is *blind*, meaning that they cannot learn the input, the output and the algorithm used by the client. Other protocols have also been developed, trying to reduce the interactivity, the communication, the quantum capabilities of the client or to provide verification [FK17, ABE08, BJ15, DSS16, FBS+14, Bro15, Lia15] (see also the following reviews [Fit17, GKK19]).

Unfortunately, in all these protocols the client needs to have some small quantum capabilities and to share a quantum channel (i.e. a channel in which it is possible to transmit quantum states) with the server. This is a strong requirement, not only because building a quantum internet is a very daunting task[3], but also because some technologies used to build quantum computers are not easily integrated with a photonic quantum network. It is therefore crucial to find tools to get rid of quantum channels, which naturally raises the following questions:

Can a purely classical client perform blind quantum computations on a remote quantum server?

Is it possible to fake quantum channels using purely classical communication?

And more generally, what are the achievable and unachievable protocols between a classical client and a quantum server?

In her breakthrough work [Mah18a], Mahadev demonstrated that blind quantum computing can be accomplished with a purely classical client. This paper was the starting point of numerous works, leading to (sometimes surprisingly related) applications like tests of quantumness [BCM+18, MV21, HG21, BKV+20], device-independent quantum key distribution [MDC+21], verifiable computation [Mah18b] or Remote State Preparation [GV19] (we will see that our independent work lies in this last category). [Bra18]

[3]Typically, quantum states are composed of very few photons that can easily be lost due to imperfections on the transmission line. Moreover, quantum repeaters are hard to produce due to the fact that most photonic gates are intrinsically non-deterministic: as a result quantum communications over long distances are challenging.

also reduced the security assumptions required in [Mah18a] in order to rely on the more standard hardness assumption of LWE with polynomial noise ratio. Note that another work [MDM+17] tried to get rid of quantum communication before [Mah18a], however, due to the design of this protocol, the server does learn some information about the computation and the method for translating arbitrary circuits into this framework is not clear. Another work [Zha21] also provides a protocol which is useful to obtain blind quantum computing. This protocol is based on the Random Oracle Model (ROM) and the client do need to have some quantum capabilities. However, the quantum communication is "succinct" in the sense that the number of sent qubits does not depend on the length of the computation.

Contributions

Chapter 4: RSP and Classical-Client Blind Quantum Computing. In Chapters 4 and 5[4], we present a complementary and independent[5] approach to obtain classical-client blind quantum computing [CCK+18, CCK+19, CGK21]. Compared to Mahadev's monolithic protocol, we build a more modular primitive called QFactory: QFactory is the *first classical-client Remote State Preparation* (RSP) protocol, meaning that it is able to fake a quantum channel using only classical communication. More precisely, at the end of an honest run of QFactory, the server ends up with an unknown quantum state that can only be described by the classical client. All of this happens while the client and the server communicate purely classically.

It is then possible to use this protocol as a sub-routine inside other protocols to replace the quantum interaction. Notably, we can combine QFactory with the UBQC protocol to obtain a protocol achieving classical-client blind quantum computing. In Section 4.8, we provide a detailed comparison of our approach with related works.

This modular approach is particularly interesting: classical-client RSP protocols could potentially be used to replace quantum interactions in a large number of protocols (notably when the sender of the quantum state always knows its classical description). Of course, a different and potentially non-trivial security proof must be written for each application, and our QFactory protocol surely needs to be adapted depending on the situation. In particular, it is not yet clear if QFactory can be used and/or adapted to fit

[4] Chapter 4 defines our protocols while Chapter 5 describes our cryptographic constructions.

[5] We started to develop our protocol during my Master 1 internship [Col17], when Mahadev's result was not yet online. As described in Chapter 4, back that time we had a first working protocol but no proof of security nor any explicit construction for the cryptographic family required in our protocol.

the requirements of a verifiable and blind quantum computation. In fact, the question of the feasibility of a superpolynomially secure classical-client verifiable and blind quantum computing is still open since [GV19] is polynomially secure.

Nevertheless, the list of potential application is huge, including:

- blind and/or verifiable quantum computing as already mentioned,
- but also quantum secure multiparty (QSMPC) computing [KP17, KMW17, KW17, KKM+21] (note that the protocols [DNS12, CGS02, DGJ+20, LRW20, ACC+21] also achieve QSMPC, but since the clients can forward states for which they do not always know the classical description, in these protocols there is no clear way to replace quantum communication with a classical-client RSP protocol),
- position verification [KMS11, BFS+13, CGM+09, Unr14, LLQ21] (see also the review in [Chr21]),
- or quantum homomorphic encryption[6] [BJ15, DSS16].

Using RSP, all these protocols (and certainly more) could get a chance to be implementable using purely classical clients.

Chapter 5: Cryptographic construction. Our QFactory protocol requires the existence of a cryptographic primitive having some special properties. In a nutshell, our family must notably be 2-to-1 and efficiently invertible with a trapdoor (a few more properties are also required). In order to realize this construction, we need to rely on the hardness of the famous *Learning With Errors* (LWE) problem [Reg05]. For now, no known algorithm, quantum or classical, can efficiently solve it [Pei16]. As a result, LWE now stands as the major post-quantum candidate to replace RSA and elliptic curves: these two primitives are the keystones of modern cryptography, but are unfortunately vulnerable to quantum adversaries. The LWE problem can be used to build many cryptographic primitives, for instance to obtain public-key encryption [Reg05, GPV08, Pei09], digital signatures [GPV08, CHK+10, LM08], (hierarchical) identity-based encryption [GPV08, CHK+10] or (fully) homomorphic encryption [BV14] (first achieved with different assumptions in the breakthrough work of [Gen09]).

We can characterize more precisely the security of an LWE instance using a parameter known as the *modulus to noise ratio*, or simply noise ratio. If this noise ratio is polynomial, then the LWE problem is considered as secure. If this noise ratio is exponential, then the LWE problem is easy (and therefore not usable in cryptographic applications). However, if the noise ratio is superpolynomial, but not exponential, then we also do not know

[6] Since RSP protocols typically require at least one round of communication, the RSP part may be considered as a setup phase to preserve non-interactivity during the protocol.

any algorithm to solve the LWE problem in polynomial time [Sch87]. In spite of this, using LWE with polynomial noise ratio is considered as more standard than using LWE with superpolynomial noise ratio, but the superpolynomial assumption is sometimes required [BGG+14].

In Chapter 5, we describe a construction to obtain a (nearly) 2-to-1 and trapdoor family of functions, fulfilling all the requirements needed in our construction. Unfortunately, we can only obtain a family approximately 2-to-1: the quality of the approximation depends on the choice of the hardness assumption. If we allow ourselves to use LWE with a superpolynomial noise ratio, we can obtain a function which is 2-to-1 for an overwhelming fraction of inputs. However, when considering LWE with a polynomial noise ratio, we obtain a function which is 2-to-1 for a fraction of inputs converging polynomially fast to 1. As a result, we propose in Chapter 4 two variants of the QFactory protocol depending on the chosen hardness assumption[7].

Chapter 6: Impossibility results. As explained previously, if QFactory is used internally in an existing protocol to replace a quantum channel with a classical channel, the security proof of the new composite protocol must be reassessed. This can be easily explained as QFactory, along with most of the other protocols cited above—except for [GV19], but it has other problems like polynomial security as discussed in Remark 6.3.11—are proven secure in a so-called "game-based" security model which does not guarantee security under composition. But other security frameworks allow general composability [Can01, MR11, Unr10, BPW03], leading to the following natural question:

Is it possible to prove the security of a classical-client RSP protocol in a general composable framework? If not, can we prove the composable security of a classical-client blind quantum computing protocol?

Note that when considering only statistical security (not even mentioning composable security), we already know that it is highly improbable that a secure classical-client blind quantum computing protocol exists. In [MK19], the authors showed a negative result by presenting a *scheme-dependent* impossibility proof. This was further improved in [ACG+19] which showed that such a statistically secure classical blind quantum computing protocol would have implications in computational complexity theory.

But none of the aforementioned works consider impossibility for *computational* security, and for classical-client RSP protocols. Of course, given the previously mentioned positive

[7] Note however that when we rely on LWE with a polynomial noise ratio, we need to use an additional conjecture.

results on classical-client blind quantum computing and RSP, such a generic impossibility result is unlikely to exist (unless the hardness assumption on LWE collapses).

However, we show in Chapter 6—based on our result in [BCC+20]—that there exist *no* classical-client RSP protocols that is secure in a general composable framework (we focus on the Constructive Cryptography security model [MR11]). Our result is very generic as our definition of RSP resources is very broad and even includes potentially noisy RSP resources.

Secondly, we also show that the UBQC protocol cannot produce a secure composable protocol if the quantum interaction is replaced with a (correct) RSP protocol.

Chapter 7, Section 7.3: Zero-Knowledge on Quantum States. We also consider an, a priori, completely different problem: *Non-Interactive and Non-destructive Zero-Knowledge proofs on Quantum States* (NIZKoQS). Suppose that Alice would like to send a quantum state to Bob (non-interactively, meaning that a single message is sent from Alice to Bob):

Can Alice prove (non-interactively) to Bob that the sent quantum state belongs to a given set, without destroying or revealing any additional information about that state?

For instance, Alice may want to prove that in the n-qubits state that has been sent, two qubits are entangled (forming a Bell state) and that the remaining qubits are random $|0\rangle$ or $|1\rangle$ states. Moreover, Alice may want to partially reveal, completely reveal or completely hide the position of this Bell pair. Or even stronger, Alice may want to prove that the first qubit is part of the Bell pair only if she knows the private key associated to some Bitcoin public keys; one can imagine a lot of such properties.

This task is the quantum analogue of what is known classically as *Zero-Knowledge* (ZK). Classical Zero-Knowledge proofs and Interactive Proofs systems have been introduced thirty years ago [GMR85, BM88], and allow a prover to prove a statement to a verifier without revealing anything beyond the fact that this statement is true. Zero-Knowledge proofs have been proposed for any language in NP [GMW91]. While *Non-Interactive* Zero-Knowledge (NIZK) proofs are known to be impossible in the plain model [FS87], NIZK can be obtained in the Common Reference String model [BFM88] or in the Random Oracle model by using the famous Fiat-Shamir transformation [GO94]. The security of classical Zero-Knowledge proofs were also extended to be secure against malicious quantum provers [Wat09, Unr12, BS20], and much work has been done to extend these protocols to remove interactivity [DFM+19, LZ19] or deal with multiple (potentially quantum)

provers, targeting larger classes like IP, QMA, MIP, MIP* or considering "dequantized" verifiers ([IY88, BJS+16, GSY19, CVZ20, ACG+20, BG20, BCK+20, VZ20, Shm20, MY21], see also the review [VW16]).

However, these works focus on *classical* languages (even in QMA the language is still classical, despite the fact that the witness can be quantum). A method for deriving similar properties on a *quantum* language is to use a generic quantum secure multiparty computing protocol (QSMPC) [DNS12, DGJ+20, KKM+21]. However, these protocols are interactive, and to the best of our knowledge, there are no results which provide one-shot Zero-Knowledge proofs on quantum states. Note that because a quantum state cannot even be copied, the above protocols cannot be turned into non-interactive protocols using the Fiat-Shamir transformation [FS87].

At a first glance, non-destructive and non-interactive proofs on quantum states seems impossible: when receiving a normal quantum state, quantum mechanics tells us that the only way to extract information from this state is to alter it irrevocably. Moreover, quantum mechanics asserts that even an unbounded party cannot distinguish some classes of quantum states: for instance, it is impossible to distinguish a qubit sampled uniformly at random from the set $\{|0\rangle, |1\rangle\}$ from a qubit sampled uniformly from the set $\{|+\rangle, |-\rangle\}$. However, we show in Chapter 7 how these fundamental limitations can be circumvented. We explain how it is possible to achieve Non-Interactive and Non-Destructive Zero-Knowledge proofs on Quantum States (NIZKoQS) for non-trivial quantum languages. Surprisingly, the non-interactive classical-client RSP protocols introduced in the previous chapters turn out to be essential tools in our methodology.

Chapter 7, Section 7.4: Extension to a Mutiparty Setting. RSP protocols are typically designed to be used between one sender and one receiver. However, we also consider extensions in which multiple (potentially untrusted) senders generate a given state collaboratively on a (potentially untrusted) source. This also allows us to quantify the leakage of our RSP protocol when partial information is leaked about the secret key. Combined with NIZKoQS we can show how a source can share a GHZ state by arbitrarily filtering the participants, in such a way that nobody—not even the source—knows who shared a part of the GHZ state.

As discussed in Section 7.1.3, this could have use cases in all the protocols in which the protocol starts by the distribution of a GHZ state (or a Bell pair), including, but not limited to, Quantum Secret Sharing [HBB99], Quantum Teleportation [BBC+93], Entanglement Distillation [BBP+96a, BBP+96b, BDS+96], Device-Independent Quantum-Key-Distribution [MY98], Anonymous Transmission [CW05], Quantum Routing [PWD18,

MMG19]. More precisely, the source could run our protocol to generate the GHZ state to distribute, and the participants will then be able to use the source as before: that way, nobody—not even a malicious source—should be able to know who is participating in the protocol, while the source has guarantees that the participants fulfill some properties.

In particular, our approach would allow a party to do a quantum secret sharing of a quantum state to unknown filtered parties, and we can also imagine new applications, including a quantum onion-like routing protocol, in which a quantum message is routed through an untrusted network while preserving both the source and the destination of the quantum state as discussed in Section 7.1.3.

Publications. The results presented in this thesis were presented in the following works:

- [CCK+18] is the first publication regarding classical-client RSP protocols (it was published much later in [CCK+21] and improves the work started during my Master 1 thesis [Col17]). We describe there a first version of our QFactory protocol, but we only prove the security in a weak "honest-but-curious" model.
  I presented this work at QCrypt2018[8].
- [CCK+19], published in ASIACRYPT 2019[9], improves the QFactory protocol and provides a full proof of security, against arbitrary malicious adversaries. We also study how to deal with more standard security assumptions, namely LWE with polynomial noise ratio.
- [BCC+20] proves the impossibility results regarding composable classical-client RSP protocols and classical-client UBQC. We also prove that QFactory can securely be used inside UBQC when targeting non-composable security[10].
  I presented this work at ASIACRYPT 2020[11].
- [CGK21] presents our work on Non-Interactive and Non-Destructive Zero-Knowledge on Quantum States and the extension to multiparty authorized GHZ preparation. It also extends QFactory to produce multi-qubit states and does a proper analysis of the LWE parameters when relying on LWE with superpolynomial noise ratio. It is currently under submission.

[8] https://youtu.be/u8gUPcLyuPo

[9] ASIACRYPT is a conference with proceedings.

[10] In this thesis this last result is actually improved, as we extend it to any "basis-blind" computationally secure RSP protocol.

[11] https://youtu.be/ROqk9tZ_VxA

Organization of the Thesis. The thesis is split into two introductory chapters and four research chapters. You can find in Figure 1.1 the dependency relation between the different chapters of this thesis. The full table of contents is available on page xi.

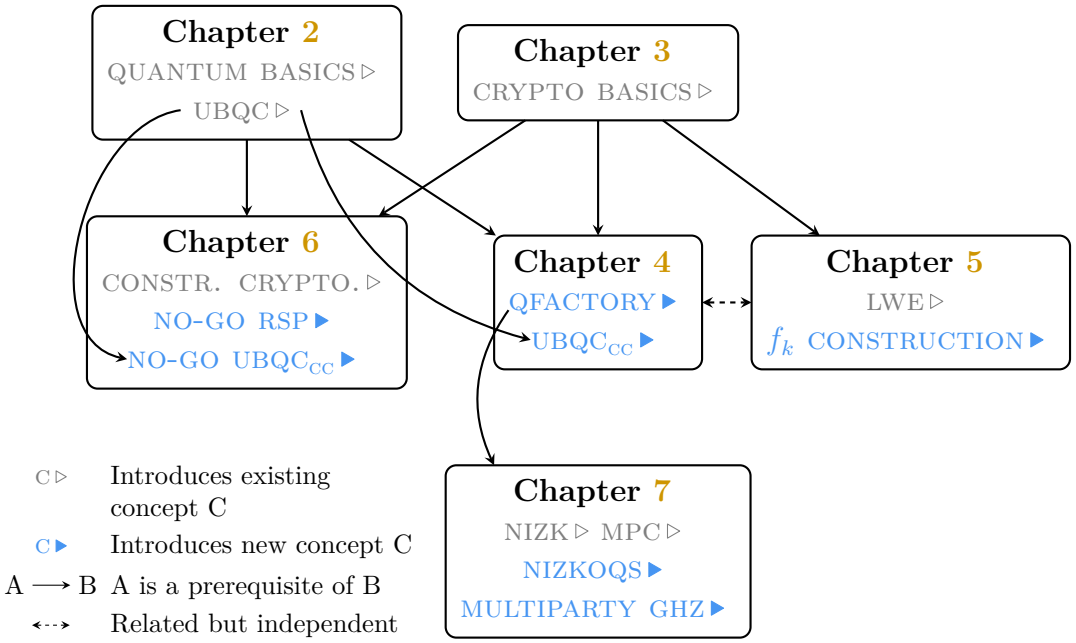


Figure 1.1: Dependency relation between the different chapters.

Chapter 2

Introduction to Quantum Computing

"The totality is not, as it were, a mere heap, but the whole is something besides the parts."

— Aristotle, *Metaphysica*

Since a quantum system is first and foremost a physical system, it must obey the "laws of Nature". However, these laws can behave very differently depending on the scale at which they are considered. The aim of quantum mechanics is to provide a fundamental theory characterizing them, targeting especially the world of the infinitely small (typically at the scale of a photon or an atom). Quantum mechanics embeds many concepts, ranging from photons wave functions to atomic orbitals, and from electrons spins to the Schrödinger equation. However, in quantum computing, we prefer to abstract the underlying physical system used to perform the computations, similar to the way classical computer science uses bits instead of electric currents.

In this chapter, we will cover the basics of quantum computing. We will start by introducing its mathematical formalism, before describing the ZX-Calculus, a very handy tool to graphically reason on quantum circuits. Note that it is not mandatory to know the ZX-Calculus to understand this thesis—a careful reader could check any computation done in the ZX-Calculus using standard linear algebra—but computations can be highly simplified this way, are more elegant, and also provide a better intuition of what's going on, beyond the famous "Shut up and calculate" motto [Mer89]. We will finish this chapter by showing some quantum protocols and properties that will prove useful later.

Note that an interested reader can find much more details in the famous book of Nielsen and Chuang [NC10]. For the ZX-calculus, initially introduced by Bob Coecke and Ross Duncan in [CD08], we recommend the review [vdWet20] which provides a nice overview of the existing tools offered by the ZX-calculus. The "Dodo book" [CK17] also offers a very interesting and general approach to diagrammatic reasoning.

2.1 Quantum Computing: Mathematical Formalism

2.1.1 Quantum States and Dirac Notation

Hilbert Space. Hilbert spaces play a central role in quantum computing since they represent the world in which quantum states are living. A finite dimensional Hilbert space $\mathcal{H}$—we only consider finite dimensions in this thesis, and more generally in quantum computing—is a finite dimensional complex vector space equipped with an Hermitian inner product $\langle \cdot | \cdot \rangle \colon \mathcal{H} \times \mathcal{H} \to \mathbb{C}$. More precisely, it fulfills the following conditions:

- **Complex vector space**: $\mathcal{H}$ is a set equipped with two operations $+ \colon \mathcal{H} \times \mathcal{H} \to \mathcal{H}$ and $\cdot : \mathbb{C} \times \mathcal{H} \to \mathcal{H}$ (the symbol $\cdot$ is often omitted) such that the following axioms hold for all vectors $(\mathbf{u}, \mathbf{v}, \mathbf{w}) \in \mathcal{H}^3$ and scalars $(\lambda, \mu) \in \mathbb{C}^2$:
  - *Associativity of addition*: $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
  - *Commutativity of addition*: $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
  - *Identity element of addition*: there exists an element $\mathbf{0} \in \mathcal{H}$ such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$
  - *Inverse element of addition*: for any $\mathbf{v} \in \mathcal{H}$ there exists an element $-\mathbf{v} \in \mathcal{H}$ such that $\mathbf{v} - \mathbf{v} \coloneqq \mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
  - *Compatibility of multiplication*: $\lambda(\mu \mathbf{v}) = (\lambda\mu)\mathbf{v}$
  - *Identity of multiplication*: $1 \cdot \mathbf{v} = \mathbf{v}$
  - *Distributivity*: $\lambda(\mathbf{u} + \mathbf{v}) = \lambda\mathbf{u} + \lambda\mathbf{v}$ and $(\lambda + \mu)\mathbf{v} = \lambda\mathbf{v} + \mu\mathbf{v}$

  A basis for $\mathcal{H}$ is a set $\mathcal{B} \subseteq \mathcal{H}$ such that any element in $\mathcal{H}$ can be written as a linear combination of vectors in $\mathcal{B}$. The dimension n of $\mathcal{H}$ is the smallest possible number of elements in any basis of $\mathcal{H}$. We assume that $\mathcal{H}$ is finite dimensional, i.e. that $n \in \mathbb{N}$.
- **Hermitian inner product**: $\mathcal{H}$ is equipped with an operation $\langle \cdot | \cdot \rangle \colon \mathcal{H} \times \mathcal{H} \to \mathbb{C}$ which is:
  - *Conjugate symmetric*: for all $(\mathbf{v}, \mathbf{w}) \in \mathcal{H}^2$, we have $\langle \mathbf{v} | \mathbf{w} \rangle = \overline{\langle \mathbf{w} | \mathbf{v} \rangle}$
  - *Linear in the second argument*: for all $(\mathbf{v}, \mathbf{w}_1, \mathbf{w}_2) \in \mathcal{H}^2, (\lambda, \mu) \in \mathbb{C}^2$, we have $\langle \mathbf{v} | \lambda \mathbf{w}_1 + \mu \mathbf{w}_2 \rangle = \lambda \langle \mathbf{v} | \mathbf{w}_1 \rangle + \mu \langle \mathbf{v} | \mathbf{w}_2 \rangle$

– *Positive definite*: for all $\mathbf{v} \in \mathcal{H}$[2], we have $\langle \mathbf{v}|\mathbf{v}\rangle \geq 0$, with equality if and only if $\mathbf{v} = 0$

The inner product also defines a norm $\|\cdot\|_2 \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$ (or simply $\|\cdot\|$) on $\mathcal{H}$: for any $\mathbf{v} \in \mathcal{H}$, $\|\mathbf{v}\| \coloneqq \sqrt{\langle \mathbf{v}|\mathbf{v}\rangle}$.

In finite dimensions all Hilbert spaces $\mathcal{H}_n$ of dimension $n \in \mathbb{N}$ are isomorphic to $\mathbb{C}^n$ equipped with its canonical inner space:

$$\forall (v, w) \in \mathbb{C}^n, \langle v|w\rangle \coloneqq \sum_i \bar{v}_i w_i \tag{2.1}$$

For this reason, $\mathcal{H}_n$ can always be assumed to be this later Hilbert space. Note that in the following we will slighly abuse notations and we may use vectors to denote elements of a Hilbert space $\mathcal{H}$ and matrices to denote linear operations on Hilbert spaces. This must be understood as if we fixed an orthonormal basis[1] for this Hilbert space (such as the canonical basis when considering $\mathbb{C}^n$), and expressed each vector/linear operation in this basis.

Pure Quantum State. Quantum states exist in two flavours: pure and mixed. Mixed states are a generalisation of pure quantum states, in which some parts of the system can be discarded. However, pure states are still extremely useful as they characterize isolated systems. Pure states are formally defined as follows:

A *pure quantum state* is fully described as a unit vector in a Hilbert space $\mathcal{H}$.

Moreover, two quantum states $\mathbf{v} \in \mathcal{H}$ and $\mathbf{w} \in \mathcal{H}$ are considered equal[2] if and only if they differ only by a global phase, i.e. if there exists $\theta \in [0, 2\pi)$ such that $\mathbf{v} = e^{i\theta}\mathbf{w}$. We say that the global phase is not observable.

The Hilbert space $\mathcal{H}_2$ plays an essential role since vectors in $\mathcal{H}_2$ are *qubits*, i.e. quantum bits. Two states are particularly important:

$$|0\rangle \coloneqq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle \coloneqq \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.2}$$

The state $|0\rangle$ (pronounced "ket 0", for a reason that will become clear later) informally corresponds to a classical bit 0, while $|1\rangle$ corresponds to the classical bit 1. Note that

[1] An orthonormal basis is a basis $\{e_i\}_i$ in which $\forall i, j$, $\langle e_i|e_j\rangle = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta.

[2] One could have been slightly more formal by defining quantum states using a quotient space, but it is more cumbersome to use and describe.

$\{|0\rangle, |1\rangle\}$ is an orthonormal basis of $\mathcal{H}_2$. In particular, any state $|\psi\rangle \in \mathcal{H}_2$ can be written as

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \tag{2.3}$$

for some $(a, b) \in \mathbb{C}^2$, with the normalization condition $|a|^2 + |b|^2 = 1$. When a and b are not null, we say that $|\psi\rangle$ is in *superposition* since it is a sum of multiple basis vectors.

Note that in practice, a qubit should be implemented using a physical system, like a cold atom or a polarized photon. For instance, in that latter case, the modulus squared of the first coordinate of a qubit may represent its amount of horizontal polarization while those of the second coordinate may represent its amount of vertical polarization. However, these technical "details" are not important in quantum computing, only the vector representing that state will matter.

Adjoint and Dirac Notation. The symbol $|\cdot\rangle$ comes from the *Dirac notation*, and will be extremely useful to represent quantum states in a concise way. In this notation, the symbol $|\cdot\rangle$ (pronounced "ket") is used to denote the fact that the object $\cdot$ is a vector. For instance, we usually denote by $|\psi\rangle$ an arbitrary quantum state labelled ψ (Greek letters will be used to denote arbitrary vectors). We can of course do operations on kets like on any other matrix, for instance to define the following states (θ being a real angle):

$$|+\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \qquad |-\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix} \tag{2.4}$$

$$|+_\theta\rangle \coloneqq \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}\,|1\rangle) = \begin{pmatrix} 1/\sqrt{2} \\ e^{i\theta}/\sqrt{2} \end{pmatrix} \tag{2.5}$$

Note that $\{|+\rangle, |-\rangle\}$ forms an orthonormal basis, like $\{|0\rangle, |1\rangle\}$.

The *Hermitian adjoint* $\mathbf{A}^\dagger \colon \mathcal{H}_B \to \mathcal{H}_A$ (read "dagger") of a linear operator $\mathbf{A} \colon \mathcal{H}_A \to \mathcal{H}_B$ will prove to be particularly useful when defining the different operations that one can perform on a quantum state. The operator $\mathbf{A}^\dagger$ is the unique linear operator such that for all $v_1 \in \mathcal{H}_A$ and $v_2 \in \mathcal{H}_B$, $\langle v_2|\mathbf{A}v_1\rangle = \langle \mathbf{A}^\dagger v_2|v_1\rangle$. This translates quite nicely in term of matrices: if $\hat{\mathbf{A}}$ is the matrix representation[3] of the linear operator $\mathbf{A}$ in an orthonormal basis, then the matrix $\hat{\mathbf{A}}^\dagger$—corresponding to the matrix representation of $\mathbf{A}^\dagger$ in this basis—is the complex conjugate transpose of $\hat{\mathbf{A}}$, i.e. $\hat{\mathbf{A}}^\dagger = \overline{\hat{\mathbf{A}}}^T$. If we identify the matrix and its operator, this gives:

$$\mathbf{A}^\dagger = \overline{\mathbf{A}}^T \tag{2.6}$$

[3] We will later identify $\mathbf{A}$ and its matrix representation $\hat{\mathbf{A}}$ for simplicity.

The Dirac notation also defines a $\langle \cdot |$ operation (pronounced "bra") to denote the Hermitian adjoint of a vector:

$$\langle \psi | \coloneqq |\psi\rangle^\dagger = \overline{|\psi\rangle}^T = \begin{pmatrix} \bar{\psi}_1 & \dots & \bar{\psi}_n \end{pmatrix} \tag{2.7}$$

This way, the Dirac notation now has a nice property; multiplying a "bra" with a "ket" gives you a "braket", i.e. an inner product:

$$\langle \psi | \, |\psi\rangle = \begin{pmatrix} \bar{\psi}_1 & \dots & \bar{\psi}_n \end{pmatrix} \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_n \end{pmatrix} = \sum_i \bar{\psi}_i \psi_i = \langle \psi | \psi \rangle \tag{2.8}$$

Moreover, $|\psi\rangle \langle \psi|$ is the projector on $\langle \psi |$. Note also that $\langle v_2 | \mathbf{A} v_1 \rangle = \langle v_2 | \, \mathbf{A} \, | v_1 \rangle$.

Bloch Sphere. We will now see a way to graphically represent a single qubit. This is made possible thanks to the fact that the global phase of a quantum state is not observable, which means that it is impossible to distinguish two states $|\phi\rangle$ and $|\psi\rangle$ when there exists an angle $\theta \in \mathbb{R}$ such that $|\phi\rangle = e^{i\alpha} |\psi\rangle$. For that reason, we can always factor out and remove the global phase of an arbitrary qubit.

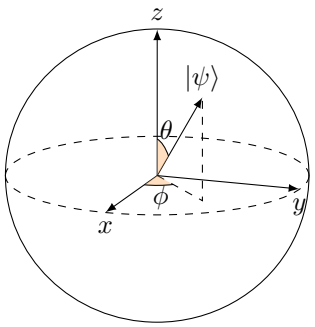
$$\begin{aligned} |\psi\rangle &= a |0\rangle + b |1\rangle && (2.9) \\ &= r_1 e^{i\alpha} |0\rangle + r_2 e^{i\beta} |1\rangle && (2.10) \\ &= e^{i\alpha} (r_1 |0\rangle + r_2 e^{i(\beta-\alpha)} |1\rangle) && (2.11) \\ &\simeq r_1 |0\rangle + r_2 e^{i(\beta-\alpha)} |1\rangle && (2.12) \end{aligned}$$

where $a = r_1 e^{i\alpha}$, $b = r_2 e^{i\beta}$, $(r_1, r_2) \in \mathbb{R}^2_{\geq 0}$, $(\alpha, \beta) \in [0, 2\pi)$ and $|a|^2 + |b|^2 = r_1^2 + r_2^2 = 1$. Therefore, there exists $\theta' \in [0, \pi/2)$ such that $r_1 = \cos(\theta')$ and $r_2 = \sin(\theta')$. By defining $\theta \coloneqq 2\theta' \in [0, \pi)$ and $\phi \coloneqq \beta - \alpha \in [0, 2\pi)$, we have $r_1 = \cos\left(\frac{\theta}{2}\right)$ and $r_2 = \sin\left(\frac{\theta}{2}\right)$ and therefore:

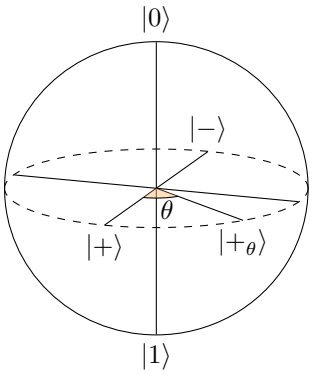
$$|\psi\rangle \simeq \cos\left(\frac{\theta}{2}\right) |0\rangle + \sin\left(\frac{\theta}{2}\right) e^{i\phi} |1\rangle \tag{2.13}$$

While $|\psi\rangle$ was before characterized by two complex numbers (dimension 4), it is now possible to describe $|\psi\rangle$ using two real angles (dimension 2). These angles can be used to uniquely represent any qubit as a point on a sphere known as the *Bloch sphere*, pictured Figure 2.1a.

On the Bloch sphere, orthogonal states are antipodal. While this may seem counter intuitive, it comes directly from the fact that we combined points equal up to a global phase as pictured in Figure 2.2.

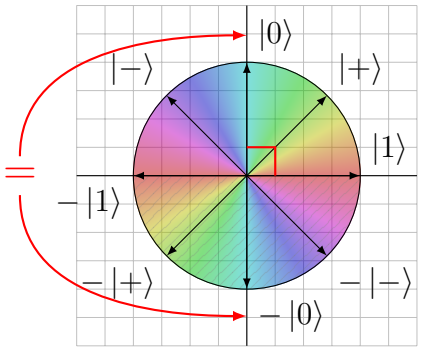


(a) Representation of an arbitrary qubit $|\psi\rangle$ on the Bloch sphere.

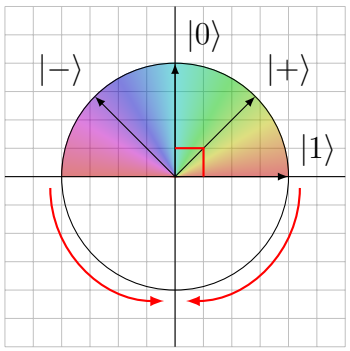


(b) Representation of the most usual vectors on the Bloch sphere.

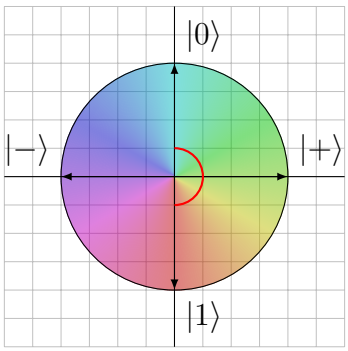
Figure 2.1: Bloch sphere.



(a) Usual planar representation of vectors. Orthogonal states are represented orthogonally (right angle). Notice the redundancy of the vectors when doing the quotient on the global phase (for instance, $|0\rangle = -|0\rangle$).



(b) It is therefore tempting to remove this redundant part and merge antipodal points...



(c) Now, orthogonal states are antipodal, and we have a unique representation for each vector.

Figure 2.2: Intuitive construction of the Bloch sphere (we represent only a 2D cut of the space $\mathcal{H}_2$ where all vectors are real).

2.1.2 Operations on qubits

A quantum state can be modified in multiple ways. The two basic operations that can be applied on a quantum state are unitaries and measurements.

Unitaries. A *unitary* $\mathbf{U}\colon \mathcal{H} \to \mathcal{H}$ is a linear map such that $\mathbf{U}^\dagger\mathbf{U} = \mathbf{U}\mathbf{U}^\dagger = \mathbf{I}$ (where $\mathbf{I}$ is the identity matrix). In particular, it is invertible (its inverse is $\mathbf{U}^\dagger$) and it preserves the norm of the input state, which is essential to ensure that the output state is a well formed quantum state. Some unitaries (sometimes called *gates* by analogy with circuits) will be particularly useful later:

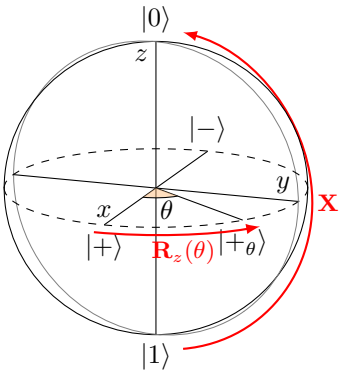
$$\mathbf{I} \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{X} \coloneqq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \mathbf{Z} \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.14}$$

$$\mathbf{T}(\theta) \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \qquad \mathbf{R}_z(\theta) \coloneqq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \qquad \mathbf{H} \coloneqq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.15}$$

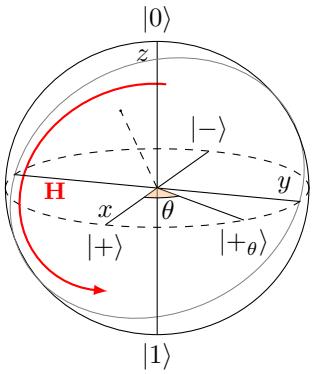
The unitary $\mathbf{I}$ is the identity map: it does not change the input state. In particular, $\mathbf{I}\left|0\right\rangle = \left|0\right\rangle$ and $\mathbf{I}\left|1\right\rangle = \left|1\right\rangle$. On the other hand, $\mathbf{X}$ is the equivalent of the classical NOT operation: it turns $\left|0\right\rangle$ into $\left|1\right\rangle$ and $\left|1\right\rangle$ into $\left|0\right\rangle$. Moreover, the states $\left|+\right\rangle$ and $\left|-\right\rangle$ are kept unchanged when applying $\mathbf{X}$ (up to a global phase). The unitaries $\mathbf{Z}$, $\mathbf{T}$ and $\mathbf{R}_z$ (note that $\mathbf{Z}$ and $\mathbf{T}$ are particular cases of $\mathbf{R}_z(\theta)$) add a phase on the $\left|1\right\rangle$ component: $\mathbf{R}_z(\theta)(a\left|0\right\rangle + b\left|1\right\rangle) = a\left|0\right\rangle + be^{i\theta}\left|1\right\rangle$. In particular, $\mathbf{Z}$ behaves similarly to $\mathbf{X}$ except that it swaps $\left|0\right\rangle$ and $\left|1\right\rangle$, and the fix points are $\left|+\right\rangle$ and $\left|-\right\rangle$. Finally, the Hadamard gate $\mathbf{H}$ turns the basis $\{\left|0\right\rangle, \left|1\right\rangle\}$ into $\{\left|+\right\rangle, \left|-\right\rangle\}$. In particular, it maps the state $\left|0\right\rangle$ to $\left|+\right\rangle$, which is a uniform superposition on all vectors of the $\{\left|0\right\rangle, \left|1\right\rangle\}$ basis. This property will also be very useful later when dealing with multiple qubits.

Note that all the unitaries that can be performed on a single qubit are rotations on the Bloch sphere: the rotations of the above gates are drawn in Figure 2.3. Moreover, the gates $\mathbf{H}$ and $\mathbf{R}_z$ can be composed to produce any one-qubit unitary, for instance using the Euler angle decomposition. It is also possible to efficiently approximate any one-qubit unitary using only $\mathbf{H}$ and $\mathbf{T}$ gates, for instance using the Solovay-Kitaev algorithm [Kit97, DN06].

Tensor product. Classically, having a single bit is usually not very useful, and we often need to use many bits. Quantumly, we can also gather multiple qubits (or more generally



(a) The $\mathbf{X}$ and $\mathbf{R}_z$ unitaries. Observe that $\mathbf{X}$ is a rotation around the x axis and $\mathbf{R}_z$ is a rotation around the z axis



(b) The Hadamard unitary is a rotation around the $x + z$ axis.

Figure 2.3: Some unitaries pictured on the Bloch sphere.

quantum states) into a single state: the state obtained after combining $|\phi\rangle \in \mathcal{H}_A$ and $|\psi\rangle \in \mathcal{H}_B$ will be denoted $|\phi\rangle \otimes |\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$. For instance, $|0\rangle \otimes |1\rangle$ represents a system composed of two qubits, the first qubit being $|0\rangle$ and the second being $|1\rangle$.

Formally, $(\mathcal{H}_A \otimes \mathcal{H}_B, \otimes)$ is defined[4] as a tensor product, i.e. a couple in which:

- $\mathcal{H}_A \otimes \mathcal{H}_B$ is a vector space over the same field $\mathbb{F}$ as $\mathcal{H}_A$ and $\mathcal{H}_B$ (we only consider $\mathbb{F} = \mathbb{C}$ in the case of quantum mechanics),
- $\otimes\colon \mathcal{H}_A \times \mathcal{H}_B \to \mathcal{H}_A \otimes \mathcal{H}_B$ is a bilinear map (we use the infix notation for $\otimes$), i.e. it respects the following properties:
  - **Homogeneity**: For any $c \in \mathbb{F}$, $|\phi\rangle \in \mathcal{H}_A$, $|\psi\rangle \in \mathcal{H}_B$:

$$c(|\phi\rangle \otimes |\psi\rangle) = (c\,|\phi\rangle) \otimes |\psi\rangle = |\phi\rangle \otimes (c\,|\psi\rangle) \tag{2.16}$$

  - **Left additivity**: For any $(|\phi_1\rangle, |\phi_2\rangle) \in \mathcal{H}_A^2$, $|\psi\rangle \in \mathcal{H}_B$:

$$(|\phi_1\rangle + |\phi_2\rangle) \otimes |\psi\rangle = |\phi_1\rangle \otimes |\psi\rangle + |\phi_2\rangle \otimes |\phi_2\rangle \tag{2.17}$$

  - **Right additivity**: For any $|\phi\rangle \in \mathcal{H}_A$ and $(|\psi_1\rangle, |\psi_2\rangle) \in \mathcal{H}_B^2$:

$$|\phi\rangle \otimes (|\psi_1\rangle + |\psi_2\rangle) = |\phi\rangle \otimes |\psi_1\rangle + |\phi\rangle \otimes |\psi_2\rangle \tag{2.18}$$

- if $\{|e_i\rangle\}_i$ is a basis of $\mathcal{H}_A$ and $\{|f_i\rangle\}$ is a basis of $\mathcal{H}_B$, then $\{|e_i\rangle \otimes |f_i\rangle\}$ is a basis of $\mathcal{H}_A \otimes \mathcal{H}_B$,
- for any $(|\phi_1\rangle, |\phi_2\rangle) \in \mathcal{H}_A^2$, $(|\psi_1\rangle, |\psi_2\rangle) \in \mathcal{H}_B^2$:

$$\langle |\phi_1\rangle \otimes |\phi_2\rangle \,|\, |\psi_1\rangle \otimes |\psi\rangle_2 \rangle \coloneqq (\langle\phi_1| \otimes \langle\phi_2|)(|\psi_1\rangle \otimes |\psi\rangle_2) = \langle\phi_1|\psi_1\rangle\langle\phi_2|\psi_2\rangle \tag{2.19}$$

[4]The reader should not be confused by the usage of $\otimes$ inside $\mathcal{H}_A \otimes \mathcal{H}_B$: $\mathcal{H}_A \otimes \mathcal{H}_B$ is just the name of the vector space, and could have been replaced with any symbol, like $\mathcal{T}$.

For a bit string $\mathbf{b} = b_1| \dots |b_n \in \{0,1\}^n$, we also use the notation $|\mathbf{b}\rangle \coloneqq |b_1\rangle\,|b_2\rangle \dots |b_n\rangle \coloneqq |b_1\rangle \otimes \dots |b_n\rangle$ (the tensor product can often omitted). We can for instance check that applying $\mathbf{X} \otimes \mathbf{I}$ on $|00\rangle$ flips the first bit:

$$\begin{aligned}(\mathbf{X} \otimes \mathbf{I})\,|00\rangle &= (\mathbf{X} \otimes \mathbf{I})(|0\rangle \otimes |0\rangle) && (2.20)\\ &= (\mathbf{X}\,|0\rangle) \otimes (\mathbf{I}\,|0\rangle) && (2.21)\\ &= (|1\rangle) \otimes (|0\rangle) && (2.22)\\ &= |10\rangle && (2.23)\end{aligned}$$

If $\mathcal{H}_n$ and $\mathcal{H}_m$ are respectively n-dimensional and m-dimensional Hilbert spaces, then the dimension of $\mathcal{H}_n \otimes \mathcal{H}_m$ is $n \times m$. In particular, the dimension of a system composed of n qubits is 2^n, which is exponential in the number of qubits. We will denote by $\mathcal{H}^{\otimes n}$ the tensor product $\underbrace{\mathcal{H} \otimes \dots \otimes \mathcal{H}}_{n \text{ times}}$: a system composed of n qubits will therefore be denoted $\mathcal{H}_2^{\otimes n}$. It is also common to group qubits together into *registers*: for instance if the n first qubits of the space $\mathcal{H}_2^{n+m}$ are used to encode an input, and the m remaining qubits are used to encode some outputs, we will refer to the first group of n qubits as the first register, and to the second group of m qubits as the second register. This notion trivially extends to more than 2 registers.

One can also "lift" operations acting separately on each system to operations acting on the global system. Given $\{|e_i\rangle\}_i$ a basis of $\mathcal{H}_{A_1}$, $\{|f_i\rangle\}$ a basis of $\mathcal{H}_{B_1}$ and two linear map $\mathbf{U}_A \colon \mathcal{H}_{A_1} \to \mathcal{H}_{A_2}$ and $\mathbf{U}_B \colon \mathcal{H}_{B_1} \to \mathcal{H}_{B_2}$, we define $\mathbf{U}_A \otimes \mathbf{U}_B : \mathcal{H}_{A_1} \otimes \mathcal{H}_{B_1} \to \mathcal{H}_{A_2} \otimes \mathcal{H}_{B_2}$ as the only linear map such that for all i, j:

$$(\mathbf{U}_A \otimes \mathbf{U}_B)(|e_i\rangle \otimes |f_j\rangle) \coloneqq (\mathbf{U}_A\,|e_i\rangle) \otimes (\mathbf{U}_B\,|f_j\rangle) \tag{2.24}$$

It is also possible to use the *Kronecker product* to compute a tensor product in term of matrices: Given two matrices or vectors $\mathbf{A} \in \mathbb{C}^{m \times n}$ and $\mathbf{B} \in \mathbb{C}^{p \times q}$, we define the Kronecker product between $\mathbf{A}$ and $\mathbf{B}$ as:

$$\mathbf{A} \otimes \mathbf{B} \coloneqq \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix} \in \mathbb{C}^{mp \times nq} \tag{2.25}$$

For instance:

$$|0\rangle \otimes |0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |0\rangle \otimes |1\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \qquad |1\rangle \otimes |0\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \qquad |1\rangle \otimes |1\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \tag{2.26}$$

and

$$\mathbf{X} \otimes \mathbf{I} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{2.27}$$

The above example Eq. (2.26) shows the nice property of the Kronecker tensor product: let $\mathbf{b} = b_{n-1}| \dots |b_0 \in \{0,1\}^n$ be a bit string, let $i = \sum_j b_j 2^j \in \mathbb{N}$ be the number whose binary representation is $\mathbf{b}$, then $|\mathbf{b}\rangle = |b_{n-1}\rangle \otimes \cdots \otimes |b_0\rangle$ is the vector having a 1 at the i-th entry and 0 elsewhere. Moreover, we can easily check again that $(\mathbf{X} \otimes \mathbf{I}) |00\rangle = |10\rangle$:

$$(\mathbf{X} \otimes \mathbf{I}) |00\rangle \overset{(2.27)}{=} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle \tag{2.28}$$

It is now possible to define unitaries on multiple qubits:

$$\mathbf{CNOT} \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \wedge\mathbf{Z} \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.29}$$

These two gates are called *controlled unitaries*, because they apply a given unitary ($\mathbf{X}$ for $\mathbf{CNOT}$ and $\mathbf{Z}$ for $\wedge\mathbf{Z}$) on the second qubit when the first qubit is $|1\rangle$ and the identity when the first qubit is $|0\rangle$. This way for $a \in \{0,1\}$, $\mathbf{CNOT} |0a\rangle = |0a\rangle$ and $\mathbf{CNOT} |1a\rangle = |1(1-a)\rangle$. Similarly, $\wedge\mathbf{Z} |11\rangle = -|11\rangle$ while the states $|00\rangle$, $|01\rangle$ and $|10\rangle$ are fixed points of $\wedge\mathbf{Z}$ (remark that $\wedge\mathbf{Z}$ is symmetric).

For any gate $\mathbf{G}$ and integer n, we define $\mathbf{G}^{\otimes n} \coloneqq \mathbf{G} \otimes \cdots \otimes \mathbf{G}$ as the tensor of n gates $\mathbf{G}$. We will often use the following properties on the Hadamard gate:

Lemma 2.1.1. *Let $n \in \mathbb{N}$ and $x \in \{0,1\}^n$. Then, $H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{b \in \{0,1\}^n} (-1)^{\langle b,x \rangle} |b\rangle$, where we define $\langle b, x \rangle \coloneqq \oplus_i b_i x_i$. In particular, if $x = 0 \dots 0$, it creates $\frac{1}{\sqrt{2^n}} \sum_{b \in \{0,1\}^n} |b\rangle$, the uniform superposition on $\{0,1\}^n$.*

Proof. This well known property can be proved by induction on n. We have

$$\mathbf{H} |x_0\rangle = \frac{1}{\sqrt{2}} |0\rangle + (-1)^{x_0} |1\rangle = \frac{1}{\sqrt{2}} \sum_{b \in \{0,1\}} (-1)^{b x_0} |b\rangle \tag{2.30}$$

and

$$\mathbf{H}^{\otimes(n+1)} |x\rangle = (\mathbf{H}^{\otimes n} |x_{1:n}\rangle) \otimes (\mathbf{H} |x_{n+1}\rangle) \tag{2.31}$$

$$= \left(\frac{1}{\sqrt{2^n}} \sum_{b\in\{0,1\}^n} (-1)^{\oplus_i b_i x_i} |i\rangle\right) \otimes \left(\frac{1}{\sqrt{2}} \sum_{b'\in\{0,1\}} (-1)^{b' x_{n+1}} |b'\rangle\right) \tag{2.32}$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{b\in\{0,1\}^{n+1}} (-1)^{\oplus_i b_i x_i} |b\rangle \qquad \square$$

For a bit string $\mathbf{a} \in \{0,1\}^n$ and a one-qubit gate $\mathbf{G}$, we use the notation $\mathbf{G}^{\mathbf{a}} = \mathbf{G}^{\mathbf{a}_1} \otimes \cdots \otimes \mathbf{G}^{\mathbf{a}_n}$ to denote the circuit that applies G on qubit i iff $a[i] = 1$.

It is possible to use the Dirac notation to represent matrices, by remarking that any matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ can be decomposed as:

$$\mathbf{A} = \sum_{i,j} \mathbf{A}_{ij} |i\rangle \langle j| \tag{2.33}$$

For instance, $\mathbf{X} = |1\rangle\langle 0| + |0\rangle\langle 1|$ and this way we have:

$$\mathbf{X} |0\rangle = (|1\rangle\langle 0| + |0\rangle\langle 1|) |0\rangle \tag{2.34}$$

$$= |1\rangle \underbrace{\langle 0|0\rangle}_{=1} + |0\rangle \underbrace{\langle 1|0\rangle}_{=0} \tag{2.35}$$

$$= |1\rangle \tag{2.36}$$

Measurements. While a classical bit can only have two values (0 or 1), a qubit can have infinitely many values (all the states of the form $a|0\rangle + b|1\rangle$ with norm 1). However, it does not mean that it is possible to use a qubit to store an infinite amount of data: very little information is actually accessible to an outside observer. For instance, as we will see later, it is impossible to extract more than one bit of information given access to an unknown qubit. Any such extraction process is called a *measurement*: measurements are therefore bridges between the classical world and the quantum world. Measurements also inherently alter quantum states: some measurements even completely destroy the state. We will first see the most general definition of a measurement, and we will then study some useful special cases.

A *quantum measurement* is described by an indexed family $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ such that

$$\sum_m \mathbf{M}_m^\dagger \mathbf{M}_m = \mathbf{I} \tag{2.37}$$

When measuring a state $|\psi\rangle$, we obtain a classical outcome $m \in \mathcal{M}$: the probability of getting outcome m is given by $\langle\psi| \mathbf{M}_m^\dagger \mathbf{M}_m |\psi\rangle$ (Eq. (2.37) is necessary to ensure the

probabilities sum up to one). The state of the system after the measurement will be the state $\mathbf{M}_m |\psi\rangle$, up to a normalization factor:

$$\frac{\mathbf{M}_m |\psi\rangle}{\sqrt{\langle\psi| \mathbf{M}_m^\dagger \mathbf{M}_m |\psi\rangle}} \tag{2.38}$$

Remark 2.1.2. Note that this definition is very general: any process that outputs a classical information can be described as a single measurement. Notably, applying a unitary $\mathbf{U}$ on a state before performing a measurement $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ is the same thing as performing directly the measurement $\{\mathbf{M}_m\mathbf{U}\}_{m\in\mathcal{M}}$.

Remark 2.1.3. It is also important to see that the outcome of the measurement cannot be forced. Forcing the outcome of a measurement is called *postselection* and is very unlikely to be a physical process. While the standard complexity class associated with quantum computing is called BQP (Bounded-Error Quantum Polynomial-Time), the complexity class allowing postselection in quantum computing is called PostBQP and was proven [Aar05] to be equal to PP (Probabilistic Polynomial-Time), a class believed to be much wider than BQP.

Some special kinds of measurements will prove to be particularly useful. For instance, we define a *measurement in the computational basis* of a system $\mathcal{H}_n^{\otimes n}$ composed of n qubits as the measurement obtained with $\mathbf{M}_m \coloneqq |m\rangle \langle m|$, for $m \in \{0, \dots, 2^n - 1\}$. Note that $\mathbf{M}_m^\dagger \mathbf{M}_m = |m\rangle \langle m|$: therefore, when measuring a qubit $|\psi\rangle = a|0\rangle + b|0\rangle$ in the computational basis, we get outcome 0 with probability $\langle\psi| \mathbf{M}_0^\dagger \mathbf{M}_0 |\psi\rangle = \langle\psi|0\rangle\langle 0|\psi\rangle = |\langle 0|\psi\rangle|^2 = |a|^2$, and similarly we get outcome 1 with probability $|b|^2$. Note that this definition extends similarly for any orthonormal basis $\{|m\rangle\}_{m\in\mathcal{M}}$: in particular we talk about measurement in the *Hadamard* basis when the basis is made of combination of $|+\rangle$ and $|-\rangle$: this name comes from the fact that we can perform such a measurement by first applying Hadamard gates on each qubit and measuring in the computational basis.

It is also possible to define a *partial measurement* on a register, which measures only the qubits of a given register: If $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ is a measurement on $\mathcal{H}_B$, we can extend it to a measurement on the Hilbert space composed of multiple registers $\mathcal{H}_{A_1} \otimes \dots \mathcal{H}_{A_n} \otimes \mathcal{H}_B \otimes \mathcal{H}_{C_1} \otimes \dots \otimes \mathcal{H}_{C_l}$ by defining for all $m \in \mathcal{M}$, $\mathbf{M}'_m \coloneqq \mathbf{I} \otimes \dots \mathbf{I} \otimes \mathbf{M}_m \otimes \mathbf{I} \otimes \mathbf{I}$ (where the dimensions and number of the identity matrices match the number and dimensions of the corresponding registers). For instance, if we consider a state of the form $|\psi\rangle = a|0010\rangle + b|0011\rangle + c|0100\rangle$, if we measure the first two qubits in the computational basis we can get two possible outcomes: 00 and 01. We measure 00 with

probability:

$$\langle\psi|\,\mathbf{M}_{00}'^{\dagger}\mathbf{M}_{00}'\,|\psi\rangle = \langle\psi|\,(|00\rangle\langle 00|\otimes\mathbf{I})^{\dagger}(|00\rangle\langle 00|\otimes\mathbf{I})\,|\psi\rangle \tag{2.39}$$

$$= \langle\psi|\,((|00\rangle\langle 00|)^{\dagger}\otimes\mathbf{I}^{\dagger})(|00\rangle\langle 00|\otimes\mathbf{I})\,|\psi\rangle \tag{2.40}$$

$$= (\bar{a}\,\langle 0011| + \bar{b}\,\langle 0010|)(a\,|0011\rangle + b\,|0010\rangle) \tag{2.41}$$

$$= |a|^2 + |b|^2 \tag{2.42}$$

In that case, the state after the measurement will be $\frac{a|0011\rangle+b|0010\rangle}{\sqrt{|a|^2+|b|^2}}$ (note that we basically remove all terms not starting with 00 and renormalize the obtained state). Similarly, we measure 01 with probability $|c|^2$ and the post-measured state is $\frac{c}{|c|}\,|0100\rangle$ which is equal to $|0100\rangle$ when removing the global phase.

Another kind of measurements is common: *Positive Operator-Valued Measure (POVM)*. They are useful when knowing the post-measured state is not required, and they correspond to the most general operation we can perform to extract a classical information from a quantum state. A POVM is described by an indexed sequence $\{\mathbf{E}_m\}_{m\in\mathcal{M}}$ of operators such that each operator is positive (i.e. such that for all v and m, $\langle v|\,\mathbf{E}_m\,|v\rangle \geq 0$) and such that $\sum_m \mathbf{E}_m = \mathbf{I}$. One can turn a general measurement $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ into a POVM by simply defining $\mathbf{E}_m \coloneqq \mathbf{M}_m^{\dagger}\mathbf{M}_m$: this gives us that the probability of outputting outcome m when measuring $|\psi\rangle$ is $\langle\psi|\,\mathbf{E}_m\,|\psi\rangle$. Conversely, we can turn any POVM $\{\mathbf{E}_m\}_{m\in\mathcal{M}}$ into a general measurement by defining $\mathbf{M}_m \coloneqq \sqrt{\mathbf{E}_m}$ (which is well defined because the operator is positive, which implies by the spectral theorem that it is diagonalizable in an orthonormal basis with only positive eigenvalues: $\sqrt{\mathbf{E}_m}$ is then obtained by taking the square root of the diagonal entries).

This drives us to another kind of measurements: *projective measurements*. Sometimes a measurement is specified by a single Hermitian operator $\mathbf{M}$ (i.e. $\mathbf{M}^{\dagger} = \mathbf{M}$), called an *observable*. By the spectral theorem, $\mathbf{M}$ can be decomposed as $\mathbf{M} = \sum_m m\mathbf{P}_m$, where each $\mathbf{P}_m$ is a projector. We can then define our corresponding measurement as $\mathbf{M}_m \coloneqq \mathbf{P}_m$, and our probability of outputting outcome m when measuring a state $|\psi\rangle$ becomes

$$\langle\psi|\,\mathbf{M}_m^{\dagger}\mathbf{M}_m\,|\psi\rangle = \langle\psi|\,\mathbf{P}_m^{\dagger}\mathbf{P}_m\,|\psi\rangle \tag{2.43}$$

$$= \langle\psi|\,\mathbf{P}_m\,|\psi\rangle \tag{2.44}$$

where the last inequality applies because $\mathbf{P}_m$ are projectors.

2.1.3 Density Operator

Mixed quantum states. It is possible to generalize the notion of pure quantum states to *mixed quantum states*, in order to characterize distributions of quantum states and the discarding process. Indeed, we can consider the following scenario: a process outputs a pure quantum state according to the distribution $\{(p_i, |\psi_i\rangle)\}_i$, i.e. it outputs a state $|\psi_i\rangle$ with probability p_i. Then, let us consider an arbitrary measurement $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ performed on a state produced by this process. Then, for a fixed m, the probability of getting outcome m is equal to

$$p(m) = \sum_i p_i \langle\psi_i| \mathbf{M}_m^\dagger \mathbf{M}_m |\psi_i\rangle \tag{2.45}$$

$$= \sum_i p_i \operatorname{Tr}\left(\langle\psi_i| \mathbf{M}_m^\dagger \mathbf{M}_m |\psi_i\rangle\right) \tag{2.46}$$

$$= \operatorname{Tr}\left(\mathbf{M}_m^\dagger \mathbf{M}_m \left(\sum_i p_i |\psi_i\rangle \langle\psi_i|\right)\right) \tag{2.47}$$

We can see that $\rho \coloneqq \sum_i p_i |\psi_i\rangle \langle\psi_i| \in \mathscr{L}(\mathcal{H})$ (where $\mathscr{L}(\mathcal{H})$ is the set of linear operators from $\mathcal{H}$ to itself) is enough to characterize all the possible measurement outcomes for this distribution: ρ is called the *density operator*. Note that ρ has trace 1 and is positive (for all $|\phi\rangle$, $\langle\phi| \rho |\phi\rangle \geq 0$): Conversely, the spectral theorem tells us that any positive, trace 1 operator ρ can be written as a density operator $\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$ with $\sum_i p_i = 1$. It is therefore necessary and sufficient for a density operator to be positive with trace 1. The probability of getting outcome m can now be expressed only using the density operator ρ:

$$\boxed{p(m) = \operatorname{Tr}\left(\mathbf{M}_m^\dagger \mathbf{M}_m \rho\right)} \tag{2.48}$$

If we restrict ourselves to a POVM measurement $\{\mathbf{E}_m\}_{m\in\mathcal{M}}$, this simplifies further:

$$p(m) = \operatorname{Tr}(\mathbf{E}_m \rho) \tag{2.49}$$

As a side node, the following inequality can be useful when considering systems on multiple systems: one can easily prove—using standard linear algebra—that for any matrix A and B,

$$\operatorname{Tr}(A \otimes B) = \operatorname{Tr}(A) \otimes \operatorname{Tr}(B) \tag{2.50}$$

Note that density operators generalize pure quantum states since a pure quantum state $|\psi\rangle$ can be represented as the density operator $\rho \coloneqq |\psi\rangle \langle\psi|$. Conversely, a density operator ρ is said to be a *pure state* if there exists a state $|\psi\rangle \in \mathcal{H}$ such that $\rho = |\psi\rangle \langle\psi|$. Pure states can easily be recognized since their trace is equal to 1.

It is possible to translate all the previous axioms in the formalism of density operators: If we consider the state obtained after measuring outcome m, it is not hard to show that this gives rise to a new distribution of quantum states whose density operator is

$$\rho_m = \frac{\mathbf{M}_m \rho \mathbf{M}_m^\dagger}{\mathrm{Tr}(\mathbf{M}_m^\dagger \mathbf{M}_m \rho)} \tag{2.51}$$

Similarly, if we sample a state according to the distribution $\{(p_i, |\psi_i\rangle)\}_i$ (whose density operator is ρ) and apply on it a unitary $\mathbf{U}$, the resulting distribution is equivalent to $\{(p_i, \mathbf{U}|\psi_i\rangle)\}$ and its corresponding density operator can be expressed using solely ρ:

$$\rho' = \sum_i p_i \mathbf{U} |\psi_i\rangle \langle\psi_i| \mathbf{U}^\dagger \tag{2.52}$$

$$= \mathbf{U} \rho \mathbf{U}^\dagger \tag{2.53}$$

Discarding. Density operators can also be useful to represent a discarding process or a partial system. For instance, if we consider $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ a quantum state shared by two parties Alice and Bob, we can try to see what Alice can learn about $|\psi\rangle$ without having access to Bob's system (i.e. Bob's system is discarded). But first, let us introduce the partial trace Tr_B, which will prove to be useful later.

Let us consider two quantum systems A (held by Alice) and B (held by Bob), whose associated Hilbert spaces are respectively $\mathcal{H}_A$ and $\mathcal{H}_B$. Then, for any vectors $(|a_1\rangle, |a_2\rangle) \in \mathcal{H}_A^2$ and $(|b_1\rangle, |b_2\rangle) \in \mathcal{H}_B^2$, we define the *partial trace* over system B[5] as

$$\mathrm{Tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) \coloneqq |a_1\rangle\langle a_2| \,\mathrm{Tr}(|b_1\rangle\langle b_2|) = \langle b_2|b_1\rangle |a_1\rangle\langle a_2| \tag{2.54}$$

Then, we can extend this definition by linearity to any density operator ρ on the joint system $\mathcal{H}_A \otimes \mathcal{H}_B$, and we denote $\rho^A \coloneqq \mathrm{Tr}_B(\rho^{AB})$ the *reduced density operator* on system A obtained after *tracing out* the system B. Note that for any ρ^{AB} and $\mathbf{M}$, $\mathrm{Tr}(\mathrm{Tr}_B(\rho^{AB})) = \mathrm{Tr}(\rho^{AB})$ and $\mathrm{Tr}_B((\mathbf{M} \otimes \mathbf{I})\rho^{AB}) = \mathbf{M}\,\mathrm{Tr}_B(\rho^{AB}) = \mathbf{M}\rho^A$.

We can now see how to represent a discarding operation. As pointed out in Remark 2.1.2, any information Alice can learn about ρ^{AB} can be described by a single measurement $\{\mathbf{M}_m\}_{m\in\mathcal{M}}$ performed on her system, which is equivalent to performing a measurement $\{\mathbf{M}_m \otimes \mathbf{I}\}_{m\in\mathcal{M}}$ on the joint system $\mathcal{H}_A \otimes \mathcal{H}_B$. The probability of getting

[5]This definition can similarly be extended to partial trace over system A and to partial trace over larger systems.

outcome m is then equal to:

$$\begin{aligned} p(m) &= \mathrm{Tr}((\mathbf{M}_m \otimes \mathbf{I})^\dagger (\mathbf{M}_m \otimes \mathbf{I})\rho^{AB}) && (2.55)\\ &= \mathrm{Tr}(((\mathbf{M}_m^\dagger \mathbf{M}_m) \otimes \mathbf{I})\rho^{AB}) && (2.56)\\ &= \mathrm{Tr}(\mathrm{Tr}_B(((\mathbf{M}_m^\dagger \mathbf{M}_m) \otimes \mathbf{I})\rho^{AB})) && (2.57)\\ &= \mathrm{Tr}(\mathbf{M}_m^\dagger \mathbf{M}_m \, \mathrm{Tr}_B(\rho^{AB})) && (2.58)\\ &= \mathrm{Tr}(\mathbf{M}_m^\dagger \mathbf{M}_m \rho^A) && (2.59) \end{aligned}$$

Therefore, when discarding a part of a system (here the system B), the remaining part of the system can be represented by the (reduced) density operator $\rho^A = \mathrm{Tr}_B(\rho^{AB})$.

Quantum Channels. With this new representation, we can see what is the most general operations one can perform on a density operator. Such an operator $\mathcal{E} : \mathscr{L}(\mathcal{H}_A) \to \mathscr{L}(\mathcal{H}_B)$, called *Completely-Positive Trace-Preserving map* (*CPTP map* for short), must be:

- *Linear*: For any probability distribution $\{p_i\}$ and for any density operators $\{\rho_i\}$

$$\mathcal{E}\left(\sum_i p_i \rho_i\right) = \sum_i p_i \mathcal{E}(\rho_i) \tag{2.60}$$

- *Trace Preserving*: Since a density operator must have norm 1, we expect to have for any positive operator ρ, $\mathrm{Tr}(\mathcal{E}(\rho)) = \mathrm{Tr}(\rho)$. This requirement is sometimes weakened ($\mathcal{E}$ is then said to be a *quantum operation*), and $\mathcal{E}$ is then asked to be a non-trace-increasing map, i.e. for any (normalized) density operator ρ, $0 \leq \mathrm{Tr}(\mathcal{E}(\rho)) \leq 1$. In this case, the quantity $\mathrm{Tr}(\mathcal{E}(\rho))$ corresponds to the probability of applying $\mathcal{E}$, and the post-measured state must be renormalized into $\mathcal{E}(\rho)/\mathrm{Tr}(\mathcal{E}(\rho))$.
- *Completely-positive*: For any positive ρ, $\mathcal{E}(\rho)$ is positive. Moreover, for any Hilbert space C and any positive operator ρ: $\mathscr{L}(\mathcal{H}_A \otimes \mathcal{H}_C) \to \mathscr{L}(\mathcal{H}_B \otimes \mathcal{H}_C)$, $(\mathcal{E} \otimes \mathbf{I})(\rho)$ is positive. This is required to ensure $\mathcal{E}$ maps density operators to density operators.

The Kraus decomposition is useful to characterize all CPTP maps and quantum operations:

Theorem 2.1.4 (Choi-Kraus' theorem)**.** *A map $\mathcal{E} : \mathscr{L}(\mathcal{H}_A) \to \mathscr{L}(\mathcal{H}_B)$ is a CPTP map (respectively a quantum operation) if and only if there exist some operators $\{\mathbf{E}_i\}$ such that $\sum_i \mathbf{E}_i^\dagger \mathbf{E}_i = \mathbf{I}$ (respectively $\sum_i \mathbf{E}_i^\dagger \mathbf{E}_i \leq \mathbf{I}$) and if for any $\rho \in \mathscr{L}(\mathcal{H}_A)$:*

$$\mathcal{E}(\rho) = \sum_i \mathbf{E}_i \rho \mathbf{E}_i^\dagger \tag{2.61}$$

This is known as the Kraus decomposition of $\mathcal{E}$.

CPTP maps can be shown to be isometries:

$$\mathrm{Tr}(\mathcal{E}(\rho)\mathcal{E}(\sigma)) = \mathrm{Tr}(\rho\sigma) \tag{2.62}$$

Note that we can also generalize CPTP maps to processes having both a classical and a quantum output as introduced in [DL70]:

Definition 2.1.5 (Quantum Instrument)**.** *A map* $\Lambda : \mathbb{C}^{n\times n} \to \{0,1\}^{m_1} \times \mathbb{C}^{m_2\times m_2}$ *is said to be a* quantum instrument *if there exists a collection* $\{\mathcal{E}_y\}_{y\in\{0,1\}^{m_1}}$ *of trace-non-increasing completely positive maps such that the sum is trace-preserving (i.e. for any positive operator* ρ*,* $\sum_y \mathcal{E}_y(\rho) = \mathrm{Tr}(\rho)$*), and, if we define* $\rho_y = \frac{\mathcal{E}_y(\rho)}{\mathrm{Tr}(\mathcal{E}_y(\rho))}$*, then* $\Pr\left[\Lambda(\rho) = (y, \rho_y)\right] = \mathrm{Tr}(\mathcal{E}_y(\rho))$*.*

2.2 Graphical representation of Quantum Operations

2.2.1 Quantum Circuits

It is frequent to create a quantum state or run an algorithm by applying several elementary operations, usually corresponding to single or two-qubits unitaries and measurements. Quantum circuits are commonly used to represent these operations, mimicking classical circuits. A quantum circuit is made of multiple wires, each wire representing typically a qubit. Some gates are drawn on these wires and are applied on the corresponding qubit(s), the leftmost gates being applied first.

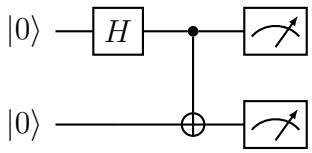
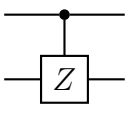


Figure 2.4: A quantum circuit.

For instance, a quantum circuit is drawn in Figure 2.4. In this circuit, the initial state is $|00\rangle$. Then, we apply the Hadamard gate $\mathbf{H}$ on the first qubit (i.e. we apply $\mathbf{H}\otimes\mathbf{I}$ on the 2-qubit system), giving the state $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$. The next step is to apply a **CNOT** gate (not to be confused with the $\wedge\mathbf{Z}$ gate, which is symmetric and represented by  or 

), which produces the state

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.63}$$

This state is very famous and is called a *Bell pair* (also known as EPR pair). Finally, we measure this Bell pair in the computational basis, leading to the outcome 00 with probability $\frac{1}{2}$ and to the outcome 11 with probability $\frac{1}{2}$.

Similar to classical circuits, it is possible to define a *universal set of gates*, which is a minimal set of gates that guarantees that any computation can be performed using only these gates. This is possible when using the set

$$\{\mathbf{H}, \mathbf{CNOT}\} \bigcup \{\mathbf{R}_z(\theta) : \theta \in [0; 2\pi)\} \tag{2.64}$$

However, this set is not really practical since it is continuous and hard to implement in fault-tolerant quantum computing. Fortunately, as proven by Solovay and Kitaev [Kit97, DN06] the following famous and simpler set can be used to approximate any unitary up to an arbitrarily small precision:

$$\{\mathbf{H}, \mathbf{CNOT}, \mathbf{T}\} \tag{2.65}$$

2.2.2 Diagrammatic Reasoning on Quantum Computing: the ZX-Calculus

While quantum circuits are handy to represent operations on quantum states, they are not a substitute to algebra: in order to compute the final state obtained after running a circuit, it is typically necessary to do quite heavy computations, potentially involving large vectors. Moreover, it can be difficult to gain much intuition when using raw algebra to perform computations. On the other hand, the ZX-Calculus—initially introduced by Bob Coecke and Ross Duncan [CD08]—provides a way to graphically represent a circuit or a matrix using a diagram with a graph-like structure: computations can be performed directly by rewriting parts of the graph. It has numerous advantages over plain linear algebra: computations are often simpler, more elegant, less error-prone, and it is easier to gain an intuition on the underlying computation. Again, it is not mandatory to know the ZX-Calculus to understand this thesis—a careful reader could check any computation done in the ZX-Calculus using standard linear algebra—but the ZX-Calculus will help in multiple places. As already mentioned, we redirect the reader interested in the ZX-calculus to this review [vdWet20] and to the "Dodo book" [CK17] which offers a very interesting and general approach to diagrammatic reasoning (I would have loved to introduce the no-cloning and no-signaling principles using the approach formulated in [CK17], but I refrained for the sake of brevity).

ZX-diagrams. The basic structure in the ZX-calculus is an open graph known as a ZX-diagram, with some input and output wires. The time flows from left to right, so the inputs are entering the nodes from the left, and the outputs are leaving the nodes from the right. Each graph has an associated interpretation, denoted $[\![\cdot]\!]$, which is a complex matrix. Here are the generators of a ZX-diagram, together with their interpretation:

- Empty diagram: $[\![\ ]\!] = \begin{pmatrix} 1 \end{pmatrix}$
- Wire (identity): $[\![—]\!] = |0\rangle\langle 0| + |1\rangle\langle 1|$
- Green spider: $[\![n \overset{\alpha}{\bullet} m]\!] = |0\rangle^{\otimes m}\langle 0|^{\otimes n} + e^{i\alpha}|1\rangle^{\otimes m}\langle 1|^{\otimes n}$
- Red spider: $[\![n \overset{\alpha}{\circ} m]\!] = |+\rangle^{\otimes m}\langle +|^{\otimes n} + e^{i\alpha}|-\rangle^{\otimes m}\langle -|^{\otimes n}$
- Hadamard[6]: $[\![—\square—]\!] = |+\rangle\langle 0| + |-\rangle\langle 1|$
- Swap: $[\![\times]\!] = |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11|$
- Bell state: $[\![\cup]\!] = |00\rangle + |11\rangle \propto \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Bell projection: $[\![\cap]\!] = \langle 00| + \langle 11| \propto \frac{1}{\sqrt{2}}(\langle 00| + \langle 11|)$

Moreover, when $\alpha = 0$, we can omit the α in the spiders. Similarly to quantum circuits, it is then possible to compose multiple ZX-diagrams in two ways, as shown in Figure 2.5:

- Sequentially, by connecting two ZX-diagrams horizontally and making sure that the output wires of the first diagram D_1 are connected to the input wires of the second diagram D_2. The resulting interpretation is the product of the matrices: $[\![D_1 D_2]\!] \coloneqq [\![D_2]\!][\![D_1]\!]$.
- In parallel, by stacking two ZX-diagrams vertically. The resulting interpretation is the tensor product of the matrices: $\left[\!\!\left[\begin{matrix} D_1 \\ D_2 \end{matrix} \right]\!\!\right] \coloneqq [\![D_1]\!] \otimes [\![D_2]\!]$.

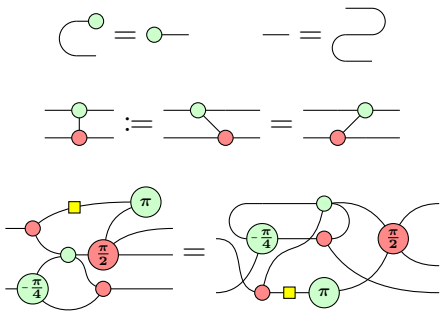


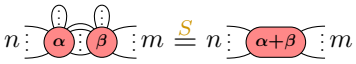
Figure 2.5: Example of ZX-diagrams, and illustration of the *only connectivity matters* principle which provides equalities between diagrams

[6]Hadamard is introduced here as a generator, but it is mostly a "syntaxic sugar": it can be made from green and red spiders as shown later in the rules

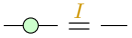
Rules for Clifford ZX-calculus. In order to obtain the interpretation of a given diagram, it is of course possible to follow an approach similar to the one used in quantum circuits, i.e. replace each generator by its interpretation and manually compute the resulting matrix. However, in many cases it is more convenient to use a set of rules to rewrite the ZX-diagram into a simpler ZX-diagram. The first rule of the ZX-calculus is that *only connectivity matters*, i.e. two diagrams have the same interpretation if their nodes are connected in the same way (regardless of whether the wire goes out or in the node), even if the nodes are moved across the plane and the wires are bended. The only requirement is that the order and number of inputs and outputs of the *whole ZX-diagram* must be preserved. This principle is illustrated in Figure 2.5.

The other rules are presented below. Note that these rules are complete for a fragment of the circuits called the Clifford fragment, but additional rules (which are not useful in this thesis) can be used to obtain a complete generic ZX-calculus. Note that all these rules stay true when read from right to left, when colors are inverted and when phases are negated. The dots means "0, 1 or more wires".

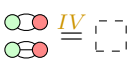
- **Only connectivity matters (C)**: See explanation above and illustration in Figure 2.5.
- **Spider rule and loop rule (S)**: It is possible to merge any connected spider with the same color, even if they have some loops by summing angles modulo 2π.



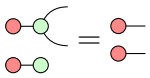
- **Identity rule (I)**: Any node with no angle (i.e. angle 0) can be removed.



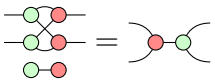
- **Scalar rules (IV)**: $\sqrt{2} \times \frac{1}{\sqrt{2}} = 1$.



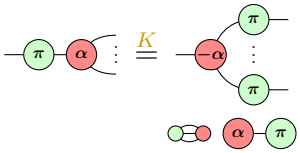
- **Copy rule (CP)**:



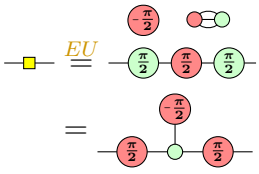
- **Bialgebra rule (B)**: Illustrates the fact that the copy operation "commutes" with XOR.



- **π-commutation and copy (K)**

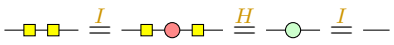


- **Euler decomposition (EU)**: The Hadamard gate can be decomposed into multiple rotations on the block sphere.

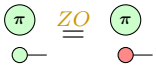


- **Hadamard (H)**: The Hadamard gate swaps the basis.

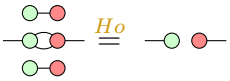
In particular:



- **Zero (ZO)**: A node with no input/output with an angle π is equal to zero. In that case, basically all inequalities become true, even changing the color of a node.



- **Hopf law**: (derivable from the above rules)



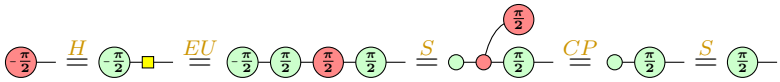
Scalars. Note that these rules include some *scalars* like $[\![\bullet\!-\!\circ]\!] = \sqrt{2}$, which are disconnected sub-graphs without any inputs nor outputs (and therefore whose interpretation is a matrix of size 1×1). Scalars can be freely moved across the diagram, and multiply the whole interpreted matrix. It is also often convenient to get rid of the scalars: all equalities stay true up to a global phase and a normalization factor, factor which is usually easy to recover at the end of the computation if needed since the norm must stay unchanged (and a quantum state always have norm 1). We will therefore often omit them, but we included the scalars in the rules for completeness. We list here the interpretation of common scalars (these equations are also true when the colors are inverted):

$$\circ = 2 \qquad (\pi) = 0 \qquad (\alpha) = 1 + e^{i\alpha}$$

$$\alpha\!-\!\circ = \sqrt{2} \qquad \alpha\!-\!\pi = \sqrt{2}e^{i\alpha} \qquad \alpha\!=\!\circ = \frac{1}{\sqrt{2}} \tag{2.66}$$

$$\alpha\!-\!\beta = \frac{1}{\sqrt{2}}(1 + e^{i\alpha} + e^{i\beta} - e^{i(\alpha+\beta)})$$

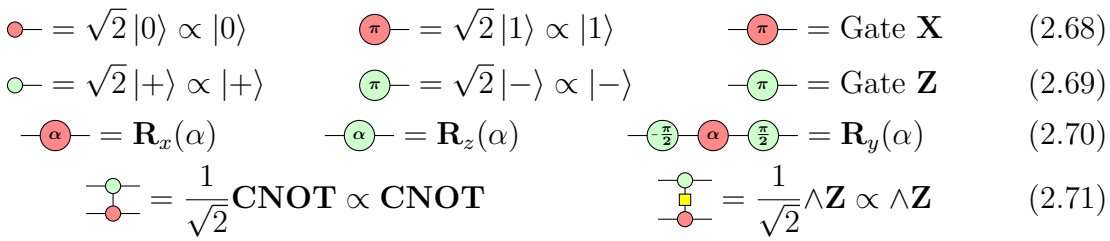
In the following, we will remove all non-null scalars. For instance, when removing scalars we can derive what is called the Y-state identity $-\frac{\pi}{2}- = \frac{\pi}{2}-$ (which is true up to a global phase), which is useful for instance to prove the equivalence between the two versions of the EU rule:



(2.67)

Usual states and gates. One can turn a quantum circuit into a ZX-diagram using the following table. A careful reader can check the correctness of the gates either by computing the interpretation of each gate, or by evaluating them on each basis vector

using the ZX rules presented above.



(2.68) (2.69) (2.70) (2.71)

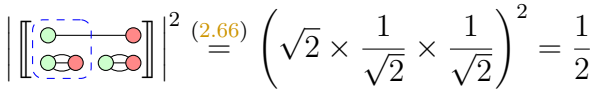
Measurements. As it, the ZX-calculus does not provide a way to add a measurement, and it is only possible to use projectors like (the scalar is required for normalization but will be dropped soon):

(2.72)

While these "gates" are technically not physical because they look like postselection (i.e. it forces the outcome of a measurement), they can be seen as a mathematical tool to compute the probability of measuring a given outcome. For instance, using standard linear algebra, when measuring the state $|+\rangle$, the probability of measuring 0 is given by

$$|\langle 0|+\rangle|^2 = \left|\langle 0| \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2} \tag{2.73}$$

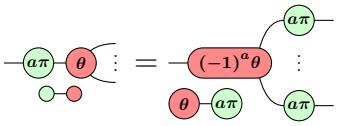
Similarly, one can compute in ZX-calculus



(2.74)

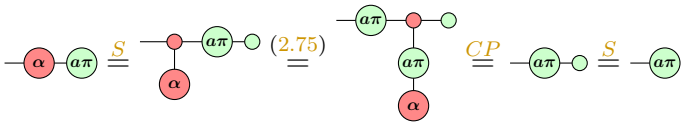
where the blue dashed box corresponds to a $|+\rangle$ state with the appropriate normalization scalar $\frac{1}{\sqrt{2}}$ and the rest is the projector.

However, this is cumbersome to use for two reasons: first the normalization scalars are a bit annoying to maintain, and secondly we need to do these computation multiple times, one for each measurement outcome: the advantage over linear algebra is not clear. The later problem can easily be solved in multiple ways. One solution is to introduce binary variables $a \in \{0, 1\}$ in order to code the outcome of the measurement. A measurement in the computational basis would then become (up to the normalization factor) —(aπ), with $a = 0$ when the outcome is 0 and $a = 1$ when the outcome is 1 (a green spider would represent a measurement in the $\{|+\rangle, |-\rangle\}$ basis). The ZX rules are still correct ($a\pi$ can be seen as an arbitrary angle), and the π-commutation law can be improved:



(2.75)

We can also prove that (up to scalars) $-\alpha-a\pi- = -a\pi-$:

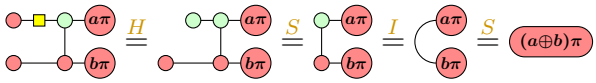


(2.76)

That way, instead of maintaining one diagram per outcome, we can maintain a single diagram, simplifying the computations.

To avoid using normalization scalars, it is often enough to use the fact that quantum states have norm 1 to recover the normalization factor at the end of the computation. The only problematic case is if we want to know the probability of measuring a given outcome. In that case, if we simplified the computation using the variables $a\pi$ and the above rules, it is possible to recover the probability at the end of each measurement outcome using the fact the probabilities must sum to one.

Example. We can now compute as before the outcome given by the circuit pictured in Figure 2.4, after replacing each part of the circuit by the appropriate diagram:



(2.77)

We see on the fourth diagram that the part of the circuit before the measurement produces a Bell state. Then, to know the outcome of the measurement, we use the fact that when $c \in \{0,1\}$, $[\![c\pi]\!] = 1 - c$. Therefore $[\![(a\oplus b)\pi]\!] = 1 - a \oplus b$. So the probability of getting outcome (a,b) when $a \oplus b = 1$ (i.e. $a \neq b$) is 0, and the probability of getting outcome $(0,0)$ is the same as the probability of getting $(1,1)$, i.e. with probability $1/2$ we get outcome $(0,0)$ and with probability $1/2$ we get outcome $(1,1)$.

2.3 Well-Known Quantum Protocols and Properties

2.3.1 No-cloning principle

One of the most stunning and basic property of quantum mechanics is that it is impossible to copy perfectly an unknown qubit. More specifically:

Theorem 2.3.1 (No-cloning principle)**.** *Let $\mathcal{H}$ be a Hilbert space of dimension $n \geq 2$. There exists no CPTP map $\mathcal{E} : \mathscr{L}(\mathcal{H}) \to \mathscr{L}(\mathcal{H}\otimes\mathcal{H})$ such that for any state ρ, $\mathcal{E}(\rho) = \rho\otimes\rho$ (even if we restrict ρ to pure states).*

Proof. Let $(\rho, \sigma) \in \mathscr{L}(\mathcal{H})^2$, and $\{\mathbf{E}_i\}_i$ be the Kraus decomposition of $\mathcal{E}$. By assumption, $\mathcal{E}(\rho) = \rho \otimes \rho$ and $\mathcal{E}(\sigma) = \sigma \otimes \sigma$. We can now compute $\mathrm{Tr}(\mathcal{E}(\rho)^\dagger \mathcal{E}(\sigma))$ in two different ways using these equations. Either using the fact that $\mathcal{E}$ is an isometry:

$$\mathrm{Tr}(\mathcal{E}(\rho)^\dagger \mathcal{E}(\sigma)) \overset{(2.62)}{=} \mathrm{Tr}(\rho^\dagger \sigma) \tag{2.78}$$

Or using the fact that $\mathcal{E}$ is a copy map:

$$\begin{aligned} \mathrm{Tr}(\mathcal{E}(\rho)^\dagger \mathcal{E}(\sigma)) &= \mathrm{Tr}((\rho \otimes \rho)^\dagger (\sigma \otimes \sigma)) && (2.79)\\ &= \mathrm{Tr}(\rho^\dagger \sigma \otimes \rho^\dagger \sigma) && (2.80)\\ &\overset{(2.50)}{=} \mathrm{Tr}(\rho^\dagger \sigma)^2 && (2.81) \end{aligned}$$

Therefore, $\mathrm{Tr}(\rho^\dagger \sigma) = \mathrm{Tr}(\rho^\dagger \sigma)^2$, i.e. $\mathrm{Tr}(\rho^\dagger \sigma)$ is either 0 or 1 (i.e. ρ and σ are either equal or orthogonal). Taking for instance the quantum states $|0\rangle$ and $|+\rangle$ for which $\mathrm{Tr}(|0\rangle\langle 0||+\rangle\langle +|) = \mathrm{Tr}(\langle 0| + |)\rangle^2 = \frac{1}{2} \notin \{0, 1\}$, we conclude that there exist no CPTP map that can perfectly copy $|0\rangle$ and $|+\rangle$. □

In particular, it is also impossible to distinguish perfectly two non-orthogonal states. The trace distance is of great importance to quantify this indistinguishability:

Definition 2.3.2 (Trace Distance)**.** *The trace distance between two density operators ρ and σ is defined as:*

$$D_{TD}(\rho, \sigma) \coloneqq \frac{1}{2} \mathrm{Tr}(|\rho - \sigma|) \tag{2.82}$$

where $|A| \coloneqq \sqrt{A^\dagger A}$ can be computed by diagonalizing $A^\dagger A$ (it is always possible by the spectral theorem) and by applying a square root on all eigenvalues.

The trace distance is a measure on how hard it is to distinguish two quantum states as it can be shown that

$$D_{TD}(\rho, \sigma) = \max_{0 \leq P \leq I} \mathrm{Tr}(P(\sigma - \rho)) \tag{2.83}$$

(the proof basically diagonalize $\rho - \sigma$ into a positive and a negative part, see for instance [NC10, Sec 9.2.1]). Notably, it is equal to the advantage in distinguishing if a state is equal to ρ or σ (this is a particular case of the Helstrom bound [Hel69, Hol73, BK15]).

2.3.2 No-signaling principle

Because the state of a bipartite system instantaneously changes when one party performs a local operation, one may want to use this to communicate faster than light. However, this is impossible, and known as the *no-signaling principle*, or *no-communication theorem*.

To see that, let us imagine that two parties A and B share a bipartite state ρ, decomposed into $\rho = \sum_j \mathbf{T}_j \otimes \mathbf{S}_j$. Then, if the first party A applies locally a CPTP map $\mathcal{E}(\sigma) = \sum_i \mathbf{E}_i \sigma \mathbf{E}_i^\dagger$, the view of party B is:

$$\begin{aligned}
\mathrm{Tr}_A\left(\sum_i (\mathbf{E}_i \otimes \mathbf{I})\rho(\mathbf{E}_i \otimes \mathbf{I})^\dagger\right) &= \mathrm{Tr}_A\left(\sum_{i,j} (\mathbf{E}_i \otimes \mathbf{I})(\mathbf{T}_j \otimes \mathbf{S}_j)(\mathbf{E}_i \otimes \mathbf{I})^\dagger\right) && (2.84)\\
&= \sum_{i,j} \mathrm{Tr}_A\left(\left(\mathbf{E}_i \mathbf{T}_j \mathbf{E}_i^\dagger\right) \otimes \mathbf{S}_j\right) && (2.85)\\
&= \sum_{i,j} \mathrm{Tr}\left(\mathbf{E}_i \mathbf{T}_j \mathbf{E}_i^\dagger\right) \mathbf{S}_j && (2.86)\\
&= \sum_{i,j} \mathrm{Tr}\left(\mathbf{T}_j \mathbf{E}_i^\dagger \mathbf{E}_i\right) \mathbf{S}_j && (2.87)\\
&= \sum_{i,j} \mathrm{Tr}\left(\mathbf{T}_j\right) \mathbf{S}_j && (2.88)\\
&= \mathrm{Tr}_A(\rho) && (2.89)
\end{aligned}$$

Said differently, the view of B does not depend on $\mathcal{E}$ at all, which means that no-communication is possible between A and B by applying only local operations.

2.3.3 Quantum Unitaries of Classical Functions

Given a classical function $f\colon \{0,1\}^n \to \{0,1\}^m$ described by a classical circuit, it is always possible to efficiently derive a quantum circuit computing this function in superposition, i.e. a unitary $\mathbf{U}_f$ on $\mathcal{H}_{2^n} \otimes \mathcal{H}_{2^m}$ such that for any $x \in \{0,1\}^n$ and $b \in \{0,1\}^m$:

$$\mathbf{U}_f(|x\rangle\,|b\rangle) \coloneqq |x\rangle\,|b \oplus f(x)\rangle \tag{2.90}$$

where $\oplus$ is a bitwise XOR operation (sum modulo 2). Note that in general it is not possible to obtain for any f a unitary $\mathbf{U}'_f$ such that $\mathbf{U}'_f\,|x\rangle = |f(x)\rangle$ because a unitary must be invertible, which is also the reason of the presence of the XOR. However, in this thesis we will be particularly interested by the particular case where $b = 0$:

$$\mathbf{U}_f(|x\rangle\,|0\rangle) = |x\rangle\,|f(x)\rangle \tag{2.91}$$

It is frequent to group qubits depending on their role, and such a group is called a *register*. For instance, here we have two registers, one with n qubits containing $|x\rangle$, and one with m qubits containing $|f(x)\rangle$.

The equation Eq. (2.90) can be obtained by using the fact that the classical Toffoli gate $(a, b, c) \mapsto (a, b, c \oplus ab)$ is universal when using an auxiliary register, i.e. it is always possible to create a circuit using only Toffoli gates computing

$$(x, 0 \dots 0, 1 \dots 1) \mapsto (x, f(x), g(x)) \tag{2.92}$$

where $g(x)$ is an extra garbage output. Then, the Toffoli gate (and therefore the whole circuit) can be implemented quantumly, using for instance the construction given in [NC10, p. 182], leading to a unitary $\tilde{\mathbf{U}}_f |x\rangle |0 \dots 0\rangle |1 \dots 1\rangle = |x\rangle |f(x)\rangle |g(x)\rangle$. The trick to remove the $|g(x)\rangle$ part is to first apply **CNOT** gates to copy $f(x)$ on a fourth register:

$$|x\rangle |f(x)\rangle |g(x)\rangle |b\rangle \mapsto |x\rangle |f(x)\rangle |g(x)\rangle |b \oplus f(x)\rangle \tag{2.93}$$

and then applying $\tilde{\mathbf{U}}_f^\dagger$ to invert what was done on the first 3 registers, giving back:

$$|x\rangle |0 \dots 0\rangle |1 \dots 1\rangle |b \oplus f(x)\rangle \tag{2.94}$$

The second and third registers, called *auxiliary registers* (or sometimes *ancilla*) are therefore left untouched by the whole operation and are usually omitted. Therefore, the whole process turns $|x\rangle |b\rangle$ into $|x\rangle |b \oplus f(x)\rangle$.

2.3.4 Entanglement

One of the main features of quantum information theory is *entanglement*, which states that to describe a quantum state, it is *not* enough to describe each part separately. More precisely, a bipartite state $|\psi\rangle$ (i.e. a state belonging to a Hilbert space composed of two systems $\mathcal{H}_A \otimes \mathcal{H}_B$) is said to be *entangled* if it *cannot* be written as a tensor product of two states, i.e. if for any $|\phi_A\rangle \in \mathcal{H}_A$ and $|\phi_B\rangle \in \mathcal{H}_B$, we have

$$|\psi\rangle \neq |\phi_A\rangle |\phi_B\rangle \tag{2.95}$$

The Bell states

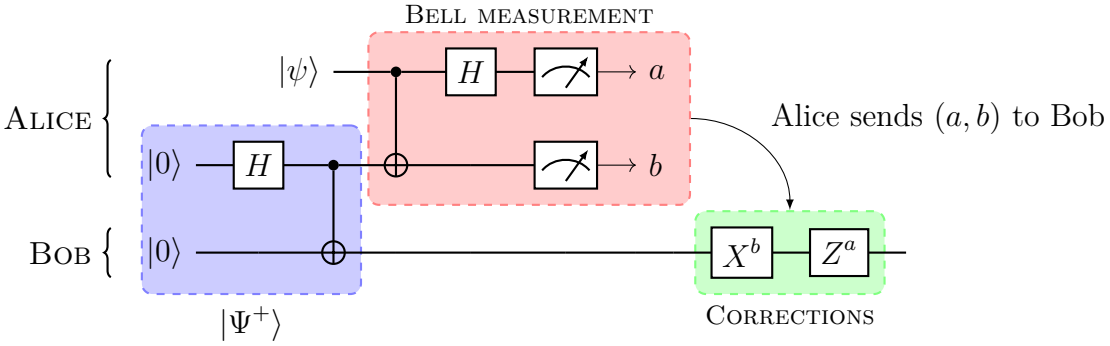
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{2.96}$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{2.97}$$

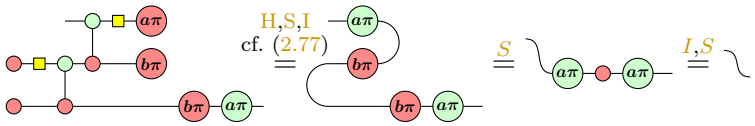
are famous entangled state and form a basis of $\mathcal{H}_{2^2}$. Entanglement is a crucial quantum property, and turns out to be useful in many protocols as we will see.

2.3.5 Quantum Teleportation

Quantum teleportation is a quantum protocol between two parties, say *Alice* and *Bob*, pre-sharing a Bell state. The goal of Alice is to send a qubit to Bob by communicating only classically. It is surprising that such a protocol exists because a qubit can potentially require an infinite amount of classical bits to be fully described, and measuring a quantum state usually disturbs (if not destroys) the quantum state. In contrast, in this protocol Alice only needs to send two bits of information to Bob.



The ZX-calculus provides a simple and elegant proof of correctness, where the flow of information is well visible (note that the gate $\mathbf{Z}^a$ is translated into —(aπ)—):



(2.98)

In the second diagram, it is clear that Alice and Bob share a Bell state and that the operation performed by Bob is a Bell measurement, i.e. a projection on one of the four Bell states. At the end, we obtain as expected an identity wire, so any qubit (even arbitrarily entangled into a larger systems) given by Alice will be transfered to Bob.

2.3.6 Measurement-Based Quantum Computing

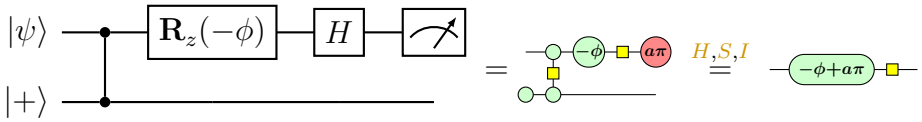
In the next two sections, we will see how a weak quantum client can do blind quantum computing using the Universal Blind Quantum Computing (UBQC) protocol. The take-home message of these sections is that it is possible to do blind quantum computing as soon as the client can send many quantum states $|+_\theta\rangle$, where θ is sampled uniformly at random over $\{0, \frac{\pi}{4} \dots \frac{7\pi}{4}\}$. The QFactory protocol—which is one of the main contributions of this thesis—will fake this quantum channel using a purely classical channel, and will be modularly inserted into the UBQC protocol. For that reason, a reader interested by the QFactory but not by its applications can skip these sections.

First approach to MBQC. *Measurement-Based Quantum Computing* (MBQC, it is also called *one-way quantum computer*) [RB01, RB02, RBB03, DK06] is a method used to perform computations: instead of doing a computation by applying appropriate unitaries, in MBQC the computation is done by doing appropriate measurements. This will prove to be extremely useful later to do blind quantum computing [BFK09], as we will see how to hide a measurement. In MBQC, instead of using few qubits on which we apply many operations, we use many qubits but apply few operations on them. We typically use $\wedge\mathbf{Z}$ gates and measurements with angle θ, i.e. destructive measurements in the $\{|+_\phi\rangle, |+_{-\phi}\rangle\}$ basis (when measuring $|+_\phi\rangle$ we say that the outcome is 0 and 1 otherwise). We will soon see the precise definition of an MBQC computation, but for now it is enough to say that an MBQC computation is performed in 3 stages:

- During a first stage the input qubits and some auxiliary $|+\rangle$ qubits will be entangled using $\wedge\mathbf{Z}$ gates: this will form an entangled *graph state* $G = (V, E)$, where $(i, j) \in E$ if a gate $\wedge\mathbf{Z}$ was applied between qubits i and j.
- During a second stage, some qubits i will be measured with angle ϕ_i' (note that ϕ_i' will depend on the previous measurement outcomes in order to obtain determinism). The non-measured qubits will be the *output qubits*.
- During the first stage, we will apply some corrections on the output qubits to correct non-determinism coming from measurements.

Intuitively, when measuring a qubit in an MBQC computation, this qubit will be "teleported" on the neighboring qubits and slightly modified depending on the value of the measurement angle. By repeating this process we will be able to perform any computation.

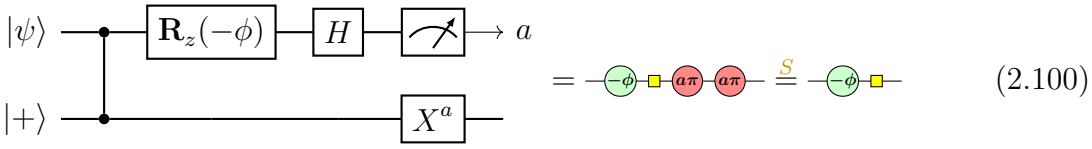
Forcing deterministic output. Before giving the formal MBQC definition and algorithm, we will start with three small examples that will come in handy to understand how MBQC corrects the non-determinism inherent to quantum measurements. First consider the following MBQC-like circuit (composed of a $\wedge\mathbf{Z}$ gate and a measurement with angle ϕ whose outcome is a) applied on a qubit $|\psi\rangle$:



(2.99)

We see that this circuit is outputting $\mathbf{HR}_z(-\phi + a\pi)\,|\psi\rangle$. Note that this gate is not (yet) deterministic, as it depends on the outcome a. Because $-\phi+a\pi \overset{S}{=} -\phi\ a\pi \overset{H}{=} -\phi\ a\pi \overset{H}{=} -\phi\ a\pi$, it is easy to see that one can get rid of the $a\pi$ node by

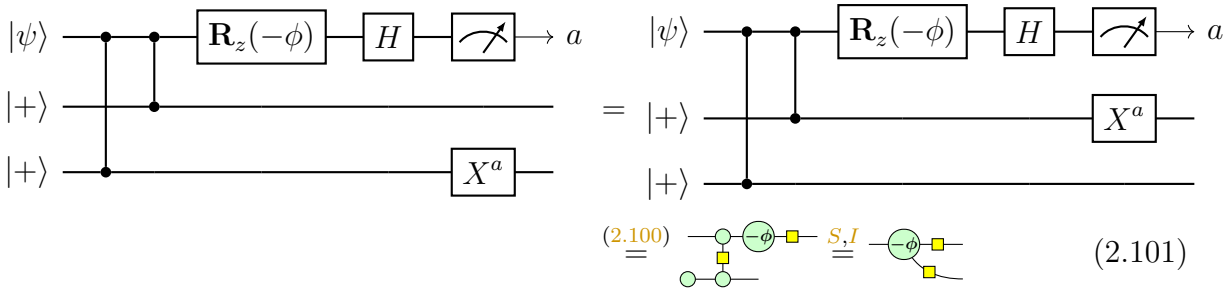
applying a gate $\mathbf{X}^a$ (called a *correction*) on the second qubit after the measurement:



(2.100)

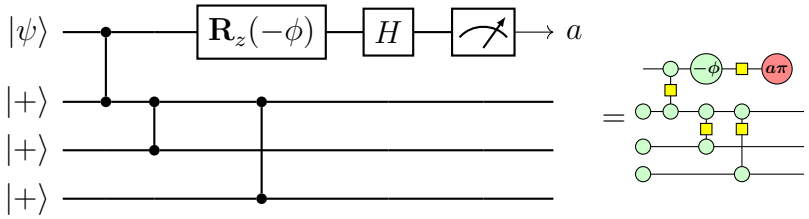
The resulting applied gate is $\mathbf{HR}_z(-\phi)$, which is now deterministic.

Now, let us see how determinism can be enforced on larger graph states. First, if we increase the "width" of the graph, i.e. if more than one $|+\rangle$ is entangled with $|\psi\rangle$, we can apply the $\mathbf{X}^a$ correction on any neighbouring $|+\rangle$ of our choice, as soon as we apply this correction only once (we can see that by reordering wires and commuting $\wedge\mathbf{Z}$ gates):

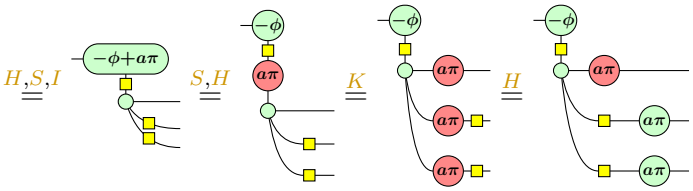


(2.101)

We can also increase the "depth" of the graph by chaining $\wedge\mathbf{Z}$ gates:



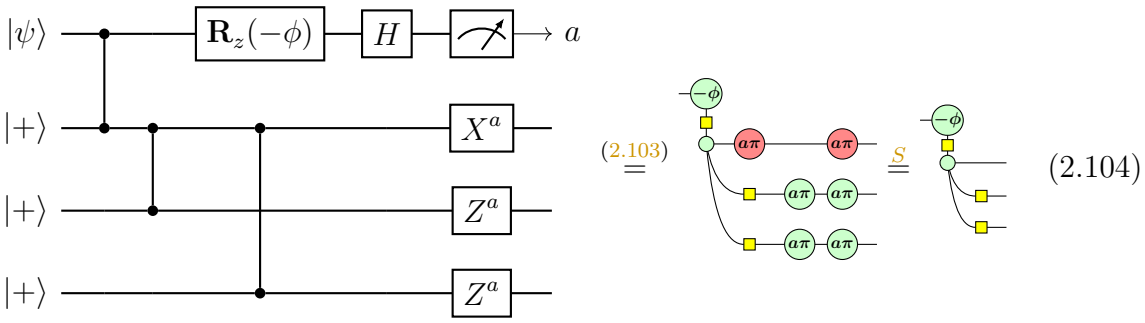
(2.102)



(2.103)

This shows that in order to come back to a deterministic computation, one must perform an $\mathbf{X}^a$ gate on the first qubit (i.e. the direct neighbor), and a $\mathbf{Z}^a$ gate on the second and

third qubits (i.e. the neighbors of the neighbor):

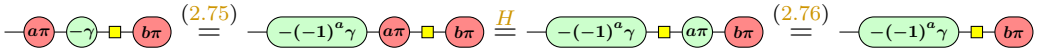


(2.104)

One may want to try to consider longer sequences of $\wedge\mathbf{Z}$, but it is not needed since the corrections will not propagate further due to the fact that $\wedge\mathbf{Z}$ commutes with $\mathbf{Z}$.

By generalizing these three examples, we can see that in order to have a deterministic output when measuring a qubit i, we need to apply an $\mathbf{X}^a$ correction on *one* neighbor qubit j (this choice can be expressed using a function $f(i) \coloneqq j$), and a $\mathbf{Z}^a$ correction on *all* the neighbors k of j (except i). Of course, this works only if all the neighbors k have not yet been measured, otherwise we will not be able to apply the correction. If we can find an order of measurement and f such that these conditions are met (we say that G has a *flow*), then we can obtain a deterministic computation.

Merging corrections with measurements. The above corrections were applied directly using a gate. It was fine because all the qubits were output qubits. However, if the qubits are supposed to be measured after, we do not want to apply these gates directly (first for efficiency reasons, but also because for cryptographic applications a will be kept secret). So we prefer to update instead the measurement angle: if a qubit gets an $\mathbf{X}^a$ correction and is then supposed to be measured with angle γ, we can instead measure it with angle $(-1)^a\gamma$:

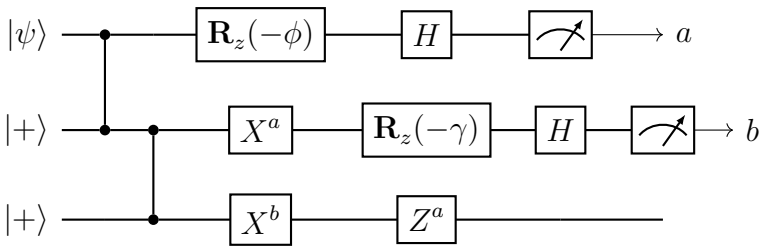


(2.105)

Similarly, if the correction was $\mathbf{Z}^a$, we can simply perform a measurement with angle $\gamma + a\pi$:

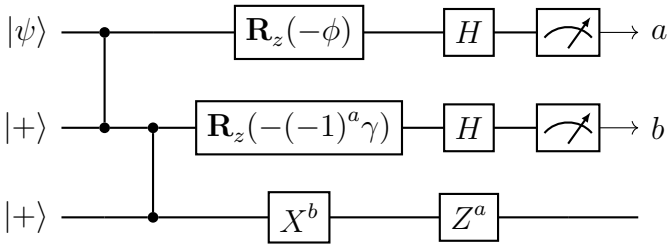
(2.106)

For instance, the following deterministic computation



(2.107)

can be turned into



(2.108)

Formal definition of MBQC. We can now combine the above ideas to properly define MBQC:

Definition 2.3.3. *An MBQC computation is described by a pattern* $(G, I, O, \{\phi_i\}_{i\in O^c}, <, f)$*, where* $G = (V, E)$ *is a graph,* $I \subseteq V$ *is the set of input nodes,* $O \subseteq V$ *is the set of output nodes,* $\{\phi_i\}_{i\in O^c}$ *(where* O^c *denotes the set of nodes in* V *which are not in* O*) is a family of measurement angles,* $<$ *is an order on* V *(intuitively corresponding to the order in which the qubits will be measured), and* $f\colon O^c \to I^c$ *is a function such that:*

- $(i, f(i)) \in E$
- $f(i) > i$
- *for any neighbor* k *of* $f(i)$*,* $k > i$

Moreover, given such a pattern, we can run an MBQC computation as explained in Algorithm 1.

Universal set of gates. The Algorithm 1 shows how to run an MBQC computation given a pattern, but does not explain how to choose the pattern. We describe now how to design this patterns to run any computation. But quantum circuits quickly get too big when considering more than a few qubits, so we will use a different representation to describe a pattern:

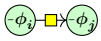
- The graph G will be represented using green nodes ○ for vertices, and edges will have a yellow node[7] in between like —□—.

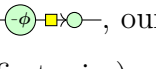
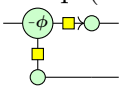
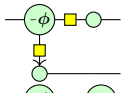
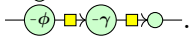
[7] Yes, it looks like ZX-calculus... but no worries, we will see that this representation also make sense when read as a ZX-diagram. Note that our representation is not exactly the original one (a pattern was

Algorithm 1 MBQC computation

Inputs: An MBQC pattern $(G = (V, E), I, O, \{\phi_i\}_{i \in O^c}, <, f)$, and for each $i \in I$ an input qubit labelled i.
Output: For each $i \in O$, an output qubit labelled i.
Algorithm:

1. For each $i \in I^c$, prepare a qubit $|+\rangle$ and label it i.
2. For each edge $(i, j) \in E$, apply $\wedge\mathbf{Z}$ between qubits i and j.
3. For each $i \in V$, initialize $s_i^X := 0$ and $s_i^Z := 0$.
4. For each $i \in O^c$ (taken in the ascending order implied by $<$):
   a) Measure qubit i with angle $\phi_i' := (-1)^{s_i^X}\phi_i + s_i^Z\pi$ to obtain a measurement outcome s_i.
   b) Update $s_{f(i)}^X := s_{f(i)}^X \oplus s_i$ and for all neighbors k of $f(i)$ except i, update $s_k^Z := s_k^Z \oplus s_i$.
5. For each output qubit $i \in O$, apply $\mathbf{X}^{s_i^X}\mathbf{Z}^{s_i^Z}$ on qubit i.

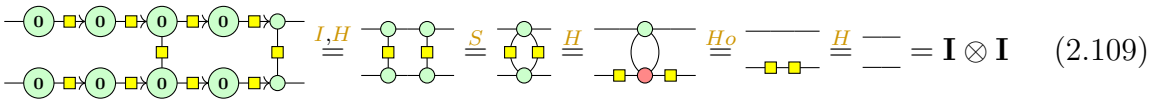
- Any vertex $i \in O^c$ is labelled  .
- Any input vertex $i \in I$ has an additional edge (without any yellow node) arriving from its left: .
- Any output vertex $i \in O$ has an additional edge (without any yellow node) leaving from its right: .
- For any vertex $i \in O^c$, we add on the edge $(i, f(i))$ an arrow tip going towards $f(i)$: 
- The order $<$ will be implicitly described by the position of the vertices in the grid: the left-most qubits are measured first, and if two nodes are in the same column we measure going from the top to the bottom.

For instance, our first example pictured in Eq. (2.100) has the pattern  , our second example Eq. (2.101) has the pattern  (when correcting the first wire) or  (when correcting the second wire), the example Eq. (2.108) has the pattern  . What is nice with this notation is that if we interpret this pattern as a ZX-diagram (forgetting the arrows), then the MBQC computation is exactly applying the operation corresponding to the ZX-diagram. This can be checked for instance for our first example in Eq. (2.100). This can also easily be verified formally by noting that if

typically represented using a labelled graph with standard edges (potentially directed to describe f) and different shapes for input and outputs), but our representation will soon come in handy since we will be able to use all the ZX-calculus machinery.

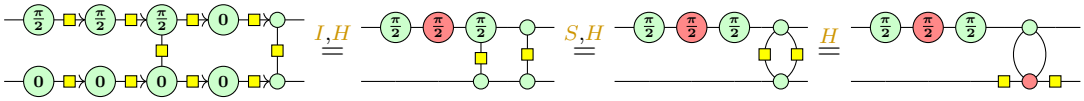
in MBQC the outcome is zero then we are actually projecting on $-(\phi_i)$: using the spider fusion rule we can merge it with the auxiliary $\circ-$ to obtain $(-\phi_i)$. And if the measurements are not zero, then the corrections forces the final state to be exactly the same as the one obtained if the measurements would have all been zero.

We can now use this property to design the pattern of our gates. Note that due to our interest in blind quantum computing, we want the graph to be the same for all gates to make sure it does not reaveal information about the performed gates (this kind of graph is called a *brickwork state* [BFK09] due to its shape). For these reasons, our basic set of gates will be on two wires, and one-wire gates will be tensored with an identity wire. Here is the pattern associated to the **I** gate, together with the proof of correctness:

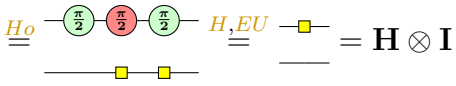


(2.109)

Here is the pattern associated to the **H** gate:

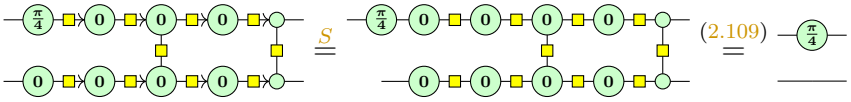


(2.110)



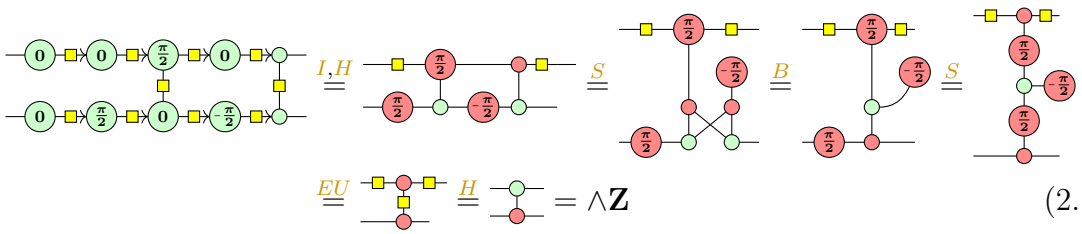
(2.111)

Now, this is the pattern associated to the $\mathbf{R}_z(\pi/4)$ gate:



(2.112)

And finally the pattern associated to the $\wedge\mathbf{Z}$ gate:



(2.113)

Combining the gates to form a circuit. Now that we have our set of elementary gates, it is possible to combine them to build any circuit. If we connect the output of one pattern to the input of another patter, we obtain (after applying one trivial S rule) a new pattern corresponding to the sequential composition of two computations. Because our gates act on two qubits, we also need to the pattern pictured in Figure 2.6 (known as *brickwork state*) to make sure we can apply any unitary on any qubit.

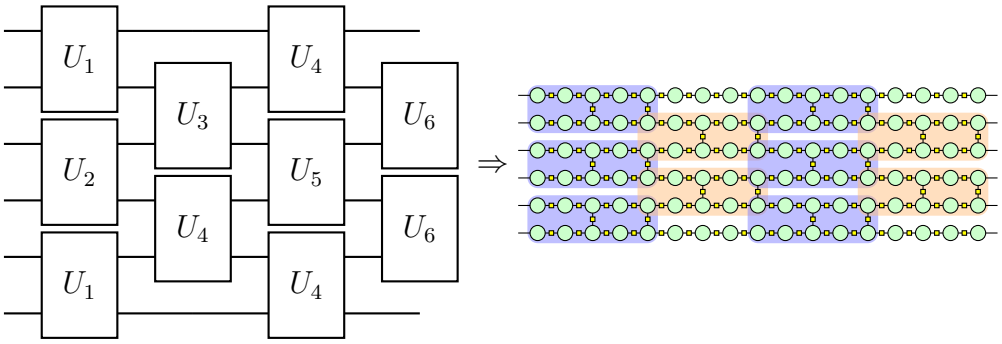
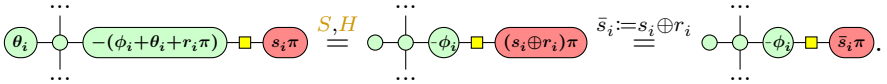


Figure 2.6: Composing multiple gates in an MBQC computation. The nodes which are not colored in the pattern are supposed to have phase 0, or for efficiency reason they can also be replaced with wires. Other nodes have phases corresponding to the gates.

2.3.7 Universal Blind Quantum Computing

Universal Blind Quantum Computing [BFK09] (UBQC) is a protocol that allows a weak quantum client Alice—able to prepare and send only $|+_\theta\rangle$ states—to delegate computations on a remote quantum server Bob without revealing her input, output and algorithm (besides its size). It heavily relies on MBQC, except that:

- The potential input qubits of the client are one-time-padded[8] by applying a random $\mathbf{X}^a\mathbf{Z}^b$ gate before being sent to the server. The corrections will be adapted accordingly.
- The auxiliary qubits are sent by the clients. Moreover, instead of sending $|+\rangle$, the client sends a random $|+_{\theta_i}\rangle$ where $\theta_i \in \{0, \frac{\pi}{4}, \ldots, \frac{7\pi}{4}\}$.
- The measurements are done by the server, but the corrected angles are sent by the client.
- Instead of instructing a measurement with angle ϕ_i, the client will replace it with $\delta_i \coloneqq \phi_i + \theta_i + r_i\pi$, where $r_i \xleftarrow{\$} \{0,1\}$ is randomly sampled. The r_i's are useful to hide the measurement outcome to the server using a one-time pad. For this reason, the client needs to update the measurement s_i given by the server into $\bar{s}_i \coloneqq s_i \oplus r_i$ and use this new value to compute the corrections as in the UBQC protocol.

This works because 
.

Definition 2.3.4 (UBQC)**.** *The Universal Blind Quantum Computing protocol is described in Protocol 1.*

[8] The *one-time-pad* is a statistically secure encryption scheme in which a bit b is encrypted by computing $b \oplus r$ where $r \in \{0,1\}$ is sampled uniformly at random.

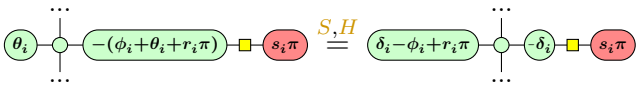
Protocol 1 UBQC protocol

Parties: One weak quantum client (Alice) and one quantum server (Bob).
Alice's inputs: A quantum circuit to evaluate, and a set I of input qubits (in this thesis we will often restrict ourselves to circuits where the inputs are hard-coded into the circuit).
Alice's outputs: The qubits or measurements obtained after the evaluation of the circuit on her inputs (in this thesis we will often restrict ourselves to circuits where the outcomes are classical measurements).
Protocol:

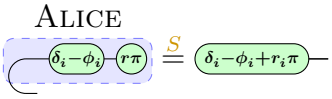
1. Alice converts the circuit into a brickwork MBQC pattern $(G, I, O, \{\phi_i\}_{i\in O^c}, <, f)$ as explained in Section 2.3.6 and sends the description of the graph $G = (V, E)$ to Bob (typically the size of the circuit is enough to derive G).
2. For each $i \in V$, Alice initializes $s_i^X := 0$ and $s_i^Z := 0$.
3. For each input qubit $i \in I$, Alice randomly samples $(a_i, b_i) \xleftarrow{\$} \{0,1\}^2$, applies $\mathbf{X}^{b_i}\mathbf{Z}^{a_i}$ on qubit i and updates $s_i^Z := s_i^Z \oplus a_i$, $s_i^X := s_i^X \oplus b_i$ and for the neighbors k of i, $s_k^Z := s_k^Z \oplus b_i$.
4. For each $i \in I^c$, Alice samples $\theta_i \xleftarrow{\$} \{0, \frac{\pi}{4}, \ldots, \frac{7\pi}{4}\}$, and sends to Bob a qubit $|+_{\theta_i}\rangle$ labeled i.
5. For each edge $(i, j) \in E$, Bob applies $\wedge\mathbf{Z}$ between qubits i and j.
6. For each $i \in O^c$ (taken in the ascending order implied by $<$):
   a) Alice samples $r_i \xleftarrow{\$} \{0, 1\}$ and sends $\delta_i := \phi_i + (-1)^{s_i^X}\theta_i + s_i^Z\pi + r_i\pi$ to Bob.
   b) Bob measures qubit i with angle δ_i, and sends the measurement outcome s_i to Alice.
   c) Alice corrects the measurement by defining $\bar{s}_i := s_i \oplus r_i$.
   d) Alice updates $s_{f(i)}^X := s_{f(i)}^X \oplus \bar{s}_i$ and for all neighbors k of $f(i)$ except i, update $s_k^Z := s_k^Z \oplus \bar{s}_i$.
7. If Alice wants classical outcomes, Bob measures each qubit $i \in O$ in the computational basis, sends the outcome s_i to Alice, and Alice updates it into $\bar{s}_i := s_i \oplus s_i^X$ before outputting them.
8. Otherwise, Bob sends each output qubit $i \in O$ to Alice, who applies on it $\mathbf{X}^{s_i^X}\mathbf{Z}^{s_i^Z}$ and outputs them.

To see that this protocol does not leak to Bob any information about the computation (i.e. about any θ_i), we can rewrite this protocol into an equivalent protocol in which it is clear that no information about θ_i can leak to Bob. First, we can remark that instead of computing $\delta_i := \phi_i + \theta_i + r_i\pi$, we can rather sample the measurement angle δ_i uniformly at random in $\{0, \frac{\pi}{4}, \ldots, \frac{7\pi}{4}\}$, then sample $r_i \xleftarrow{\$} \{0, 1\}$ and finally compute

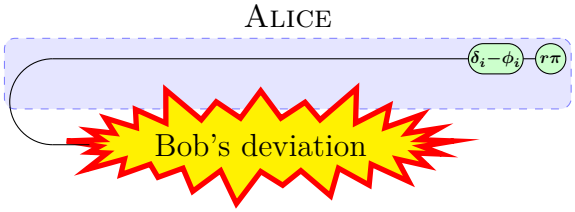
$\theta_i \coloneqq \delta_i - \phi_i + r_i\pi$. Of course, the correctness is not impacted:



Moreover, instead of preparing $|+_{\tilde{\phi}_i-\phi_i+r_i\pi}\rangle$, Alice can instead send half of a Bell pair to Bob, and perform the appropriate rotation before measuring in the $\{|+\rangle, |-\rangle\}$ basis (this virtual protocol is used of course only for the security proof, and would be too inefficient to be used in practice):



The interest of this construction is that it is now possible to push any action depending on the secret computation ϕ_i *after* any deviation performed by Bob:



In particular, due to the no-signaling principle (see Section 2.3.2), the view of Bob is independent of the operations performed by Alice. Therefore, Bob cannot learn any information about the computation performed by Alice. This same argument can be used to show that UBQC is secure even in the strong Constructive Cryptography framework [DFP+14]

CHAPTER 3

Introduction to Cryptography

"If you want to keep a secret, you must also hide it from yourself."

— George Orwell, *1984*

Intuitively, a "secure" protocol is a protocol in which no attack is possible. However, when formalizing this notion, we end up with different kinds of adversaries and models of security that we introduce in this chapter. Note that the notions which are necessary only for a single chapter (like LWE, Constructive Cryptography, Zero-Knowledge) will be introduced in the relevant chapters.

3.1 Notations

We use the notation $\mathsf{poly}(\lambda)$ to denote any non-negative function f smaller than some polynomials: $\exists d \in \mathbb{N}, \Lambda \in \mathbb{N}, \forall \lambda > \Lambda, f(\lambda) \leq \lambda^d$. $\mathsf{negl}(\lambda)$ will denote any *negligible* function, i.e. any function f which decays faster than any inverse polynomial: $\forall d \in \mathbb{N}, \exists \Lambda \in \mathbb{N}, \forall \lambda > \Lambda, 0 \leq f(x) \leq \frac{1}{\lambda^N}$. An *overwhelming* function (or probability) is a function f negligibly close to 1: $f = 1 - \mathsf{negl}(\lambda)$.

We denote by $s \xleftarrow{\$} X$ the action of sampling s uniformly at random over a finite set X. For two probability distributions P and Q on a countable set X, we define the (total variation) statistical distance as $\Delta(P, Q) \coloneqq \sup_{A \subseteq X} |P(A) - Q(A)| = \frac{1}{2} \sum_{x \in X} |P(x) - Q(x)|$. When $X = \{0, 1\}$, this quantity $\Delta(P, Q) = |P(1) - Q(1)|$ is called the *advantage* and is linked with the best probability $\frac{1}{2}(1 + \Delta(P, Q))$ of being able to distinguish a sample from P from a sample from Q.

3.2 Parties, Protocols and Non-Uniformity

Protocol. A protocol $\boldsymbol{\pi} = (\mathsf{P}^{(1)}, \ldots, \mathsf{P}^{(n)})$ assigns to each party $\mathsf{P}^{(i)}$ a list of instructions that they should follow in order to perform a specific task (we will use the same notation for the identifier of the party $\mathsf{P}^{(i)}$ and for its instructions). Each party $\mathsf{P}^{(i)}$ can have an input, an output and can receive messages and send messages to other parties. Moreover, each protocol is typically parameterized by a *security parameter* $\lambda \in \mathbb{N}$: the bigger λ is, the harder it is to attack the protocol. We will denote by

$$\mathsf{OUT}_{\mathsf{Alice}}(\mathsf{Alice}_\lambda(x) \leftrightsquigarrow \mathsf{Bob}_\lambda(y)) \tag{3.1}$$

the random variable representing the output of the party Alice, where Alice's input is $(1^\lambda, x)$, Bob's input is $(1^\lambda, y)$, λ being the security parameter of the protocol. Alone, $\mathsf{Alice}_\lambda(x) \leftrightsquigarrow \mathsf{Bob}_\lambda(y)$ will be the random variable containing in a tuple the outputs of Alice and Bob.

Interactive Party. Because cryptographic protocols are usually interactive (i.e. messages are sent and received at various points throughout a protocol), it is practical to model any party (or adversary) P as a randomized process which is called each time a message m is received from party s:

$$(((m'_1, d_1), \ldots, (m'_n, d_n)), \rho_{i+1}) \leftarrow \mathsf{P}(m, s, \rho_i) \tag{3.2}$$

This process outputs a list of messages m'_i to send, together with the corresponding recipient d_i. Moreover, ρ_i and ρ_{i+1} are the internal states of P before and after the communication, playing the role of a memory between runs and whose initial value ρ_0 encodes[1] the actual input of the party. To also allow parties to send the first message without having received any message, s will be equal to $\perp$ for the first call. If one d_j is equal to $\perp$, the output of the party P is defined as ρ_{i+1} and the party will be "stopped" (it won't be called anymore).

In this thesis, we will often have only two parties and a fixed number of rounds of communication: in that case we will often decompose P into $(\mathsf{P}_1, \ldots, \mathsf{P}_n)$. We write $(m'_i, \rho_i) \leftarrow P_i(m_i, \rho_{i-1})$ to denote the fact that after receiving from the other party a message m_i, the ith message sent by P is m'_i (ρ_i and ρ_{i-1} being defined as before). ρ_0

[1] The exact encoding is not very important, for instance for quantum parties we could choose to put the first qubit of the internal state in state $|0\rangle$ to denote the fact that this is the input, and add afterwards a quantum state encoding x. More generally, we always assume that we can encode all objects into classical bit strings or quantum states (any bit string x can always be encoded into $|x\rangle$).

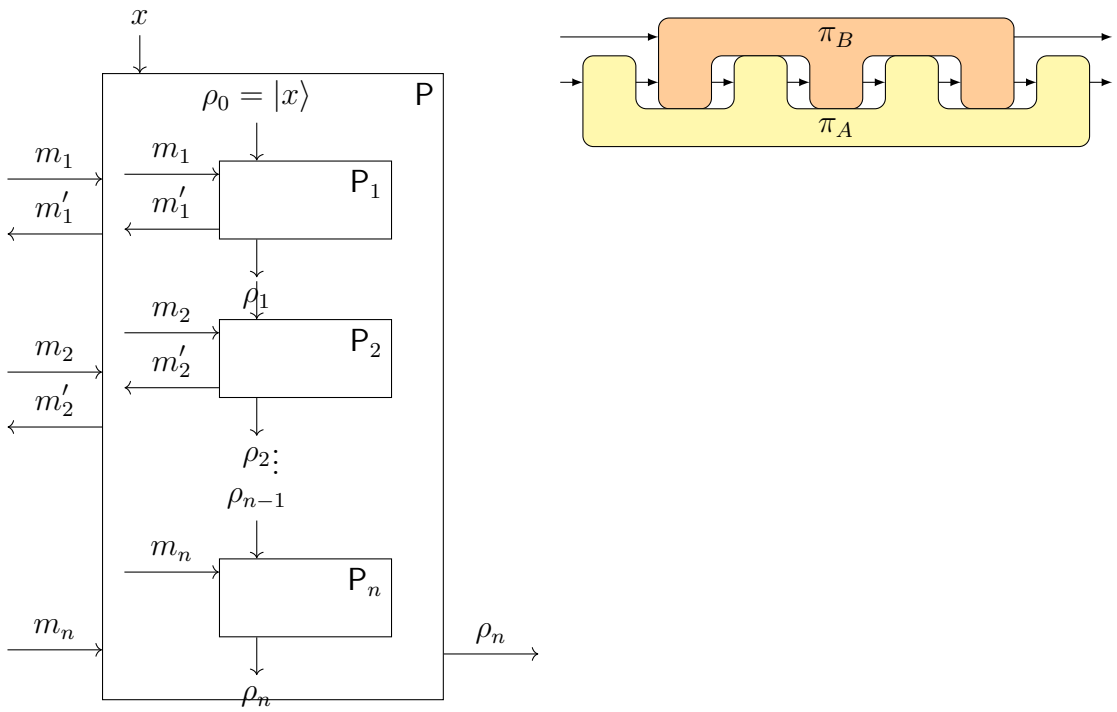


Figure 3.1: Representation of an interactive quantum party P into a sequence $(\mathsf{P}_1, \ldots, \mathsf{P}_n)$. This representation is sometimes called "quantum comb" due to its shape.

will encode the input x of P, and ρ_n will be defined as the output of P. For brevity, we will often omit the ρ's from this notation and we will also remove m_1 from P_1 if P sends a message before receiving any message, and m'_n will be removed from P_n if P does not send any message before outputting it's result. This decomposition is sometimes referred to as *quantum comb* (due to its shape) and was studied for instance in [CDP09, GW07]. This is illustrated in Figure 3.1.

Then, the processes P or $(\mathsf{P}_1, \ldots, \mathsf{P}_n)$ can be implemented in different manners:

- Using Turing machines (with a random tape) or classical circuits: P is then said to be *classical* (all messages and the internal states are then classical). Moreover, if P runs in polynomial time (in the length of its input, and therefore also in λ) and if P is always stopped after a polynomial number of times, P is said to be PPT (Probabilistic Polynomial Time).
- Using CPTP maps or quantum circuits: P is then said to be *quantum*. Moreover, if the size of the circuit of P is polynomial (in the size of the input, i.e. in the number of input qubits and therefore in λ), and if P is always stopped after a polynomial number of times, P is said to be (interactive) QPT (Quantum Polynomial Time). If P has no restriction on the size of its circuit, P is said to be *computationally unbounded* (or simply *unbounded*).

We may also give to a process P blackbox access to some oracles $\mathcal{O}\colon X \to Y$ by denoting it $\mathsf{P}^{\mathcal{O}}$. P will then be able to query $\mathcal{O}$ but will not be able to see how $\mathcal{O}$ is implemented internally (think of $\mathcal{O}$ as a new gate that can be inserted into the circuit of P).

It is possible to define more formally the notation $\mathsf{P}_\lambda^{(1)}(x) \leftrightsquigarrow \mathsf{P}_\lambda^{(2)}(y) \ldots \leftrightsquigarrow \mathsf{P}^{(n)}$ introduced in Eq. (3.1) which defines the random variable of the output of the protocol involving the parties $(\mathsf{P}^{(1)}, \ldots, \mathsf{P}^{(n)})$. For brevity and readability, we won't enter into too much details, but this can easily be defined using an *environment* that calls each party one after the other with the appropriate parameters depending on the messages sent by other parties. There is, however, one small technical detail: the choice of the order of the calls is important to ensure the above random variable is well defined (different orders could lead to completely different distributions). When only correctness matters, this can be solved by keeping a queue of the sent messages, and calling each party following this queue. Note that well designed protocols should have the same output distribution regardless of the evaluation order. For security, it is possible for instance to use [PMM+17] to properly define this in complex time-dependent protocols with many parties. In our case, most of the time the flow of messages is simple enough that this won't be necessary and we will just use quantum combs [CDP09].

Uniform and Non-Uniform Adversaries. An *adversary* is a party involved in a protocol that may arbitrarily deviate from this protocol, typically in order to obtain sensitive information or disturb the execution of the protocol. As for parties, adversaries can be unbounded, PPT or QPT. If the protocol is proven secure against unbounded adversaries, the protocol is said to be *statistically* or *information-theoretically* secure, otherwise it is said to be *computationally* secure.. However, it is sometimes practical to assume that adversaries have a little bit more power than PPT or QPT machines. Namely, a *non-uniform* adversary $\mathcal{A}$—which must be opposed to a standard uniform adversary—accepts an additional input ρ_λ of size $\mathsf{poly}(\lambda)$. The $\{\rho_\lambda\}_{\lambda\in\mathbb{N}}$'s (which are specified together with the considered adversary) can be seen as advices helping the adversary. Note that this advice, which depends only on the size of the input, is not a realistic assumptions: a polynomial-time machine having access to an additional advice can solve problems in $\mathsf{P}/\mathsf{poly}(\lambda)$ which is believed to be bigger than P. However, this is used in practice to say that if a protocol is secure against non-uniform adversaries, then the protocol is also secure against uniform adversaries.

3.3 The Different Models of Security

Intuitively, a "secure" protocol is a protocol in which no attack is possible. However, when formalizing this notion, we end up with different models of security having different properties:

- *Game-based security*: The proofs are usually easier to derive in this model, but the security guarantees are limited since we usually do not get any guarantee—without further work—if the protocol is composed with other protocols.
- *Standalone security*: In this stronger model, we also get guarantees when the protocols are composed sequentially (i.e. when protocols are run one after the other) but not when they are composed in parallel (i.e. run at the same time). This will also prove useful when defining later Zero-Knowledge proofs.
- *General composable security*: General composability provides security guarantees when the protocol is composed both sequentially or in parallel into other protocols. In this thesis we will focus on the Constructive Cryptography (CC) framework. These guarantees are very strong, but it is usually hard to obtain general composability and some protocols are even impossible [CF01] to be proven secure in this framework (we also derive impossibility results in this thesis).

In this thesis, most of our security proofs are stated in term of game-based security as we also proved impossibility results regarding the CC framework. We will also use standalone security when considering our results on Zero-Knowledge proofs on Quantum State. We introduce now game-based security, and we will present the other models of security in the relevant chapters (Chapters 6 and 7).

More details can be found in the tutorial of Lindell on simulation-based security [Lin17], in the tutorial of Shoup on game-based security [Sho04] or in the generic book of Goldreich on the foundations of cryptography [Gol01].

Game-Based Security In the *game-based security framework*, the definition of security is pretty straightforward: we define a game between a *challenger* (playing the role of the honest party) and a malicious adversary: a protocol is said to be secure if no adversary can win this game with "good" probability: depending on the game, this probability may be negligible or smaller than $1/2 + \mathsf{negl}(\lambda)$. Depending on the targeted security,

Note that we can describe games in different ways. Either directly inlined in an equation:

$$\Pr\left[\tilde{c} = c \mid (\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda), c \xleftarrow{\$} \{0,1\}, (k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0^{(c)}), \tilde{c} \leftarrow \mathcal{A}_2(k)\right] \quad (3.3)$$

by describing the whole interaction with the adversary:

$\texttt{Game1}(\lambda)$

1 :	**Challenger**		**Adversary**
2 :		$\xleftarrow{(m^{(0)}, m^{(1)})}$	Compute $m^{(0)}$ and $m^{(1)}$
3 :	$c \xleftarrow{\$} \{0,1\}$		
4 :	$(k, t_k) \leftarrow \mathsf{Gen}(1^\lambda, m^{(c)})$		
5 :		$\xrightarrow{k}$	
6 :		$\xleftarrow{\tilde{c}}$	Guess $\tilde{c}$
7 :	**return** $\tilde{c} = c$		

or by writting only the code of the challenger, in charge of calling the adversary seen as an oracle:

$\texttt{Game1}^{\mathcal{A}}(\lambda)$

1 : $(\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
4 : $\tilde{c} \leftarrow \mathcal{A}_2(k)$
5 : **return** $\tilde{c} = c$

Depending on the context, we may use all these notations.

An example of a famous game is the IND-CPA game (for "Indistinguishability under chosen-plaintext attack", see for example [Gol04]). This game quantifies how secure is a public-key encryption scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Invert})$, where $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda)$ is a PPT algorithm that generates a public key k and a secret key t_k (the t stands for "trapdoor"), $\mathtt{Enc}$ is a function used to encrypt a message m into a ciphertext e using the public key k: $e \leftarrow \mathtt{Enc}(k, m)$, and $\mathtt{Invert}$ is a function used to decrypt a ciphertext e into the original message m using the trapdoor t_k: $m \leftarrow \mathtt{Invert}(t_k, e)$:

Definition 3.3.1 (IND-CPA). *A public-key encryption scheme $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Invert})$ is said to have (quantum) indistinguishable encryption under chosen plaintext attacks if for any QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ (again, the internal state of $\mathcal{A}_1$ will be implicitly given to $\mathcal{A}_2$) and for any set of advices $\{\rho_\lambda\}_{\lambda \in \mathbb{N}}$,*

$$\Pr\left[\text{IND-CPA}_{\mathtt{Gen}}^{\mathcal{A}, \{\rho_\lambda\}_\lambda}(\lambda)\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{3.4}$$

where $\Pr\left[\text{IND-CPA}_{\mathtt{Gen}}^{\mathcal{A}, \{\rho_\lambda\}_\lambda}(\lambda)\right]$ is a shortcut for $\Pr\left[\text{IND-CPA}_{\mathtt{Gen}}^{\mathcal{A}, \{\rho_\lambda\}_\lambda}(\lambda) = \mathbf{true}\right]$, and IND-CPA is defined as follows (note that here we only define the challenger in charge of calling the adversary).

$\text{IND-CPA}^{\mathcal{A},\{\rho_\lambda\}_\lambda}_{\texttt{Gen}}(\lambda)$

1 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda)$
2 : $(m^{(0)}, m^{(1)}) \leftarrow \mathcal{A}_1(k, \rho_\lambda)$
3 : $c \xleftarrow{\$} \{0,1\}$
4 : $\tilde{c} \leftarrow \mathcal{A}_2(\texttt{Enc}(k, m^{(c)}))$
5 : **return** $\tilde{c} = c$

The proofs of security in this model typically follow the same approach: we define a series of games ($\texttt{Game1}, \ldots, \texttt{Gamen}$) (sometimes called *hybrid games*) where `Game1` is our target game, and `Gamen` is a game impossible to win with good probability (for instance because there is not even a single reference to the secret in the game). Then, we prove for all i that the probability of winning `Gamei` is close to the probability of winning `Gamei+1` (otherwise we can use the adversary to break some hard problems: this is known as a *reduction*): this gives that the probability of winning `Game1` is close to the probability of winning `Gamen`, and therefore that it is also impossible to win `Game1` with good probability.

While game-based security proofs are easier to write than in most other frameworks, there are multiple problems to the game-based approach:

- If we compose a protocol secure in the game-based model with other protocols, the resulting protocol may not be secure when used in an arbitrary environment.
- In game-based security, it can be hard to properly define the security of unusual or complicated functionalities (like in Secure Multiparty Computing), in such a way that covers all possible attacks. I like to explain it this way: in game-based security, we characterize the security of a protocol in term of what is *not* possible to do (for instance "it is impossible to recover the message m when m is sampled uniformly at random", or "it is impossible to learn the basis of the obtained $|+_\theta\rangle$ state"). But typically, it is hard to characterize all the properties that are undesirable: for instance, a protocol leaking half of the message would also certainly not be considered as secure, or one may also want to make sure that an adversary cannot generate a state $|+_{3\theta}\rangle$. We will see later that simulation-based security proceeds differently, and characterizes instead the security in term of what *is* possible to do ("it is possible to simulate the view of any attacker given only access to the size of the message", or "it is possible to simulate the view of the attacker given only access to $|+_\theta\rangle$"). This often naturally leads to properties that are *not* possible (if

you can simulate the view knowing only the size of m, it is impossible to recover m, and if you only have access to $|+_\theta\rangle$, you cannot generate $|+_{3\theta}\rangle$ by the laws of quantum mechanics), and we claim that it is therefore more natural to design simulation-based security properties than game-based security properties.

For these reasons, other models of security have been defined and will be described later in Chapters 6 and 7.

Chapter 4

QFactory: Classically Faking a Quantum Channel

"The best way of successfully acting a part is to be it."

— Arthur Conan Doyle, *The Adventure of the Dying Detective*

The QFactory is a protocol we developed to classically fake a quantum channel. This modular functionality, known as *Remote State Preparation* (RSP) allows a purely classical client to prepare on a remote quantum server a quantum state, in such a way that the classical description of that state is only known to the client. Because of its modularity, it is possible to include it in existing quantum protocols to replace the quantum channel: we show notably in Section 4.5 how it can be composed with the UBQC protocol to obtain blind quantum computing with a purely classical client.

This chapter first gives in Section 4.1 a quick overview of our method, it presents then in Section 4.2 the cryptographic assumptions required in the protocol. We present in Section 4.3 the QFactory protocol that can produce what we call hidden GHZ states, and describe in Section 4.4 how it is possible to obtain other classes of states. In Section 4.5 we show the main application of QFactory which is to obtain classical-client blind quantum computing. Finally, in Section 4.6 we explain how we can adapt our construction to rely on a more standard security assumption, in Section 4.7 we present open questions, and in Section 4.8 we discuss how our protocol compares with the related works. The QFactory protocols requires a particular cryptographic family: in this chapter we will only

base our protocol on its abstract properties, and we will see how it can be constructed in Chapter 5.

Historical Notes. This thesis will not follow the "historical design" of QFactory: historically we first obtained QFactory protocols that were generating one-qubit states, and we generalized it to multi-qubit states later. However, in this thesis we will present it in the other way: the one-qubit QFactory will be seen as a special case of the multi-qubit QFactory. Note also that this thesis slightly improves some of our previous results (by simplifying some gadget circuits or generalizing security proofs).

More precisely, we designed a first version of the protocol between March and July 2017 [Col17] that was producing $|+_\theta\rangle$ states, but unfortunately we had no proof of security. We only managed to prove, in the next months, a weak statement against "honest-but-curious" adversaries [CCK+18] that I presented at QCrypt2018 (note that we published this original paper much later [CCK+21]). We then improved both the function construction and the protocol, and we derived a full security proof against an arbitrarily malicious adversary [CCK+19], again for single qubit states. The generalization to multi-qubits states (hidden GHZ) arrived later [CGK21] (note that this last paper also study Non-Interactive Zero-Knowledge proofs on Quantum States, presented in Chapter 7).

Two independent papers [Mah18a, GV19] achieved related results: we compare these approaches to our own result in Section 4.5 (the ground-breaking work of [Mah18a] inspired countless other works as explained in the introduction in Chapter 1).

4.1 Intuition and Overview of QFactory

We provide now a short informal explanation of our general QFactory protocol going through the intuition behind the protocol.

Goal. The goal is to allow a classical client Alice to prepare on a remote quantum server Bob a multi-qubits state, in such a way that this state should be unknown to Bob but fully describable by Alice. More precisely, the set of states that we will consider are *hidden GHZ states*, which are states whose form is a permutation of $\mathbf{X}^{\mathbf{a}}((|0\dots0\rangle \pm |1\dots1\rangle)\,|0\dots0\rangle)$ (dropping the normalization factor) for some $\mathbf{a} \in \{0,1\}^n$. These states are named that way since they are an extension of the Greenberger–Horne–Zeilinger (GHZ) states [GHZ89] whose form is $|0\dots0\rangle + |1\dots1\rangle$.

The *support* of a hidden GHZ state is the set of qubits that are part of the original GHZ state (this basically[1] corresponds to the set of entangled qubits). A qubit in the support of a hidden GHZ state is said to be *supported*. This set is described by a bit string $\mathbf{d}_0 \in \{0,1\}^n$: the ith qubit is supported iff $\mathbf{d}_0[i] = 1$. For instance, $|1010\rangle - |1100\rangle = |1\rangle X^{01}(|00\rangle - |11\rangle)|0\rangle$ is a hidden GHZ state whose support is $\mathbf{d}_0 = 0110$. Note that for any $(x, x') \in (\{0,1\}^n)^2$, the state $|x\rangle + |x'\rangle$ is always a hidden GHZ state whose support is $\left\{ i \;\middle|\; x_i \neq x'_i \right\}$ (this can be seen by factoring out qubits where $x_i = x'_i$). As a consequence, we can describe the support as:

$$\mathbf{d}_0 = x \oplus x' \tag{4.1}$$

In the QFactory protocol, Alice will be able to choose the support $\mathbf{d}_0$, and Bob will obtain a hidden GHZ state whose support is $\mathbf{d}_0$ without having any information on $\mathbf{d}_0$.

Cryptographic assumptions. In order to give some advantages to Alice over Bob, we need to use a classical cryptographic family of functions $\{f_k \colon \mathcal{X} \to \mathcal{Y}\}_{k \in \mathcal{K}}$, together with a function $h \colon \mathcal{X} \to \{0,1\}^n$ having several properties. The exact list of requirements is given in Definition 4.2.1, but here are the important ones. For any $\mathbf{d}_0 \in \{0,1\}^n$ (corresponding to the status of the hidden GHZ state), we can generate using a function $\mathtt{Gen}(1^\lambda, \mathbf{d}_0)$ an index k and a trapdoor t_k such that:

- f_k is[2] 2-to-1 (i.e. for all x, $|f_k^{-1}(f_k(x))| = 2$).
- f_k can be efficiently computed given k, but should be hard to invert without t_k. Moreover, it should be hard to obtain any information on $\mathbf{d}_0$ given k.
- Given the trapdoor t_k, f_k can be efficiently inverted.
- For any $x \neq x'$ such that $f(x) = f(x')$, $h(x) \oplus h(x') = \mathbf{d}_0$.

We will say that such a family is $\mathsf{GHZ}^{\mathsf{H}}$ capable (more details in Definition 4.2.1), and we will describe how to build such a family in the Chapter 5.

GHZ-QFactory. Instead of receiving directly a quantum state, Bob will receive classical instructions producing a quantum state in such a way that the instructions should not leak any information on the final produced quantum state. More precisely, since in our case we are interested in the preparation of hidden GHZ states, we proceed as follows: Alice samples $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$ and sends k to Bob (which can be seen as the

[1] Of course, if the initial GHZ state has size 1 it does not make sense to talk about the entangled qubits.

[2] Unfortunately we will see later that such family seems are impossible to obtain (as far as we know) with post-quantum secure assumptions so we will generalize this to approximate δ-2-to-1 functions.

instructions "encrypting" a hidden GHZ state of support $\mathbf{d}_0$). Then, in order to produce the quantum state, Bob will run the unitary corresponding to $x \mapsto (h(x), f_k(x))$ on the superposition of all inputs (if f's input set is $\{0,1\}^N$, this can be done by applying one Hadamard gate per qubit, we will discuss later how to extend to more complex sets). The state obtained by Bob will be

$$\sum_x |x\rangle |h(x)\rangle |f_k(x)\rangle = \sum_y (|x_y\rangle |h(x_y)\rangle + |x'_y\rangle |h(x'_y)\rangle) |y\rangle \tag{4.2}$$

where in the right hand side, we sum over the elements y in the image of f_k, and (x_y, x'_y) are the two preimages of y (reminder: the function is 2-to-1). In order to collapse this huge superposition, Bob will measure the last register in the computational basis, obtaining an outcome y. The remaining state will be the following (where $x = x_y$ and $x' = x'_y$ are the two preimages of y):

$$|\psi\rangle \coloneqq |x\rangle |h(x)\rangle + |x'\rangle |h(x')\rangle \tag{4.3}$$

At that step, we can notice something interesting: given y and t_k it is possible to compute x and x' and therefore it is possible to describe $|\psi\rangle$ completely. However Bob does not know t_k and therefore cannot compute x and x': this state is (informally) unknown to Bob. But so far it is hard to quantify exactly which part of $|\psi\rangle$ is known to Bob, and we do not have yet a GHZ state whose support is $\mathbf{d}_0$. So Bob will now measure the first register in the Hadamard basis to "remove" the first qubits, obtaining a state

$$|h(x)\rangle + (-1)^\alpha |h(x')\rangle \tag{4.4}$$

for some $\alpha \in \{0,1\}$ which depends on the outcome of the measurement $\{b_i\}_i$. This state is now a hidden GHZ state whose support is equal to $\mathbf{d}_0$: this can be seen using Eq. (4.1) and the fact that we assumed that $h(x) \oplus h(x') = \mathbf{d}_0$ for any two preimages x and x'.

It is now time to let Alice know which state was produced: Bob will then send to Alice y and the measurements $\{b_i\}_i$. Using t_k, Alice can invert y to obtain x, x' and α, which is enough to fully characterized the final hidden GHZ state obtained by Bob. Moreover, by assumption k does not leak any information about $\mathbf{d}_0$, so Bob cannot learn the support of the hidden GHZ state.

In the following sections of this chapter, we will formalize these assumptions, prove the security of this protocol, see how it can be extended to produce other classes of states and used (securely) inside the UBQC protocol. Note also that part of the difficulty is to construct the family f_k: this will be studied in Chapter 5.

4.2 Function Assumptions

All the protocols are based on the existence of a (post-quantum secure) cryptographic family of functions, which is said to be δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable.

Definition 4.2.1 (δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable functions)**.** *Let $\lambda \in \mathbb{N}$ be a security parameter, and $n \in \mathbb{N}$. We say that a family of functions $\{f_k : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda\}_{k \in \mathcal{K}_\lambda}$ with $\mathcal{X}_\lambda \subseteq \{0,1\}^l$ is δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable if there exists a function $h\colon \mathcal{X}_\lambda \to \{0,1\}^n$ (h could be extended to depend on k) such that the following properties are respected:*

- ***efficient generation**: for all $\mathbf{d}_0 \in \{0,1\}^n$ a PPT machine can efficiently sample $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0)$ to generate (with overwhelming probability) an index $k \in \mathcal{K}_\lambda$ and a trapdoor $t_k \in \mathcal{T}_\lambda$.*
- ***efficient computation**: for any index k, the function f_k is efficiently computable by a PPT algorithm $\texttt{Eval}(k, x)$.*
- ***trapdoor**: for any trapdoor t_k and any y, there exists a procedure `Invert` that efficiently inverts f_k when y has two preimages. More precisely, if y has exactly two distinct preimages, we have $\texttt{Invert}(t_k, y) = f^{-1}(y)$. If the number of preimages is not 2, we expect $\texttt{Invert}(t_k, y) = \perp$.*
- ***quantum input superposition**: there exists a QPT algorithm that, on input 1^λ generates a uniform superposition $\sum_{x \in \mathcal{X}_\lambda} |x\rangle$ (see Remark 4.2.2 for more details on this assumption). Moreover, we assume that there exists $l \in \mathbb{N}$ such that $\mathcal{X}_\lambda \subseteq \{0,1\}^l$.*
- *δ-2-**to**-1[3]: for all $k \in \mathcal{K}$, when sampling an input x uniformly at random in $\mathcal{X}_\lambda$, the probability that $y := f_k(x)$ has exactly two distinct preimages (denoted by x_y and x'_y or simply x and x') is at least $1 - \delta$. When $\delta = 0$, we just say that the function is 2-to-1.*
- XOR ***of** h: for all k, there exists $\mathbf{d}_0 \in \{0,1\}^n$ such that for all y, if y has exactly 2 distinct preimages x and x' (i.e. $f_k^{-1}(y) = \{x, x'\}$ with $x \neq x'$), then:*

$$h(x) \oplus h(x') = \mathbf{d}_0$$

Moreover, if k was obtained from $\texttt{Gen}(1^\lambda, \mathbf{d}_0^*)$, then $\mathbf{d}_0 = \mathbf{d}_0^*$. We will always assume that d_0 is easy to obtain from t_k (it is always possible to append d_0 to t_k). Since, fixing k fixes also $\mathbf{d}_0$, in the following we may use interchangeably $\mathbf{d}_0(k)$, $\mathbf{d}_0(t_k)$ or simply $\mathbf{d}_0$.

- ***indistinguishability**: If the index k obtained from $\texttt{Gen}(1^\lambda, \cdot)$ is seen as en encryption function, then—similarly to* IND-CPA *security—a quantum adversary*

[3] In some of our previous work, we called these functions δ-2-regular.

cannot learn $\mathbf{d}_0$ *from* k*. More formally, if we formulate it as an indistinguishability game* $\mathit{IND\text{-}DO}^{\mathcal{A}}_{\mathit{Gen}}$*, where* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *is a non-uniform* QPT *adversary (*$\mathcal{A}_1$ *gives implicitly its internal state to* $\mathcal{A}_2$*), then any non-uniform* QPT *adversary* $\mathcal{A}$ *has only a negligible advantage of winning the game* $\mathit{IND\text{-}DO}$*:*

$$\Pr\left[\mathit{IND\text{-}DO}^{\mathcal{A}}_{\mathit{Gen}}(\lambda)\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{4.5}$$

$\mathtt{IND\text{-}DO}^{\mathcal{A}}_{\mathtt{Gen}}(\lambda)$

1 : $(\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
4 : $\tilde{c} \leftarrow \mathcal{A}_2(k)$
5 : **return** $\tilde{c} = c$

Remark 4.2.2 (Preparation of the inputs). As explained above, it should be possible to create a uniform superposition on $\mathcal{X}_\lambda$, the input of f_k. If $\mathcal{X}_\lambda = \{0,1\}^N$, it can easily be done by applying $\mathbf{H}^{\otimes N}$ on $|0\rangle^{\otimes N}$. Otherwise, if f is only defined on a fraction (at least constant) of $\{0,1\}^N$, and if there exists an efficiently computable indicator function $\mathbf{1}_{\mathcal{X}} \colon \{0,1\}^N \to \{0,1\}$ such that $\mathbf{1}_{\mathcal{X}}(x) = 1$ iff $x \in \mathcal{X}$, it is also possible to efficiently create $\sum_x |x\rangle$ using *rejection sampling*: First by applying $\mathbf{H}^{\otimes N}$ on $|0\rangle$ we obtain $\sum_{x\in\{0,1\}^N} |x\rangle$. Then, we apply $\mathbf{1}_{\mathcal{X}}$ in superposition (adding one auxiliary qubit), which gives:

$$\sum_{x\in\{0,1\}^N} |x\rangle\,|\mathbf{1}_{\mathcal{X}}(x)\rangle = \Big(\sum_{x\in\mathcal{X}} |x\rangle\,|1\rangle\Big) + \Big(\sum_{x\notin\mathcal{X}} |x\rangle\,|0\rangle\Big) \tag{4.6}$$

By measuring the second register, we get an outcome b: if $b = 1$ we have obtained the state $\sum_{x\in\mathcal{X}} |x\rangle$. Otherwise, we restart from the beginning the procedure. Note that the probability of getting $b = 0$ is $\frac{|\mathcal{X}|}{2^N}$ which is at most constant. Therefore by repeating it on average around $\frac{2^N}{|\mathcal{X}|}$ times, we obtain the expected state.

We provide in Chapter 5 an explicit implementations of a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable function where δ can be made negligible if we rely on the Learning-With-Error problem with superpolynomial noise ratio. We also show in Section 4.6 and Theorem 5.3.9 how the protocol and construction can be adapted to polynomial noise ratio.

4.3 Protocol for GHZ State Preparation

We describe in Protocol 2 the protocol GHZ-QFactory introduced informally in Section 4.1.

Protocol 2 GHZ-QFactory

Assumptions: There exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1).
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's inputs: The support $\mathbf{d}_0$ of the hidden GHZ state.
Alice's outputs: The description $(\mathbf{d}, \mathbf{d}', \alpha)$ of the hidden GHZ state $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle$ obtained by Bob whose support is $\mathbf{d}_0$. If Bob is malicious, Alice can also abort.
Bob's output: A hidden GHZ state $|d\rangle + (-1)^\alpha |d'\rangle$ whose support is $\mathbf{d}_0$.
Protocol:

1. Alice generates $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$ and sends k to Bob.
2. Bob performs the following operations, also pictured in Figure 4.1:
   - create the state $\sum_{x \in \mathcal{X}_\lambda} |x\rangle |h(x)\rangle |f_k(x)\rangle$ by applying in superposition the unitary mapping $|x\rangle |0\rangle |0\rangle \mapsto |x\rangle |h(x)\rangle |f_k(x)\rangle$ on the uniform superposition of all inputs (described in Remark 4.2.2),
   - measure the third register in the computational basis, obtaining outcome y,
   - measure the first register in the Hadamard basis, obtaining outcome b.

   Then, Bob sends (y, b) to Alice and outputs the remaining quantum state.
3. Alice computes $(x, x') \leftarrow \mathtt{Invert}(t_k, y)$ (or aborts if $\mathtt{Invert}(t_k, y) = \perp$), $\mathbf{d} \coloneqq h(x)$, $\mathbf{d}' \coloneqq h(x')$ and $\alpha \coloneqq \bigoplus_i b_i(x_i \oplus x'_i) = \langle b, x \oplus x' \rangle$. If $\mathbf{d} = \mathbf{d}'$ and $\alpha = -1$, Alice aborts (this state is not physical, so Bob is malicious). Otherwise, Alice outputs $(\mathbf{d}, \mathbf{d}', \alpha)$.

Remark 4.3.1 (Note on the order of $\mathbf{d}$ and $\mathbf{d}'$). In the protocol GHZ-QFactory, the order of $\mathbf{x}$ and $\mathbf{x}'$ (and therefore of $\mathbf{d}$ and $\mathbf{d}'$) seems arbitrary. However, the order does not matter since $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle = (-1)^\alpha(|\mathbf{d}'\rangle + (-1)^\alpha |\mathbf{d}\rangle)$ (remember that $\alpha \in \{0, 1\}$) which is equal to $|\mathbf{d}'\rangle + (-1)^\alpha |\mathbf{d}\rangle$ since the global phase is not observable.

Remark 4.3.2 (Note on the abort). During a run of the GHZ-QFactory, Alice can abort: because δ is assumed to be negligible, this occurs with negligible probability if Bob is honest. However, it is surprisingly hard to show that revealing this additional bit of information (that we call the *abort bit*) does not harm the security of the protocol. For instance, Bob may have a way to maliciously sample y in such a way that an abort occurs if and only if, say, $\mathbf{d}_0 = 0 \ldots 0$. Revealing the abort bit would in that case leak

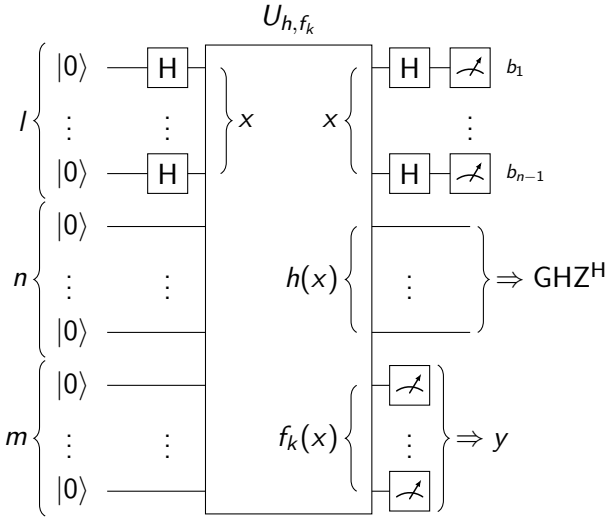


Figure 4.1: Circuit performed by the server Bob, where the superposition is created assuming $\mathcal{X} = \{0,1\}^n$.

information about $\mathbf{d}_0$. In our actual construction, it seems to be impossible to sample y in such a way, but so far we do not have an actual proof of security in that case.

To ensure that the abort bit does not leak to Bob, Alice would therefore need to behave exactly in the same way irrespective of whether the protocol aborted or not. One solution would be to ask to Alice to choose a random description $(\mathbf{d}, \mathbf{d}', \alpha)$ of a physical hidden GHZ state whose support $\mathbf{d} \oplus \mathbf{d}'$ is $\mathbf{d}_0$ and to continue like if there were no abort. This would basically be like saying that Alice sent $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle$ and Bob maliciously decided to discard this state. When δ is negligible the correctness is not impacted, but this is not the case when δ is not negligible: we will see how to mitigate this problem in Section 4.6.

Lemma 4.3.3 (Correctness of GHZ-QFactory)**.** *At the end of an honest run of the protocol GHZ-QFactory, with probability at least $1 - \delta$ (which can be made overwhelming), the state obtained by Bob is the hidden GHZ state $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle$ (where $\mathbf{d}, \mathbf{d}'$ and α are the outputs of Alice) and has support $\mathbf{d}_0$.*

Proof. First, one can easily see that the probability of measuring a y with 2 preimages is at least $1 - \delta$.

Indeed, before measuring the third register, Bob has the state $\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}} |x\rangle |h(x)\rangle |f_k(x)\rangle$ (we added back the normalization factor). But the probability of measuring a given y is $\left| (\mathbf{I} \otimes \mathbf{I} \otimes \langle y|) \left(\frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}} |x\rangle |h(x)\rangle |f_k(x)\rangle \right) \right|^2 = \frac{|f_k^{-1}(y)|}{|\mathcal{X}|}$. Therefore, if we denote by $A \coloneqq \left\{ y \;\middle|\; |f_k^{-1}(y)| = 2 \right\}$ the set of y having two preimages, the probability of

measuring a y having exactly two preimages is $\frac{1}{|\mathcal{X}|}\sum_{y\in A}|f_k^{(-1)}(y)| = \frac{2|A|}{|\mathcal{X}|}$. But this corresponds to the fraction of $x \in \mathcal{X}$ having two preimages, which is upper bounded by $1-\delta$ since we assumed f_k were δ-2-to-1.

Bob gets, after measuring the third register, the state $\sum_{x\in f^{-1}(y)}|x\rangle|h(x)\rangle$ on the first 2 registers. We saw that with probability $1-\delta$, y has two preimages x and x': in that case this state can therefore be rewritten as $\frac{1}{\sqrt{|\mathcal{X}|}}(|x\rangle|h(x)\rangle + |x'\rangle|h(x')\rangle)$. Then, we saw in Lemma 2.1.1 that for any bit string x, $\mathbf{H}^{\otimes l}|x\rangle = \frac{1}{\sqrt{2^l}}\sum_{b\in\{0,1\}^l}(-1)^{\langle b,x\rangle}|b\rangle$. Therefore, after applying the Hadamard gates (preparing the measurement in the Hadamard basis), we obtain the state:

$$(\mathbf{H}^{\otimes l}\otimes\mathbf{I})(|x\rangle|h(x)\rangle + |x'\rangle|h(x')\rangle) \tag{4.7}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{b\in\{0,1\}^l}(-1)^{\langle b,x\rangle}|b\rangle|h(x)\rangle + (-1)^{\langle b,x'\rangle}|b\rangle|h(x')\rangle \tag{4.8}$$

After measuring the first register, we obtain an outcome b and the second register in the computational basis contains

$$(-1)^{\langle b,x\rangle}|h(x)\rangle + (-1)^{\langle b,x'\rangle}|h(x')\rangle = (-1)^{\langle b,x\rangle}(|h(x)\rangle + (-1)^{\langle b,x\oplus x'\rangle}|h(x')\rangle) \tag{4.9}$$

due to the fact that

$$\langle b,x\rangle - \langle b,x'\rangle \bmod 2 = (\oplus_i b_i x_i)\oplus(\oplus_i b_i x'_i) = \oplus_i b_i(x_i\oplus x'_i) = \langle b, x\oplus x'\rangle \tag{4.10}$$

Note that one may be worried about the fact that if $\mathbf{d}_0 = 0\ldots0$, we may get $\mathbf{d} = \mathbf{d}'$ and $\alpha = -1$ and therefore the state would be $|\mathbf{d}\rangle - |\mathbf{d}\rangle = \mathbf{0}\ldots$ which is not a quantum state since it has norm 0. In an honest scenario, this is however not possible since the probability of measuring a y leading to $\alpha = -1$ is null. If Bob were malicious, this would have been possible, hence the test.

As expected, this state is equal to $|\mathbf{d}\rangle + (-1)^\alpha|\mathbf{d}'\rangle$ once we get rid of the global phase and define $\mathbf{d} \coloneqq h(x)$, $\mathbf{d}' \coloneqq h(x')$ and $\alpha = \langle b, x\oplus x'\rangle$. Moreover, by assumption $h(x)\oplus h(x') = \mathbf{d}_0$ so the support of this hidden GHZ state is $\mathbf{d}_0$. □

One can then wonder how to define the security of this construction. First, we cannot say that Bob has zero information about the hidden GHZ state $|\mathbf{d}\rangle + (-1)^\alpha|\mathbf{d}'\rangle\ldots$ because he has this state locally. So for instance, he could just measure it, and obtain either $\mathbf{d}$ or $\mathbf{d}'$ (but not both of them): this is unavoidable, and present even in the presence of a real, perfect quantum channel. However, we will see that what actually matters for the security of the upcoming protocols is that no information should leak

about the support $\mathbf{d}_0 \coloneqq \mathbf{d} \oplus \mathbf{d}'$ of that GHZ. For now, we define the security in the game-based framework, but we will see in Chapter 7 a simulation-based security proof when considering Zero-Knowledge Proofs on Quantum States.

We show now that if Bob is arbitrarily malicious, he cannot learn any information about the support $\mathbf{d}_0$ of the hidden GHZ state:

Lemma 4.3.4 (Security of GHZ-QFactory)**.** *If we define the game* IND-GHZ-QFactory *following the spirit of* IND-CPA *security, no non-uniform interactive QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *can win* IND-GHZ-QFactory *with probability better than* $\frac{1}{2} + \mathsf{negl}(\lambda)$.

$\texttt{IND-GHZ-QFactory}^{\mathcal{A}}_{\texttt{Gen}}(\lambda)$

1 : $(\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
4 : $(y, b, \tilde{c}) \leftarrow \mathcal{A}_2(k)$
5 : // No more interaction
6 : **return** $\tilde{c} = c$

Proof. The structure of the protocol—with a single round of messages and a secret support determined before the start of the protocol—allows us to have a direct link between its security and the assumptions we made on the family $\{f_k\}$ (this will be less direct for the last protocols we will consider). This is indeed a trivial reduction to the indistinguishability property: since y and b are not used, we can remove them without changing the probability of winning, and we get exactly the game IND-DO (Definition 4.2.1). This game is impossible to win for probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$ by assumption on the family $\{f_k\}$, which ends the proof. □

Note that we did not mention anything about what leaks about α. So far we do not provide any guarantee about α for two reasons. First, it turns out that in the protocols that are of interest for us, we do not need to obtain any guarantee on α. Secondly, Bob could fool Alice and make sure that she believes that $\alpha = 0$ by sending $b = 0 \dots 0$ (however, in that case Bob will not have the corresponding hidden GHZ). We may mitigate this last "attack" by forbiding Bob to return a string with to few ones for b, but it would complicate the protocol for no reasons since we never use this property.

4.4 Preparing Other Families of States

In most of the protocols, we are interested in producing single-qubit states. For instance, in the UBQC protocol we need to prepare random $|+_\theta\rangle$ with $\theta \in \{0, \frac{\pi}{4}, \dots, \frac{7\pi}{4}\}$. In this section we describe how to produce different families of states. Moreover, note that once we know how to produce $|+_\theta\rangle$ states, we can use the UBQC protocol to produce arbitrarily complicated states.

4.4.1 BB84 states

The BB84 states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\} = \left\{ \mathbf{H}^{B_1} |B_0\rangle \;\middle|\; (B_0, B_1) \in \{0,1\}^2 \right\}$[4]—whose name comes from Bennett and Brassard who discovered the famous Quantum Key Distribution protocol [BB84]—is the basic building block of many other protocols.

We present now a protocol BB84-QFactory preparing BB84 states, which is in fact a particular case of the GHZ-QFactory protocol. Note that we are also interested to prepare BB84 states since we use it as a starting point to remotely prepare $|+_\theta\rangle$ states: two runs of BB84-QFactory will be required to produce a single BB84 state [CCK+19]. In this thesis, we will also provide a more efficient method to generate $|+_\theta\rangle$ states using a single run of QFactory, but so far we are unable to prove its security in full generality. In any case, the construction presented in this section will also be useful in the Section 4.6 when considering δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable functions with a non-negligible δ (which will be the case when we will consider constructions based on LWE with polynomial noise ratio).

We describe in Protocol 3 the protocol BB84-QFactory, prove its correctness in Corollary 4.4.1 and its security in Lemma 4.4.3.

Corollary 4.4.1 (Correctness of BB84-QFactory)**.** *If Alice and Bob are honest, Bob gets a BB84 state* $\mathbf{H}^{B_1} |B_0\rangle$ *with overwhelming probability and Alice outputs* (B_1, B_0).

Proof. This is a corollary of Lemma 4.3.3. At the end of an honest run, Alice gets $|\psi\rangle := |\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle$, with $\mathbf{d} \oplus \mathbf{d}' = \mathbf{d}_0 = B_1$. Therefore, if $B_1 = 0$, $\mathbf{d} = \mathbf{d}'$ (and $\alpha = 1$, otherwise the protocol would have aborted) we have $|\psi\rangle = |\mathbf{d}\rangle$. Otherwise, $\mathbf{d} \neq \mathbf{d}'$: since we showed in Remark 4.3.1 that the order of $\mathbf{d}$ and $\mathbf{d}'$ we can assume that $\mathbf{d} = 0$ and $\mathbf{d}' = 1$. Then the state is $|0\rangle + (-1)^\alpha |1\rangle = \mathbf{H}^1 |\alpha\rangle$, which concludes the proof. □

It is now time to study the security of the protocol.

[4] The bit B_1 will be called the *basis bit* since it determines if the state belongs to $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$. On the other hand, B_0 is called the *value bit* since it determines which value is encoded into the basis fixed by B_1.

Protocol 3 BB84-QFactory

Assumptions: There exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1). Moreover, we only need this construction to work for strings $\mathbf{d}_0$ of size $n = 1$.
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's inputs: The basis $B_1 \in \{0, 1\}$ of the BB84 state (if $B_1 = 0$, we will prepare $|0\rangle$ or $|1\rangle$, otherwise we prepare $|+\rangle$ or $|-\rangle$).
Alice's outputs: The description (B_1, B_0) of the BB84 state obtained by Bob.
Bob's output: A BB84 state $\mathbf{H}^{B_1} |B_0\rangle$.
Protocol:
Run the GHZ-QFactory protocol (Protocol 2) between Alice and Bob, where Alice's input is a single bit $\mathbf{d}_0 = B_1$, and the output of Alice is denoted $(\mathbf{d}, \mathbf{d}', \alpha)$. If the protocol aborted, Alice aborts. Otherwise, if $B_1 = 0$ then Alice sets $B_0 \coloneqq \mathbf{d}$ and if $B_1 = 1$, Alice defines $B_0 \coloneqq \alpha$. Finally, Alice outputs (B_1, B_0).

Remark 4.4.2. Note that in the following protocols we will always focus on the security in term of *basis blindness*, meaning that no adversary can learn the basis in which the qubit is prepared, but we do not provide much guarantee on the security of the remaining "value bit" (denoting which of the two vectors of the basis we prepare). The reason for that is similar to the one we gave in the Section 4.3. First, this is the only security property that we need to obtain blind quantum computing. Secondly, we cannot prove that no information leaks about this value bit, simply because there are leaks even in the case of a perfect quantum channel: given the final qubit, it is always possible to pick a random basis $\{|\psi_1\rangle, |\psi_2\rangle\}$, measure the qubit in this basis obtaining an outcome b, and claim that the initial qubit was not $|\psi_{1-b}\rangle$ (we managed to rule out one of the outputs). By doing diagonal measurements, we could also obtain more information about this value bit, even in the case of a perfect quantum channel.

That said, we can still claim something about the value bit in our setting: it is not possible to have information about it without altering the output state. Indeed, if it were possible, then we could obtain information about the basis (and we will show that it is not possible): by picking a random basis and measuring our qubit in this basis, we can check if the outcome is compatible with our guessed value bit. If it's not, and if we guessed correctly the value bit, the qubit could not have been prepared in this basis: We have learnt information about the basis since we have ruled out one basis. Of course, this is only a sketch of proof, but we won't go any further since in our applications we only need basis blindness.

We show now that if Bob is arbitrarily malicious, he cannot learn any information about the basis B_1 of the hidden GHZ state:

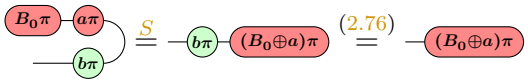
Lemma 4.4.3 (Security of BB84-QFactory)**.** *If we define the game IND-BB84-QFactory following the spirit of* IND-CPA *security, no non-uniform interactive QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *can win IND-BB84-QFactory with probability better than* $\frac{1}{2} + \mathsf{negl}(\lambda)$.

$\texttt{IND-BB84-QFactory}^{\mathcal{A}}_{\texttt{Gen}}(\lambda)$

1 : $(B_1^{(0)}, B_1^{(0)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{B_1^{(c)}})$
4 : $(y, b, \tilde{c}) \leftarrow \mathcal{A}_2(k)$
5 : // No more interaction
6 : **return** $\tilde{c} = c$

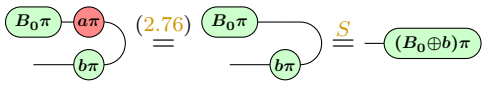
Proof. This is a special case of Lemma 4.3.4 where $\mathbf{d}_0$ is a single bit. □

Remark 4.4.4 (Hiding a measurement). Note that we can turn the BB84-QFactory protocol (that prepares a state unknown to the server) into a protocol that performs a measurement whose basis (computational or Hadamard) is unknown to the server. Indeed, if we perform a Bell measurement between the output of the BB84-QFactory protocol and an input qubit, we actually perform a measurement in the computational basis on this second qubit if $B_1 = 0$ and a measurement in the Hadamard basis otherwise. Indeed, if the basis of the prepared qubit is $B_1 = 0$, we have:



(4.11)

otherwise if $B_1 = 1$:



(4.12)

4.4.2 Producing $|+_\theta\rangle$ (and more) from BB84-QFactory

In order to run the UBQC protocol to do blind quantum computing, we need to produce random $|+_\theta\rangle$ states with $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$ (we will use the notation $\mathbb{Z}_{\frac{\pi}{4}} \coloneqq \{0, \frac{\pi}{4}, \ldots, \frac{7\pi}{4}\}$, and may be pronounced *8 states* when used inside protocols). In this section we explain how to produce $|+_\theta\rangle$ states from two BB84 states produced using the BB84-QFactory protocol. We describe in Protocol 3 the protocol BB84-QFactory.

Intuitively, what this protocol does is that it generates two BB84 states using the protocol we just studied, and it uses a gadget circuit to combine them. More precisely, this gadget circuit will rotate one BB84 to obtain a $|+_{\theta_0}\rangle$ state (it has only four possible angles for now: $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$), and it will entangle this state to a $|+_{\frac{\pi}{4}}\rangle$ state... with our second BB84 state in the middle of the entanglement. The role of this second BB84 state is to cut the entanglement when it is equal to $|0\rangle$ or $|1\rangle$, and preserve it otherwise (the adversary does not know the basis of this state, so it cannot know if the entanglement is preserved or not). Then, when measuring these two qubits in the appropriate basis, what happens is that if the entanglement was preserved, $|+_{\frac{\pi}{4}}\rangle$ will basically be teleported on $|+_{\theta_0}\rangle$, resulting in a new state whose angle is roughly the sum of θ_0 and $\frac{\pi}{4}$ (up to some additional terms). On the other hand, if the entanglement was cut, $\frac{\pi}{4}$ will not be added to θ_0. Depending on whether we add or not $\frac{\pi}{4}$ to θ_0, we are therefore able to reach height possible states, with a new angle in $\mathbb{Z}_{\frac{\pi}{4}}$.

Note that this same idea could be extended to also produce $|0\rangle$ or $|1\rangle$ states (used by some verification protocols [FK17, KW15]) by chaining another BB84 state, but we do not need that for blind quantum computing, so we will not study it in this thesis.

Protocol 4 $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory

Assumptions: There exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1). However, we only need this construction to work for strings $\mathbf{d}_0$ of size $n = 1$.
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's outputs: The description $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$ of the state obtained by Bob.
Bob's output: A quantum state $|+_\theta\rangle$.
Protocol:

1. Alice randomly samples $(B_1^{(0)}, B_1^{(1)}) \xleftarrow{\$} \{0,1\}^2$.
2. Run two times the BB84-QFactory protocol (Protocol 3) between Alice and Bob with the inputs $B_1^{(0)}$ and $B_1^{(1)}$ for the respective protocols. Bob gets two states $|\mathtt{in}^{(0)}\rangle = \mathbf{H}^{B_1^{(0)}}|B_0^{(0)}\rangle$ and $|\mathtt{in}^{(1)}\rangle = \mathbf{H}^{B_1^{(1)}}|B_0^{(1)}\rangle$, and Alice has the corresponding description $(B_1^{(0)}, B_0^{(0)})$ and $(B_1^{(1)}, B_0^{(1)})$. In case one or two runs abort, Alice continues the protocol but randomly samples the missing $B_0^{(i)}$'s in $\{0,1\}$.
3. Bob runs the "gadget" circuit pictured in Figure 4.2, and sends the measurement outcomes s_1 and s_2 to Alice. Bob outputs the remaining qubit.
4. Alice outputs θ where

$$\theta := \pi(B_0^{(0)} + B_0^{(1)} + s_1 B_1^{(0)}) + \frac{\pi}{2}(B_1^{(1)} + B_1^{(0)} B_0^{(0)} - s_2 B_1^{(0)}) + \frac{\pi}{4} B_1^{(0)} \tag{4.13}$$

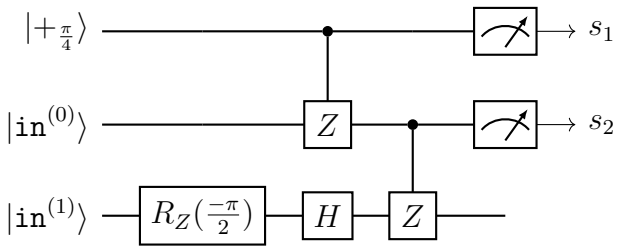
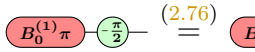
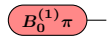
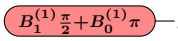
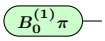


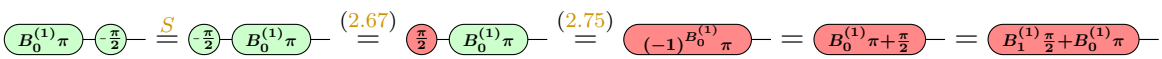
Figure 4.2: Gadget circuit needed by $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory.

Theorem 4.4.5 (Correctness of $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory)**.** *The protocol $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory (Protocol 4) is correct, in the sense that if δ is negligible, with overwhelming probability Bob outputs a state $|+_\theta\rangle$ and Alice outputs θ.*

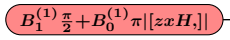
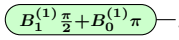
Proof. First, we remark that the abort probability of one BB84-QFactory protocol is $\delta = \mathsf{negl}(\lambda)$, so the probability that one of the run aborts is negligible. If they do not abort, then we can see that applying $\mathbf{R}_z(\frac{-\pi}{2})$ on $|\mathtt{in}\rangle = \mathbf{H}^{B_1^{(1)}}|B_0^{(1)}\rangle$ gives:

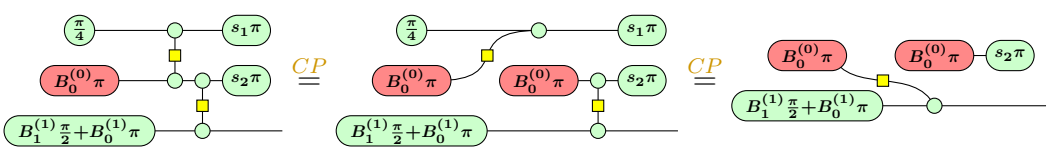
$$\mathbf{R}_z\left(\frac{-\pi}{2}\right)|\mathtt{in}\rangle = \tag{4.14}$$

We do two cases. If $B_1^{(1)} = 0$, then $|\mathtt{in}\rangle =$  , but  $\overset{(2.76)}{=}$  $=$  . If $B_1^{(1)} = 1$, then $|\mathtt{in}\rangle =$  so

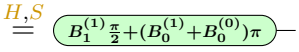


(4.15)

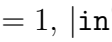
Therefore, $\mathbf{HR}_z\left(\frac{-\pi}{2}\right)|\mathtt{in}\rangle =$  $=$  . Then, if $B_1^{(0)} = 0$, $|\mathtt{in}^{(0)}\rangle = |B_0^{(0)}\rangle$, the output of the circuit is $|+_{\theta_a}\rangle$ with $\theta_a := B_1^{(1)}\frac{\pi}{2} + (B_0^{(1)} + B_0^{(0)})\pi$:

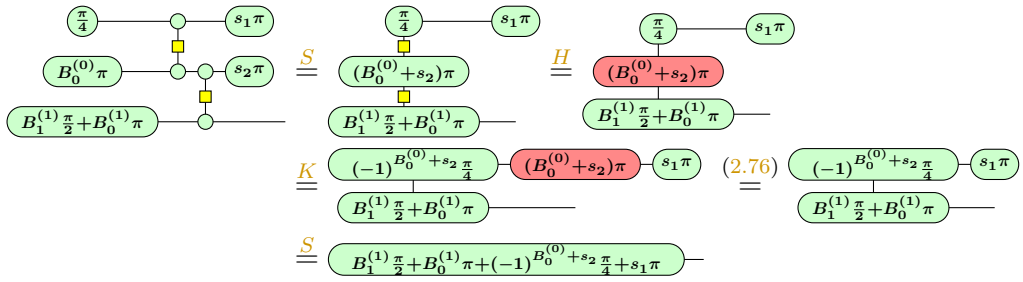


(4.16)



(4.17)

Otherwise, if $B_1^{(0)} = 1$, $|\mathtt{in}^{(0)}\rangle = \mathbf{H}\,|B_0^{(0)}\rangle =$  , and the output of the circuit is $|+_{\theta_b}\rangle$ with $\theta_b \coloneqq B_1^{(1)}\frac{\pi}{2} + B_0^{(1)}\pi + (-1)^{B_0^{(0)}+s_2}\frac{\pi}{4} + s_1\pi$:



(4.18)

(4.19)

(4.20)

To obtain a value of θ that works for both values of $B_1^{(0)}$, we can just compute:

$$(1 - B_1^{(0)})\theta_a + B_1^{(0)}\theta_b \tag{4.21}$$
$$= (1 - B_1^{(0)})(B_1^{(1)}\frac{\pi}{2} + (B_0^{(0)} + B_0^{(1)})\pi) + B_1^{(0)}(B_1^{(1)}\frac{\pi}{2} + B_0^{(1)}\pi + (-1)^{B_0^{(0)}+s_2}\frac{\pi}{4} + s_1\pi)$$

Using $a(b+c) + (1-a)(b+d) = b + ac + (1-a)d$:

$$= B_1^{(1)}\frac{\pi}{2} + B_0^{(1)}\pi + (1 - B_1^{(0)})B_0^{(0)}\pi + B_1^{(0)}(-1)^{B_0^{(0)}+s_2}\frac{\pi}{4} + B_1^{(0)}s_1\pi \tag{4.22}$$

Developing one term and using $(-1)^a\pi/4 = \pi/4 - a\pi/2$:

$$= B_1^{(1)}\frac{\pi}{2} + B_0^{(1)}\pi + B_0^{(0)}\pi - \underline{B_1^{(0)}B_0^{(0)}\pi} + B_1^{(0)}\frac{\pi}{4} - \underline{B_1^{(0)}(B_0^{(0)} + s_2)\frac{\pi}{2}} + B_1^{(0)}s_1\pi \tag{4.23}$$

Using $\underline{-a\pi - a\pi/2 = a\pi/2}$:

$$= B_1^{(1)}\frac{\pi}{2} + B_0^{(1)}\pi + B_0^{(0)}\pi + \underline{B_1^{(0)}B_0^{(0)}\frac{\pi}{2}} + B_1^{(0)}\frac{\pi}{4} - B_1^{(0)}s_2\frac{\pi}{2} + B_1^{(0)}s_1\pi \tag{4.24}$$

Grouping:

$$= \pi(B_0^{(0)} + B_0^{(1)} + s_1B_1^{(0)}) + \frac{\pi}{2}(B_1^{(1)} + B_1^{(0)}B_0^{(0)} - s_2B_1^{(0)}) + \frac{\pi}{4}B_1^{(0)} = \theta \tag{4.25}$$

which concludes the proof. □

We study now the security of the protocol. As already discussed in Remark 4.4.2, we are only interested in basis blindness. We show that if Bob is arbitrarily malicious, then the protocol $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory (Protocol 4) is *basis blind*, meaning that he cannot learn any information about the basis of the state produced by the protocol:

Theorem 4.4.6 (Security of $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory)**.** *No interactive non-uniform QPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *can win* $\mathtt{IND\text{-}\mathbb{Z}_{\frac{\pi}{4}}\text{-}QFactory}$ *with probability better than* $\frac{1}{4} + \mathsf{negl}(\lambda)$.

$\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}^{\mathcal{A}}(\lambda)$

1 : $(\theta, \tilde{\theta}_\pi) \leftarrow (\mathsf{Alice}_\lambda \leftrightsquigarrow \mathcal{A})$

2 : **return** $\tilde{\theta}_\pi = \theta \bmod \pi$

where Alice_λ *is the honest party of* $\mathbb{Z}_{\frac{\pi}{4}}$*-QFactory.*

Proof. In order to prove the above statement, we will assume that there exists an adversary $\mathcal{A}$ able to guess $\theta \bmod \pi$ with probability better than $\frac{1}{4} + \mathsf{poly}(\lambda)$, then we will say that $\mathcal{A}$ is good at determining at least one bit of $\theta \bmod \pi$, and depending on the bit that $\mathcal{A}$ manages to guess correctly, we will use it to break the security of BB84-QFactory.

First, we will decompose $\theta = \theta_0\pi + \theta_1\frac{\pi}{2} + \theta_2\frac{\pi}{4}$ where $(\theta_0, \theta_1, \theta_2) \in \{0,1\}^3$ (since Alice is honest, θ is a multiple of $\frac{\pi}{4}$), and similarly we decompose $\tilde{\theta}_\pi = \tilde{\theta}_{\pi,1}\frac{\pi}{2} + \tilde{\theta}_{\pi,2}\frac{\pi}{4}$. Note that we have:

$$\theta \coloneqq \pi(B_0^{(0)} + B_0^{(1)} + s_1 B_1^{(0)}) + \frac{\pi}{2}(B_1^{(1)} + B_1^{(0)}B_0^{(0)} - s_2 B_1^{(0)}) + \frac{\pi}{4}B_1^{(0)} \tag{4.26}$$

using the notation from the protocol $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory and therefore:

$$\theta_1 = B_1^{(1)} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) \qquad \theta_2 = B_1^{(0)} \tag{4.27}$$

The goal of the attacker is therefore to find the bits θ_1 and θ_2. Our first claim is that $\mathcal{A}$ can guess either θ_1, θ_2 or $\theta_1 \oplus \theta_2$ with probability above $\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$. This is a direct consequence of the following Lemma, by defining $X = \mathsf{Alice}_\lambda \leftrightsquigarrow \mathcal{A}_\lambda$:

Lemma 4.4.7 (Implication of guessing two predicates)**.**
Let X *be a probability distribution outputting* 4 *bits* $((a,b),(\tilde{a},\tilde{b})) \in \{0,1\}^2 \times \{0,1\}^2$*—intuitively,* $\tilde{a}$ *is the guess of the variable* a*, same for* $\tilde{b}$*—such that the probability of guessing* (a,b) *is good, i.e.* $\Pr\left[(a,b) = (\tilde{a},\tilde{b})\right] \geq 1/4 + \frac{1}{\mathsf{poly}(\lambda)}$*. Then at least one of these properties is true:*

- *a is guessed with good probability, i.e.:*
  $P_1 \coloneqq \Pr\left[\tilde{a} = a \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right] \geq 1/2 + 1/\mathsf{poly}(\lambda)$
- *b is guessed with good probability, i.e.:*
  $P_2 \coloneqq \Pr\left[\tilde{b} = b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right] \geq 1/2 + 1/\mathsf{poly}(\lambda)$
- *the* XOR *of a and b is guessed with good probability, i.e.:*
  $P_\oplus \coloneqq \Pr\left[\tilde{a} \oplus \tilde{b} = a \oplus b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right] \geq 1/2 + 1/\mathsf{poly}(\lambda)$

We can prove this Lemma as follows. Let us denote by:

- $e_1 = \Pr\left[\tilde{a} \neq a \text{ and } \tilde{b} \neq b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right]$
- $e_2 = \Pr\left[\tilde{a} = a \text{ and } \tilde{b} \neq b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right]$

- $e_3 = \Pr\left[\tilde{a} \neq a \text{ and } \tilde{b} = b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right]$
- $e_4 = \Pr\left[\tilde{a} = a \text{ and } \tilde{b} = b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right]$

By assumption, we know that the probability to do a correct guess is good, i.e. $e_4 \geq \frac{1}{4} + \frac{1}{\mathsf{poly}(\lambda)}$. Now, we assume that $\mathcal{A}$ is bad at guessing both a and b, i.e. $e_2 + e_4 \leq \frac{1}{2} + \mathsf{negl}(\lambda)$ and $e_3 + e_4 \leq \frac{1}{2} + \mathsf{negl}(\lambda)$, and we show that $\mathcal{A}$ is good to guess the XOR of aa and b. Because $e_4 \geq \frac{1}{4} + \frac{1}{\mathsf{poly}(\lambda)}$, we have $e_2 \leq \frac{1}{4} - \frac{1}{\mathsf{poly}(\lambda)}$ and $e_3 \leq \frac{1}{4} - \frac{1}{\mathsf{poly}(\lambda)}$. So $e_2 + e_3 \leq \frac{1}{2} - \frac{1}{\mathsf{poly}(\lambda)}$, and because $e_1 + e_2 + e_3 + e_4 = 1$, we get $e_1 + e_4 \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$. But $e_1 + e_4$ is exactly the probability to guess the XOR, i.e.

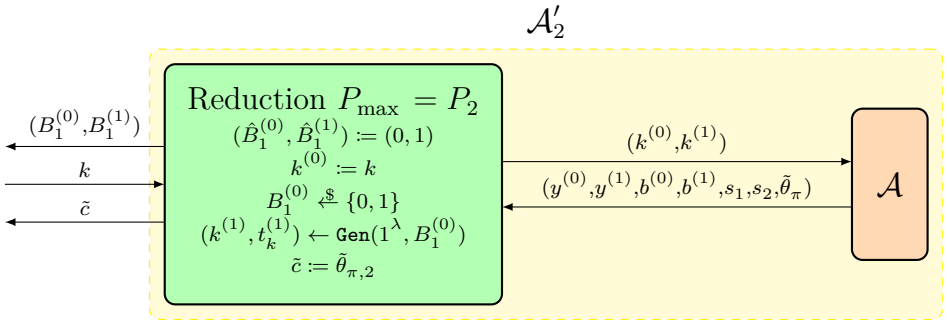
$$\Pr\left[\tilde{a} \oplus \tilde{b} = a \oplus b \mid ((a,b),(\tilde{a},\tilde{b})) \leftarrow X\right] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.28}$$

which concludes the proof of the Lemma.

We define now P_1 the probability for $\mathcal{A}$ to guess θ_1, P_2 the probability of guessing θ_2, $P_\oplus$ the probability of guessing $\theta_1 \oplus \theta_2$, and $P_{\max} = \max(P_1, P_2, P_\oplus)$. Depending on $P_{\max}$, we will derive now three different reductions able to break the security of BB84-QFactory.

Note that this proof is not constructive in the sense that we only say "there exists a method to break the security of BB84" without specifying which of the three is the appropriate one. It does not really matter in our case since we only care about security, however someone puzzled by this non-constructive proof should be able to turn it into a constructive proof by adding a step in which $P_{\max}$ is estimated by evaluating $\mathcal{A}$ on known inputs.

First case: $P_{\max} = P_2$. In this case, we define the reduction from IND-BB84-QFactory to IND-$\mathbb{Z}\frac{\pi}{4}$-QFactory as follows, in order to obtain an adversary $\mathcal{A}'_2$ able to win IND-BB84-QFactory.



Note that this reduction intuitively uses the fact that the $\frac{\pi}{4}$ component of the angle given by the adversary is often correct (because $P_{\max} = P_2$) and that when it is correct, it is equal to the basis $B_1^{(0)}$ of the first BB84 state $|\mathtt{in}^{(0)}\rangle$: but this basis is supposed to be impossible to find, so such an adversary cannot exist.

We formalize this intuition and prove now that $\Pr\left[\texttt{IND-BB84-QFactory}^{\mathcal{A}'_2}_{\texttt{Gen}}(\lambda)\right] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$, which leads to a contradiction.

In the following, we will have probability distributions over runs of IND-$\mathbb{Z}_4^\pi$-QFactory and probability distributions over runs of IND-BB84-QFactory with our adversary $\mathcal{A}'_2$. To make the distinction clear, we will use the notation $\Pr_A[\cdot]$ to denote a probability over a run of IND-$\mathbb{Z}_4^\pi$-QFactory and $\Pr_R[\cdot]$ for a probability over IND-BB84-QFactory with our adversary $\mathcal{A}'_2$. We may use inside $B_1^{(b)}$ to refer to the basis encrypted into the bth message sent to $\mathcal{A}$.

First, we can remark that no matters the number of preimages of $y^{(0)}$ and $y^{(1)}$, Alice always outputs θ such that $\theta_2 = B_1^{(0)}$ (you can observe that the output of Alice when an abort occurs is chosen to simplify the proof). Therefore since our reduction with the game IND-BB84-QFactory exactly reproduces the behavior of Alice, we have:

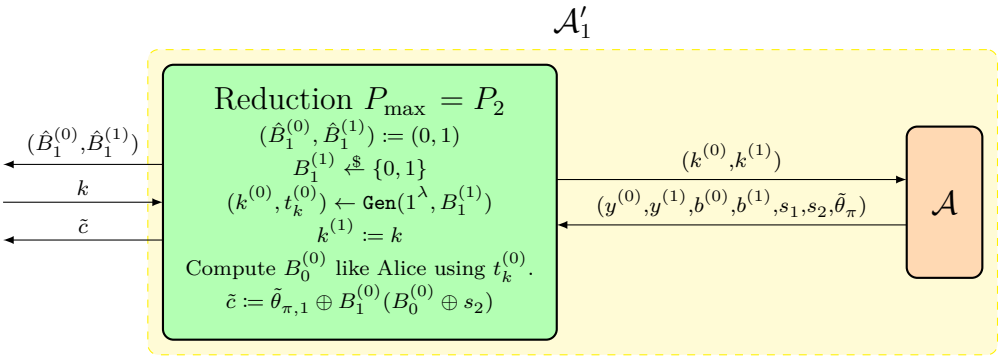
$$\Pr_R\left[\tilde{\theta}_{\pi,2} = B_1^{(0)}\right] \overset{(4.27)}{=} \Pr_A\left[\tilde{\theta}_{\pi,2} = \theta_2\right] = P_2 = P_{\max} \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.29}$$

But the reduction was chosen in such a way that $B_1^{(0)} = \hat{B}_1^{(c)} = c$ and $\tilde{c} = \tilde{\theta}_{\pi,2}$. Therefore Eq. (4.29) can be turned into

$$\Pr_R[\tilde{c} = c] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.30}$$

Hence, the probability for $\mathcal{A}'_2$ to win the game IND-BB84-QFactory is greater than $\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$ which is in contradiction with Lemma 4.4.3, so $\mathcal{A}'_0$ cannot exist.

Second case: $P_{\max} = P_1$. In this case, we define the reduction from IND-BB84-QFactory to IND-$\mathbb{Z}_4^\pi$-QFactory as follows, in order to obtain an adversary $\mathcal{A}'_1$ able to win IND-BB84-QFactory when $P_{\max} = P_1$.



Note that this reduction intuitively uses the fact that the $\frac{\pi}{2}$ component of the angle given by the adversary is often correct (because $P_{\max} = P_1$) and that when it is correct,

it is equal—up to some terms that depend only on the first state that can be created inside the reduction—to the basis $B_1^{(1)}$ of the second BB84 state $|\mathtt{in}^{(1)}\rangle$: but this basis is supposed to be impossible to find, so such an adversary cannot exist.

We formalize this intuition and prove now that:

$$\Pr\left[\texttt{IND-BB84-QFactory}^{\mathcal{A}'_1}_{\mathtt{Gen}}(\lambda)\right] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.31}$$

which leads to a contradiction.

As before, we can remark that IND-BB84-QFactory and our reduction is performing exactly the same task as Alice in IND-$\mathbb{Z}_{\frac{\pi}{4}}$-QFactory, so we have:

$$\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \leq \Pr_A\left[\tilde{\theta}_{\pi,1} = \theta_1\right] \overset{(4.27)}{=} \Pr_R\left[\tilde{\theta}_{\pi,1} = B_1^{(c)} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2)\right] \tag{4.32}$$

where the first inequality comes from $P_{\max} = P_1$. Moreover, $\tilde{c} = \tilde{\theta}_{\pi,1} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2)$ and $B_1^{(c)} = c$ so:

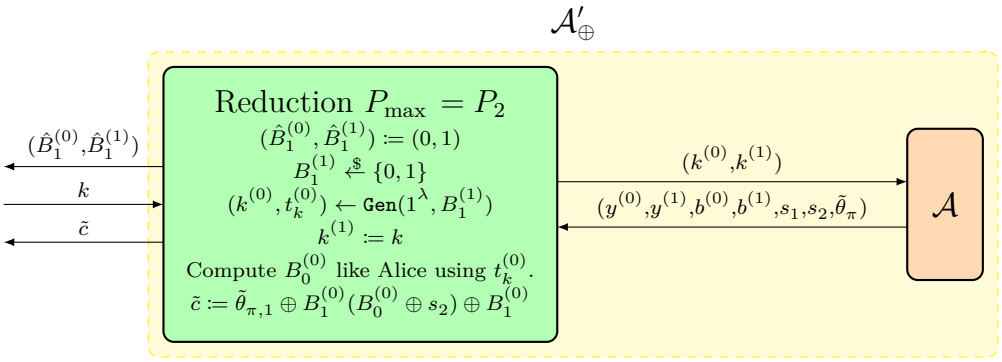
$$\Pr_R[\tilde{c} = c] = \Pr_R\left[\tilde{\theta}_{\pi,1} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) = B_1^{(c)}\right] \tag{4.33}$$

$$= \Pr_R\left[\tilde{\theta}_{\pi,1} = B_1^{(c)} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2)\right] \tag{4.34}$$

$$\overset{(4.37)}{\geq} \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.35}$$

Hence, the probability for $\mathcal{A}'_1$ to win the game IND-BB84-QFactory is greater than $\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$ which is in contradiction with Lemma 4.4.3, so $\mathcal{A}'_1$ cannot exist.

Third case: $P_{\max} = P_{\oplus}$. In this case, we define the reduction from IND-BB84-QFactory to IND-$\mathbb{Z}_{\frac{\pi}{4}}$-QFactory as follows, in order to obtain an adversary $\mathcal{A}'_{\oplus}$ able to win IND-BB84-QFactory when $P_{\max} = P_{\oplus}$.



Note that this reduction intuitively uses the fact that the $\frac{\pi}{2} \oplus \frac{\pi}{4}$ component of the angle given by the adversary is often correct (because $P_{\max} = P_{\oplus}$) and that when it is correct,

it is equal—up to some terms that depend only on the first state that can be created inside the reduction—to the basis $B_1^{(1)}$ of the second BB84 state $|\mathtt{in}^{(1)}\rangle$, nearly exactly like in the second case. Because this basis is supposed to be impossible to find, so such an adversary cannot exist.

We can prove, using a proof very similar to the second case that

$$\Pr\left[\texttt{IND-BB84-QFactory}^{\mathcal{A}'_1}_{\mathtt{Gen}}(\lambda)\right] \geq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.36}$$

leading to a contradiction.

As before, we can remark that IND-BB84-QFactory and our reduction is performing exactly the same task as Alice in IND-$\mathbb{Z}_{\frac{\pi}{4}}$-QFactory, so we have:

$$\begin{aligned} &\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \\ &\leq \Pr_A\left[\tilde{\theta}_{\pi,1} \oplus \tilde{\theta}_{\pi,2} = \theta_1 \oplus \theta_2\right] \overset{(4.27)}{=} \Pr_R\left[\tilde{\theta}_{\pi,1} = B_1^{(c)} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) \oplus B_1^{(0)}\right] \end{aligned} \tag{4.37}$$

where the first inequality comes from $P_{\max} = P_{\oplus}$. Moreover, $\tilde{c} = \tilde{\theta}_{\pi,1} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) \oplus B_1^{(0)}$ and $B_1^{(c)} = c$ so:

$$\Pr_R[\tilde{c} = c] = \Pr_R\left[\tilde{\theta}_{\pi,1} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) \oplus B_1^{(0)} = B_1^{(c)}\right] \tag{4.38}$$

$$= \Pr_R\left[\tilde{\theta}_{\pi,1} = B_1^{(c)} \oplus B_1^{(0)}(B_0^{(0)} \oplus s_2) \oplus B_1^{(0)}\right] \tag{4.39}$$

$$\overset{(4.37)}{\geq} \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.40}$$

Hence, the probability for $\mathcal{A}'_{\oplus}$ to win the game IND-BB84-QFactory is greater than $\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$ which is in contradiction with Lemma 4.4.3, so $\mathcal{A}'_{\oplus}$ cannot exist.

We have therefore covered all three cases: all of them lead to a contradiction, confirming that there exist no adversary $\mathcal{A}$ able to break the game IND-$\mathbb{Z}_{\frac{\pi}{4}}$-QFactory, which concludes the proof.. □

We have therefore saw a method to prepare $|+_\theta\rangle$ states using classical communication without revealing to the server the basis θ mod π of θ. This turns out to be enough to run the UBQC protocol as we will see in Section 4.5, allowing us to create arbitrarily complicated states.

4.5 Application to Classical-Client Blind Quantum Computing

One of the main application of the QFactory protocols is classical-client blind quantum computing. While we will show in Chapter 6 that classical-client UBQC ($\mathsf{UBQC}_{\mathsf{CC}}$) protocols cannot be proven secure in a fully composable setting, there is hope that it remains possible with a weaker definition of security. And indeed, in this section we show that $\mathsf{UBQC}_{\mathsf{CC}}$ is possible in the game-based setting by combining the UBQC protocol ([BFK09], see details in Section 2.3.7) with our protocol $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory. We first give in Protocol 5 the definition of the $\mathsf{UBQC}_{\mathsf{CC}}$ protocol, then we define and prove its security.

Protocol 5 $\mathsf{UBQC}_{\mathsf{CC}}$: Classical Blind Quantum Computing

Requirements: There exists a classical-client Remote State Preparation protocol $\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}$ producing $|+_\theta\rangle$ states with $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$ with overwhelming probability during an honest run. Moreover, this protocol should be basis-blind in the sense of IND-$\mathbb{Z}_{\frac{\pi}{4}}$-QFactory (this is the case of our protocol $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory whose security is proven in Theorem 4.4.6).
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's inputs: A circuit—where inputs are hardcoded and all output qubits will be measured in the computational basis—represented as a MBQC pattern where we denote by $\{\phi_i\}_{i\in[n]}$ the set of measurement angles on the graph G. This pattern can be obtained using for instance the brickwork construction described in Section 2.3.6.
Alice's outputs: The measurement outcomes of the circuit.
Protocol:

1. Alice and Bob run n different instances of $\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}$ (in parallel) to obtain $\{\theta_i\}_{i\in[n]}$ on Alice's side and $\{|+_{\theta_i}\rangle\}_{i\in[n]}$ on Bob's side, where for all i, $\theta_i \leftarrow \mathbb{Z}_{\frac{\pi}{4}}$.
2. Alice and Bob run the UBQC protocol (Definition 2.3.4), except that Bob uses the $|+_{\theta_i}\rangle$ obtained at the previous step. Alice forwards the output.

Definition 4.5.1 (Blindness of $\mathsf{UBQC}_{\mathsf{CC}}$)**.** *A classical-client UBQC protocol $\boldsymbol{\pi} = (\pi_A, \pi_B)$ is said to be (computationally)* blind *if no computationally bounded malicious server can distinguish between runs of the protocol with adversarially chosen circuits (i.e. the angle of the measurement pattern on the MBQC graph).*

In formal terms, $\boldsymbol{\pi}$ is said to be (computationally) blind if and only if no interactive QPT adversary $\mathcal{A}$ can win the game IND-UBQC$_{\mathsf{cc}}$ with probability better then $1/2+\mathsf{negl}(\lambda)$, where the size of the pattern has polynomial size in λ:

$$\Pr\left[\textit{IND-UBQC}_{\mathsf{cc}}{}^{\mathcal{A}}(\lambda)\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

$\texttt{IND-UBQC}_{\mathsf{cc}}^{\mathcal{A}}(\lambda)$

1 : $(\phi^{(1)}, \phi^{(2)}) \leftarrow \mathcal{A}$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : // For brevity we omit the rest of the pattern which is fixed.
4 : $\pi_A(\phi^{(c)}) \leftrightsquigarrow \mathcal{A}$
5 : $\tilde{c} \leftarrow \mathcal{A}$
6 : **return** $c = \tilde{c}$

We prove now that protocol $\mathsf{UBQC}_{\mathsf{CC}}$ allows secure classical-client blind quantum computing.

Theorem 4.5.2 (Game-based Blindness of $\mathsf{UBQC}_{\mathsf{CC}}$)**.** *The protocol $\mathit{UBQC}_{\mathit{CC}}$ (instantiated with any $\mathbb{Z}_{\frac{\pi}{4}} - \mathit{RSP}_{\mathit{CC}}$ protocol fulfilling the requirements of the protocol, which includes our $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory protocol) is blind according to the definition Definition 4.5.1.*

Proof. The proof of Theorem 4.5.2 which will be given in the remainder of this section follows three main ideas:

1. First, we proceed by an induction on the size of the graph G. At every step, we will be able to show that the last round of communication does not bring a significant advantage to the adversary, allowing us to remove it. That way, we derive a series of games with similar winning probabilities, in such a way that the last game ends up to contain no communication at all with the adversary... and that is therefore trivially secure.

2. Secondly, in order to remove one round of communication, we first realize that the one-time pad r in UBQC hides all potential leakages on the value bit of the pattern angles. Therefore, the adversary can only learn information about the basis of the computation.

3. Lastly, it is fortunately also impossible to learn any information about the basis of the pattern angles, or we could use this information to find the basis of the underlying $\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}$ protocol... which is supposed to be impossible since this protocol is basis-blind by assumption.

More formally, we define for any $j \in \{0, \ldots, n\}$ the following games $\texttt{Game}j$ and $\texttt{Game}j'$. First, $\texttt{Game}j$ is basically like $\texttt{IND-UBQC}_{\mathsf{cc}}$ except that we stop the protocol after j rounds:

$\texttt{Gamej}^{\mathcal{A}}$

	Challenger		**Adversary**
1 :	$c \xleftarrow{\$} \{0,1\}, \forall i, \phi_i \coloneqq \phi_i^{(c)}$	$\xleftarrow{\phi^{(1)}, \phi^{(2)}}$	Choose $\phi^{(1)}, \phi^{(2)} \in \mathbb{Z}\frac{\pi}{4}$
2 :	Run n $\mathbb{Z}\frac{\pi}{4} - \mathsf{RSP}_{\mathsf{CC}}$ (as Alice) to obtain $\{\theta_i\}_{i\in[n]}$	$\xleftrightarrow{\mathbb{Z}\frac{\pi}{4} - \mathsf{RSP}_{\mathsf{CC}}}$	
3 :	$\delta_1 = (-1)^{f_1(\dots)}\phi_1 + r_1\pi + \theta_1 + f_2(\dots)\pi$	$\xrightarrow{\delta_1}$	
4 :		$\xleftarrow{s_1}$	
5 :	$\vdots$		
6 :	$\delta_j = (-1)^{f_1(\dots)}\phi_j + r_j\pi + \theta_j + f_2(\dots)\pi$	$\xrightarrow{\delta_j}$	
7 :	**return** $\tilde{c} = c$	$\xleftarrow{s_j, \tilde{c}}$	

And $\texttt{Gamej}'$ is exacly like $\texttt{Gamej}$, except that we add a dummy round at the end where δ_{j+1} is sampled uniformly at random.

$\texttt{Gamej}'^{\mathcal{A}}$

	Challenger		**Adversary**
1 :	$c \xleftarrow{\$} \{0,1\}, \forall i, \phi_i \coloneqq \phi_i^{(c)}$	$\xleftarrow{\phi^{(1)}, \phi^{(2)}}$	Choose $\phi^{(1)}, \phi^{(2)} \in \mathbb{Z}\frac{\pi}{4}$
2 :	Run n $\mathbb{Z}\frac{\pi}{4} - \mathsf{RSP}_{\mathsf{CC}}$ (as Alice) to obtain $\{\theta_i\}_{i\in[n]}$	$\xleftrightarrow{\mathbb{Z}\frac{\pi}{4} - \mathsf{RSP}_{\mathsf{CC}}}$	
3 :	$\delta_1 = (-1)^{f_1(\dots)}\phi_1 + r_1\pi + \theta_1 + f_2(\dots)\pi$	$\xrightarrow{\delta_1}$	
4 :		$\xleftarrow{s_1}$	
5 :	$\vdots$		
6 :	$\delta_j = (-1)^{f_1(\dots)}\phi_j + r_j\pi + \theta_j + f_2(\dots)\pi$	$\xrightarrow{\delta_j}$	
7 :		$\xleftarrow{s_j}$	
8 :	$\delta_{j+1} \xleftarrow{\$} \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$	$\xrightarrow{\delta_j}$	
9 :	**return** $\tilde{c} = c$	$\xleftarrow{s_{j+1}, \tilde{c}}$	

Clearly, for any j, the best probability of winning these games is the same:

$$\sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamej}^{\mathcal{A}}\right] = \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamej}'^{\mathcal{A}}\right] \tag{4.41}$$

Indeed, we can easily turn any adversary winning one game into an adversary winning the other game by removing/sampling δ_{n+1} since it does not depend on any secret. Now, we prove that for any $j \in [n-1]$, $\sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamej+1}^{\mathcal{A}}\right] \leq \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamej}'^{\mathcal{A}}\right] + \mathsf{negl}$.

> To that end, let $\mathcal{A}$ be a QPT adversary of $\texttt{Gamej}'$, and let us prove that there exists a QPT adversary $\mathcal{A}'$ such that $\Pr\left[\texttt{Gamej+1}\right]^{\mathcal{A}} \leq \Pr\left[\texttt{Gamej}'\right]^{\mathcal{A}'} + \mathsf{negl}$. First, if

$\Pr\left[\texttt{Gamej+1}\right]^{\mathcal{A}} \leq \Pr\left[\texttt{Gamej}'\right]^{\mathcal{A}} + \mathsf{negl}$, we just take $\mathcal{A} = \mathcal{A}'$. Otherwise, let us assume that:

$$\Pr\left[\texttt{Gamej+1}\right]^{\mathcal{A}} \geq \Pr\left[\texttt{Gamej}\right]'^{\mathcal{A}} + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.42}$$

We will prove that this is impossible, otherwise we could use $\mathcal{A}$ to attack the game $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$ with probability $\frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$ (which is assumed to be impossible).

To that end, we proceed by reduction and create an adversary $\mathcal{A}_2$ from $\mathcal{A}$ as described in Figure 4.3 (we can imagine that $\mathcal{A}_2$ is allowed to use the code of $\mathcal{A}$ internally, represented with communications between $\mathcal{A}_2$ and $\mathcal{A}$).

Reduction

	Challenger	**Adversary $\mathcal{A}_2$**		$\mathcal{A}'$
1 :		$c \xleftarrow{\$} \{0,1\}, \forall i, \phi_i \coloneqq \phi_i^{(c)}$	$\xleftarrow{\phi^{(1)}, \phi^{(2)}}$	
2 :		Run j $\mathbb{Z}_{\frac{\pi}{4}}$ – $\mathsf{RSP}_{\mathsf{CC}}$ to get $\{\theta_i\}_{i\in[j]}$	$\xleftrightarrow{\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}}$	
3 :	Run $\mathbb{Z}_{\frac{\pi}{4}}$ – $\mathsf{RSP}_{\mathsf{CC}}$ gives θ	$\xrightarrow{\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}}$		
4 :		Run $n-j$ $\mathbb{Z}_{\frac{\pi}{4}}$ – $\mathsf{RSP}_{\mathsf{CC}}\rightarrow$ $\{\theta_i\}_{i\in[n]\setminus[j+1]}$	$\xleftrightarrow{\mathbb{Z}_{\frac{\pi}{4}} - \mathsf{RSP}_{\mathsf{CC}}}$	
5 :		$r \xleftarrow{\$} \{0,1\}^n$		
6 :		$\delta_1 = (-1)^{f_1(\dots)}\phi_1 + r_1\pi + \theta_1 + f_2(\dots)\pi$	$\xrightarrow{\delta_1}$	
7 :		$\vdots$	$\xleftarrow{s_1}$	
8 :		$\delta_j = (-1)^{f_1(\dots)}\phi_j + r_j\pi + \theta_j + f_2(\dots)\pi$	$\xrightarrow{\delta_j}$	
9 :			$\xleftarrow{s_j}$	
10 :		$\tilde{\theta}_\pi \xleftarrow{\$} \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}, r_{j+1} \xleftarrow{\$} \{0,1\}$		
11 :		$\delta_{j+1} \coloneqq (-1)^{f_1(\dots)}\phi_i^{(c)} + \tilde{\theta}_\pi + r_{j+1}\pi$	$\xrightarrow{\delta_{j+1}}$	
12 :		**if** $\tilde{c} = c$ **then** $\tilde{\theta}_g \coloneqq \tilde{\theta}_\pi$	$\xleftarrow{s_{j+1}, \tilde{c}}$	
13 :	$\xleftarrow{\tilde{\theta}_g \stackrel{?}{=} \theta \bmod \pi}$	**else** $\tilde{\theta}_g \xleftarrow{\$} \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$		

Figure 4.3: Reduction.

We compute now $\Pr\left[\tilde{\theta}_g = \theta \bmod \pi\right]$, which is the probability of winning the game $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$ (all probabilities will be expressed in term of a run of the game $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$ with the adversary $\mathcal{A}_2$). In the following, we use the notation $\theta_\pi \coloneqq \theta \bmod \pi$. Now, we will derive a few properties on the distribution of the variable involved in this game in order to derive a probability tree (picture in Figure 4.4) describing the probability of winning $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$.

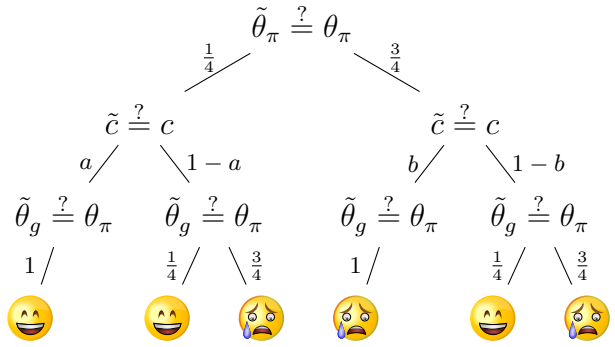


Figure 4.4: Probability tree denoting when the adversary $\mathcal{A}_2$ wins the game, i.e. when $\tilde{\theta}_g = \theta_\pi$. The nodes represent the event (conditioned on the above events in the tree), the left branch represents the probability for this event to be true (the probability is written next to the branch), and the right branch represents the probability for this event to be false. At the leaves of the tree lies an happy face when $\mathcal{A}_2$ wins the game and a sad face otherwise.

First, we can remark that since $\tilde{\theta}_\pi$ is sampled independent of θ, we have:

$$\Pr\left[\tilde{\theta}_\pi = \theta_\pi\right] = \frac{1}{4} \tag{4.43}$$

Then, when $\tilde{\theta}_\pi = \theta_\pi$ the setting is equivalent to the `Gamej+1` game.

> The reason is that the last angle δ_{j+1} sent in `Gamej+1` has the form $\delta_{j+1} = \alpha + r_{j+1}\pi$. Because r_{j+1} is sampled uniformly at random and used only once, it completely one-time pads the value bit of α: said differently, the distribution $\alpha + r_{j+1}\pi$ is exactly the same as the distribution $\alpha_\pi + r_{j+1}\pi$, with $\alpha_\pi \coloneqq \alpha \bmod \pi$. Moreover, when $\tilde{\theta}_\pi = \theta_\pi$, $\delta_{j+1} \bmod \pi$ has exactly the value we would have obtained in the game `Gamej+1`.

Therefore, according to Eq. (4.42), we have:

$$a \coloneqq \Pr\left[\tilde{c} = c \mid \tilde{\theta}\pi = \theta_\pi\right] = \Pr\left[\texttt{Gamej+1}^{\mathcal{A}}\right] \geq \Pr\left[\texttt{Gamej}'^{\mathcal{A}}\right] + \frac{1}{\mathsf{poly}(\lambda)} \tag{4.44}$$

Then, we also remark that $\tilde{\theta}_\pi$ is sampled uniformly at random and so is r_{j+1}. Since δ_{j+1} is the last message sent to $\mathcal{A}$ and is defined as a sum of $\delta_{j+1} = \alpha + \tilde{\theta}_\pi + r_{j+1}\pi$ for some α, its distribution is completely uniform for $\mathcal{A}$. Therefore, we are exactly in the setting of $\texttt{Gamej}'$ and therefore:

$$d \coloneqq \Pr\left[\tilde{c} = c\right] = \Pr\left[\texttt{Gamej}'^{\mathcal{A}}\right] \tag{4.45}$$

This is helpful to determine $b \coloneqq \Pr\left[\tilde{c} = c \mid \tilde{\theta}_\pi \neq \theta_\pi\right]$ since

$$d = \Pr\left[\tilde{c} = c\right] \tag{4.46}$$
$$= \Pr\left[\tilde{\theta}_\pi = \theta_\pi\right] \Pr\left[\tilde{c} = c \mid \tilde{\theta}_\pi = \theta_\pi\right] + \Pr\left[\tilde{\theta}_\pi \neq \theta_\pi\right] \Pr\left[\tilde{c} = c \mid \tilde{\theta}_\pi \neq \theta_\pi\right] \tag{4.47}$$
$$\overset{(4.43)}{=} \frac{1}{4}a + \frac{3}{4}b \tag{4.48}$$

we have

$$b = \frac{4}{3}(d - \frac{1}{4}a) = \frac{4}{3}d - \frac{1}{3}a \tag{4.49}$$

Finally, we can remark that due to the way $\mathcal{A}_2$ samples $\tilde{\theta}_g$, we have:

$$\Pr\left[\tilde{\theta}_g = \tilde{\theta}_\pi \mid \tilde{c} = c\right] = 1 \tag{4.50}$$
$$\Pr\left[\tilde{\theta}_g = \tilde{\theta}_\pi \mid \tilde{c} \neq c\right] = \frac{1}{4} \tag{4.51}$$

Combining all these probabilities together (following the tree drawn in Figure 4.4), we obtain that the probability of winning the game $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$ is:

$$\Pr\left[\tilde{\theta}_g = \theta_\pi\right] = \frac{1}{4}a + \frac{1}{4}(1-a)\frac{1}{4} + \frac{3}{4}(1-b)\frac{1}{4} \tag{4.52}$$
$$= \frac{1}{4}a + \frac{1}{16} - \frac{1}{16}a + \frac{3}{16} - \frac{3}{16}b \tag{4.53}$$
$$= \frac{1}{4} + \frac{3}{16}(a-b) \tag{4.54}$$

But $b \overset{(4.49)}{=} \frac{4}{3}d - \frac{1}{3}a$ so

$$a - b = a + \frac{1}{3}a - \frac{4}{3}d = \frac{4}{3}(a-d) \overset{(4.44)}{=} \frac{4}{3}\left(\Pr\left[\texttt{Gamej+1}^{\mathcal{A}}\right] - \Pr\left[\texttt{Gamej}'^{\mathcal{A}}\right]\right) \tag{4.55}$$
$$\overset{(4.42)}{\geq} \frac{1}{\mathsf{poly}(\lambda)} \tag{4.56}$$

Therefore combining this with Eq. (4.54) gives a probability of winning the game $\texttt{IND-}\mathbb{Z}_{\frac{\pi}{4}}\texttt{-QFactory}$ greater than $\frac{1}{4} + \frac{1}{\mathsf{poly}(\lambda)}$, which is supposed to be impossible.

It is now easy to conclude by induction since we have:

$$\sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{IND-UBQC}_{\texttt{cc}}{}^{\mathcal{A}}\right] = \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamen}^{\mathcal{A}}\right] \leq \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamen-1}'^{\mathcal{A}}\right] + \mathsf{negl}(\lambda) \tag{4.57}$$
$$= \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Gamen-1}^{\mathcal{A}}\right] + \mathsf{negl}(\lambda) \leq \dots \tag{4.58}$$
$$\leq \sup_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Game0}^{\mathcal{A}}\right] + \mathsf{negl}(\lambda) \tag{4.59}$$

However, in the game $\texttt{Game0}$ no message linked with the secret are sent to the adversary, therefore no adversary can win this game with probability better than $\frac{1}{2}$, i.e. $\sup_{\mathsf{QPT}_{\mathcal{A}}} \Pr\left[\texttt{Game0}^{\mathcal{A}}\right] = \frac{1}{2}$. Because there is only a polynomial number of steps, we have therefore

$$\sup_{\mathsf{QPT}_{\mathcal{A}}} \Pr\left[\texttt{IND-UBQC}_{\texttt{cc}}{}^{\mathcal{A}}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{4.60}$$

which concludes the proof. □

4.6 Non-Negligible δ: Treating the Abort Case

4.6.1 Why Abort is Important

In this section, we will discuss an extension of BB84-QFactory (used internally in $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory), whose aim is to achieve basis blindness when the δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family has a non-negligible δ (our proof also works when δ is a constant). This occurs notably when we rely on the hardness assumption of LWE with polynomial noise ratio (see Remark 5.3.8 why this assumption may make sense) as we will see in Theorem 5.3.9.

The problem of the current BB84-QFactory protocol is that when y has not exactly 2 preimages—which occurs with probability $1 - \delta$ when the server is honest—the protocol will abort. There are then multiple possible strategies to deal with this abort.

A first method is to just behave as if the protocol had not aborted by randomly choosing the output of the protocol as done in the $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory protocol. In that case, of course, the protocol is not correct anymore. When δ is negligible, this is not really an issue: this occurs with negligible probability when the server is honest. But when δ is not negligible, it means that the protocol is not correct anymore... which is obviously an issue.

An alternative approach is to reveal to the server that the protocol aborted, and to restart the protocol from scratch. Unfortunately, it is then hard to prove the security in that later case as this *abort bit* (i.e. whether we aborted or not) can potentially leak a lot of information about the basis. For instance, let us imagine that the server has a way to maliciously sample y in such a way that y has 2 preimages if and only if the basis is the computational basis (a-priori, this does not contradict any of the assumptions on f_k). Then the server will always abort when the state is in the Hadamard state, effectively producing only qubits in the computational basis... The protocol is completely insecure!

Of course, it may be impossible to sample y in such a malicious way (and with our current construction we found no obvious flow occurring when leaking the abort bit). But so far we have no proof of security.

4.6.2 A Quick Overview of Our Approach

In this section we propose a third alternative. While the protocol is given in Protocol 6, we provide now a quick overview of the method and of the proof of security.

The rough idea of our method is to run t BB84-QFactory protocols between Alice and Bob and exploit the fact that when the protocol aborts because there is a single preimage[5], the state obtained by Bob is not completely useless: it is a state in the computational basis (we have no superposition when there is a single pre-image, therefore we are left with either $|0\rangle$ or $|1\rangle$ as proven in Lemma 4.6.2. Note that we will denote by 0 the computational basis and by 1 the Hadamard basis). Having that remark in mind, we combine on Bob's side all these runs (including the runs that aborted) using a gadget circuit in order to obtain a new qubit. This qubit will be such that its basis is the XOR of the basis of all the input qubits as proven in Lemma 4.6.3 (which also corresponds to the XOR of the basis of the accepted runs since during an abort the basis is 0). Then, Alice will divide the runs in c chunks of size t_c (this is required for the proof of security) and check that the number of accepted runs in each chunk is high enough (the fraction must notably be greater than $1/2$ to avoid the aforementioned attack: this will happen with overwhelming probability if Bob is honest and $\delta > 1/2$). If this is the case, Alice will just output the description of the final qubit (whose basis is the XOR of the basis of the accepted run), and otherwise (i.e. if the server is malicious), she will just outputs a random bit value.

Intuitively, the security holds because for a given chunk $i \in [c]$, Bob cannot fully learn b_i, the XOR of the basis of all the runs in that chunk (this is formalized and proven in Lemma 4.6.7). Moreover, we also want to say that if Bob does not really know b_0, nor b_1, nor $b_2 \ldots$ then he has negligible information on $\oplus_{i\in[c]} b_i$. This property is known as privacy amplification, and the Yao's XOR Lemma is a theorem that can be used to prove this kind of statements. Unfortunately, the original theorem does not apply to our setting: this lemma as been proven in the classical case and there were no interaction (see [GNW98] for a review of this theorem as well as the main proof methods). Some works [VW07] also extended this lemma to protocols, and also to the quantum setting

[5] In our fuuntion construction we will have at most 2 preimages, so this occurs with probability δ during an honest run.

[She12, KŠdW07], but unfortunately these last works focus mostly on communication and query complexity, and are not really usable in our case. As a consequence, we need to conjecture that this theorem also applies to quantum interactive protocols.

In the following, we will call "accepted run" a run of BB84-QFactory such that the received y from the server has 2 preimages ("probability of success" also refers to the probability of this event when the server is honest), and otherwise we call it an "aborted run" (we assume that f_k can have at most 2 preimages).

We will also discuss other methods that could help us to avoid this assumption and to improve the efficiency of the protocol in Section 4.7.

4.6.3 Correctness and security of non-negl-BB84-QFactory

Now, we will formalize and prove the previous statements. First, we state the conjecture on which we will build our further results, itself based on the Yao's XOR Lemma described notably in the review of Goldreich [GNW98]. This review presents notably the proof of Levin [Lev87] and Impagliazzo [Imp95b] (see also [She12, KŠdW07, VW07] for extensions).

The original lemma is roughly stated as follows (the exact formulation is more general): if $x \in \mathcal{X}$ is sampled according to some distributions χ, if $P : \mathcal{X} \to \{0,1\}$ is a potentially not efficiently computable randomized predicate, and if no classical efficient randomized algorithm can guess $P(x)$ given x with a probability greater than $1-\delta$ where $\delta \in (0, \frac{1}{2}]$ is a constant[6], then no classical efficient randomized algorithm can guess $\oplus_{i\in[t]} P(x_i)$ given $(x_1, \ldots, x_t) \xleftarrow{\$} \chi^t$ with non-negligible (in λ and $t(\lambda)$) advantage over a random guess.

Typically, x is the result of the evaluation of a one-way function f on a random input, and $P(x)$ first (inefficiently) inverts f before computing a predicate on these preimages. Unfortunately, in our case P depends not only on the preimages, but also on the output of the adversary (the abort bit). Therefore, we need an extension of this lemma where the predicate depends on the output of the adversary. Note that we state here a version in which the rounds are processed in parallel for simplicity, but we could also adapt our proof for a version in which the rounds are processed sequentially.

Conjecture 4.6.1 (Yao's XOR Lemma for one-round protocols (classical messages) against quantum adversary)**.**
Let λ be the security parameter, let $P_\lambda : \mathcal{K}_\lambda \times \mathcal{Y}_\lambda \to \{0,1\}$ be a (possibly non-deterministic) family of functions (usually not computable in polynomial time), and let χ_λ be a distribu-

[6] δ can also be chosen as $\delta \geq \frac{1}{\mathsf{poly}(\lambda)}$ but t needs to scale appropriately.

tion on $\mathcal{K}_\lambda$ efficiently samplable. If there exists $\delta(\lambda)$—intuitively the probability of failing to guess one round—such that $|\delta(\lambda)| \geq \frac{1}{\mathsf{poly}(\lambda)}$ and such that for any polynomial (in λ) quantum adversary $\mathcal{A}_\lambda : \mathcal{K}_\lambda \to \mathcal{Y}_\lambda \times \{0,1\}$,

$$\Pr\left[\tilde{\beta} = P_\lambda(k,y) \mid (y,\tilde{\beta}) \leftarrow \mathcal{A}_\lambda(k), k \leftarrow \chi_\lambda\right] \leq 1 - \delta(\lambda)$$

then, for all $t \in \mathbb{N}_{>0}$, there is no polynomial quantum adversaries $\mathcal{A}'_\lambda : \mathcal{K}_\lambda^t \to \mathcal{Y}_\lambda^t \times \{0,1\}$ such that:

$$\begin{aligned}&\Pr\left[\tilde{\beta} = \bigoplus_{i=1}^{t} P_\lambda(k_i,y_i) \mid (y_1,\ldots,y_t,\tilde{\beta}) \leftarrow \mathcal{A}'_\lambda(k_1,\ldots,k_t), \forall i, k_i \leftarrow \chi_\lambda\right] \\ &\geq \frac{1}{2} + (1-\delta(\lambda))^t + \mathsf{negl}(\lambda)\end{aligned}$$

Lemma 4.6.2 (Aborted runs are useful)**.** *If f_k has at most two preimages, if Alice and Bob are following the BB84-QFactory protocol honestly, and if y has not 2 preimages, then the output qubit produced by Bob is in the basis $\{|0\rangle, |1\rangle\}$.*

Proof. The function f_k cannot have more than two preimages by assumption, and in the BB84-QFactory protocol the output y is in the image of f_k. So y has exactly one preimage x. Therefore, after measuring the last register, the states will be $|x\rangle \otimes |h(x)\rangle \otimes |y\rangle$. The qubit in the second register ($|h(x)\rangle$) is in the computational basis and is not entangled with the first register: after measuring the first register, the second register stays untouched. □

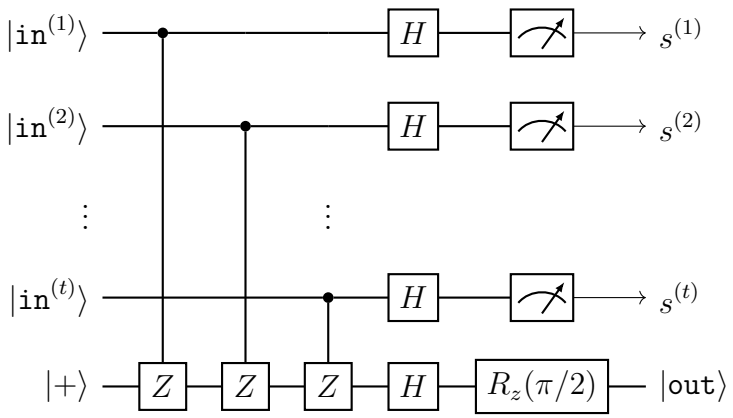
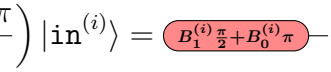


Figure 4.5: The XOR gadget circuit $\mathtt{Gad}_\oplus$ (run on server side). Note that the rightmost Hadamard and rotation on the last wire is only used to bring a $|+_\theta\rangle$ back into a BB84 state. When using other protocols that actually expect $|+_\theta\rangle$ states (the first step of the $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory protocol undoes this operation) we can remove them. We see that this circuit has been simplified since the our original publication (thanks ZX-calculus).

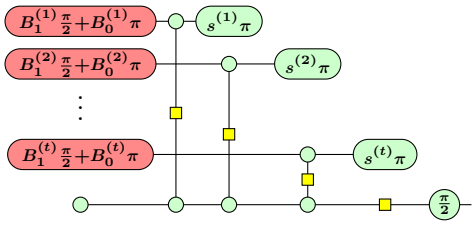
Lemma 4.6.3 (Gadget circuit $\texttt{Gad}_{\oplus}$ computes XOR)**.** *If we denote by* $B_1^{(i)}$ *the basis of* $|\boldsymbol{in}^{(i)}\rangle = \mathbf{H}^{B_1^{(i)}} |B_0^{(i)}\rangle$ *(equal to* 0 *if the basis is computational and* 1 *if the basis is Hadamard), and if we run the circuit* $\texttt{Gad}_{\oplus}$ *represented Figure 4.5 on these inputs, then the basis of* $|\boldsymbol{out}\rangle$ *is equal to* $\oplus_{i=1}^{t} B_1^{(i)}$.

Proof. First, as shown in Eq. (4.14), applying $\mathbf{R}_z(\frac{-\pi}{2})$ on $|\texttt{in}^{(i)}\rangle = \mathbf{H}^{B_1^{(i)}} |B_0^{(i)}\rangle$ gives:

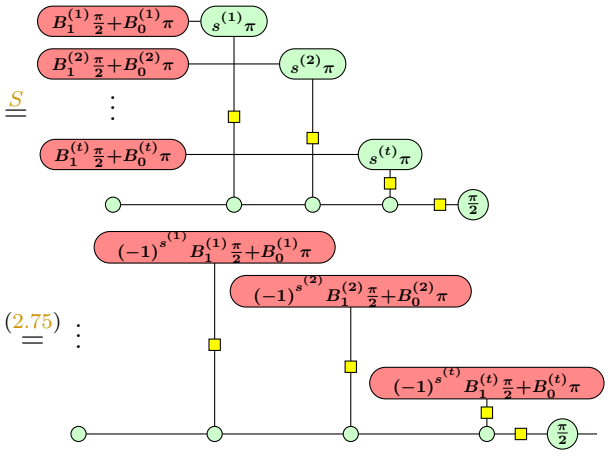
$$\mathbf{R}_z\left(\frac{-\pi}{2}\right)|\texttt{in}^{(i)}\rangle = \tag{4.61}$$



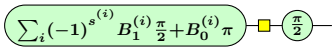
Therefore, the above circuits can be rewritten as:

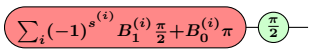


$\overset{S}{=}$ (4.62)



$\overset{(2.75)}{=}$ (4.63)

$\overset{H,S}{=}$  (4.64)

$\overset{S}{=}$  (4.65)

By using now Eq. (4.61) in the reverse order, we can conclude that we obtain a BB84 state whose basis is:

$$\sum_i (-1)^{s^{(i)}} B_1^{(i)} \frac{\pi}{2} + B_0^{(i)}\pi \bmod \pi = \oplus_i B_1^{(i)} \tag{4.66}$$

which concludes the proof. □

We describe now in Protocol 6 the protocol non-negl-BB84-QFactory.

Lemma 4.6.4 (Probability of correctness of non-negl-BB84-QFactory for one chunk)**.** *If the probability to have an accepted run in BB84-QFactory with honest parties is greater than a constant* $p_a > 1/2$, *i.e.*

$$\Pr[|f_k^{-1}(y)| = 2 \mid \mathsf{Alice}_{BB} \leftrightsquigarrow \mathsf{Bob}_{BB}] \geq p_a$$

Protocol 6 non-negl-BB84-QFactory

Assumptions: There exists a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1) for $n = 1$ with $\delta < 1/2$, such that f_k has at most 2 preimages for any y and such that the trapdoor t_k allows the complete inversion of f_k for any y. For the security, we also assume the Conjecture 4.6.1.
Parameters: We use some constants in the protocol: n_c is the number of chunks, $t_c \in \mathbb{N}$ is the number of repetitions per chunk, $t = n_c \times t_c$ is the total number of repetitions, $p_a \in (1/2, 1] > 1 - \delta$ is a lower bound on the probability of accepting, and $p_c \in (1/2, 1] < p_a$ is the fractions of the runs per chunk that must be accepted. These constants can be chosen to have overwhelming probability of success for honest players, and negligible advantage for a malicious adversaries trying to guess the basis (assuming our conjecture).
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's outputs: The description B_0, B_1 of a BB84 state whose basis is B_1.
Bob's output: A BB84 state $\mathbf{H}^{B_1} |B_0\rangle$.
Protocol:

1. Alice runs $n_c \times t_c$ times the BB84-QFactory protocol—except that Alice does not abort if there is less than 2 preimages—in order to obtain the description $\{(B_0^{(i,j)}, B_1^{(i,j)})\}$ of the produced BB84 states (in (i, j), i corresponds to the chunk number and j to the index in the chunk i). She also defines $a^{(i,j)} = 0$ if the protocol aborted and $a^{(i,j)} = 1$ otherwise. If there is a single preimage she computes $B_1^{(i,j)} := 0$ and $B_0^{(i,j)} = h(f^{-1}(y^{(i,j)}))$. If there is no preimage (it clearly means that the server is cheating), she just outputs a random value for both B_0 and B_1.
2. Bob runs the circuit Figure 4.5 on the t outputs of the previous run and outputs $|\mathtt{out}\rangle$.
3. Alice checks that for all chunks $i \in [n_c]$ the number of accepted runs is high enough, i.e. that $\sum_j a^{(i,j)} \geq p_c t_c$.
   - If at least one chunk does not respect this condition, Alice picks two random bits B_1 (the basis bit) and B_0 (the value bit) and outputs (B_1, B_0), corresponding to the description of the BB84 state $\mathbf{H}^{B_1} |B_0\rangle$.
   - If all chunks respect this condition, then she sets $B_1 := \bigoplus_{i,j} B_1^{(i,j)}$ (the final basis is the XOR of all the basis), and B_0 will be chosen to match the output of Figure 4.5.

(where Alice_{BB} and Bob_{BB} are the honest parties in the BB84-QFactory protocol) then the probability to have at least $p_b t_c$ accepted runs (with $p_b < p_a$, p_b considered as a constant) is exponentially (in t_c) close to 1:

$$\Pr\left[\sum_i a_i \geq p_b t_c \mid (\mathsf{Alice}_{1\oplus}^{t_c} \| \mathsf{Bob}_{1\oplus}^{t_c})\right] \geq 1 - \frac{1}{e^{2(p_a - p_b)^2 t_c}} = 1 - \mathsf{negl}(t_c)$$

(where $\mathsf{Alice}^{t_c}_{1\oplus}$ *and* $\mathsf{Bob}^{t_c}_{1\oplus}$ *are the (honest) parties of the Protocol 6 restricted on one chunk of size* t_c*)*

Proof. In the honest case, all runs are independents, so let us define $\{A_i\}_{i=1}^{t}$ as the (binary) random variables whose values are 1 iff the i-th run has two preimages associated with y_i. We know that for all i, $\mathbb{E}(A_i) \geq p_a > p_b$. So let us define $\varepsilon = \mathbb{E}(A_i) - p_b > p_a - p_b$. Using Chernoff inequality we have

$$\Pr\left[\frac{1}{t}\sum_{i=1}^{t} A_i < \mathbb{E}(A_i) - \varepsilon\right] \leq e^{-2\varepsilon^2 t} \leq e^{-2(p_a-p_b)^2 t} = \mathsf{negl}(t)$$

(because $p_a - p_b$ is constant) □

Lemma 4.6.5 (Correctness of Protocol 6)**.** *The Protocol 6 is correct with overwhelming probability as soon as* $t = \mathsf{poly}(\lambda)$ *and* $t_c = \Omega(\lambda)$*, i.e.*

$$\Pr\left[|\mathtt{out}\rangle = H^{B_1}Z^{B_2} \mid ((B_1, B_2), |\mathtt{out}\rangle) \leftarrow (\pi_A \| \pi_B)\right] \geq 1 - \mathsf{negl}(\lambda)$$

Proof. The Lemma 4.6.4 gives that the probability to have more than $p_c t_c$ accepted runs for a given chunk is $1 - \mathsf{negl}(t_c)$, i.e. if $t_c = \Omega(\lambda)$, this probability is $\mathsf{negl}(\lambda)$. So for n_c chunks, the probability to have one fail is $(1 - \mathsf{negl}(\lambda))^{n_c} = 1 - \mathsf{negl}(\lambda)$ as soon as $n_c = \mathsf{poly}(\lambda)$, which is the case because $t = t_c \times n_c = \mathsf{poly}(\lambda)$. Then, when all the chunks are accepted, the correctness of the output values is assured by Lemma 4.6.3. □

Definition 4.6.6. *For any public key* k *and image* y*, we define* $a(k, y) = 1$ *iff* $|f_k^{-1}(y)| = 2$*, and* $a(k, y) = 0$ *otherwise.*
Then, for all $t_c \in \mathbb{N}$ *and* $p_c \in [0, 1]$*, we define* $\beta_{t_c,p_c}(k^{(1)}, \ldots, k^{(t_c)}, y^{(1)}, \ldots, y^{(t_c)})$ *as the (randomized) function that outputs a random bit if* $\sum_i a(k^{(i)}, y^{(i)}) < p_c \cdot t_c$*, and outputs otherwise* $\oplus_i (a(k^{(i)}, y^{(i)}) \cdot d_0^{(i)})$*, where* $d_0^{(i)}$ *is the hardcore bit corresponding to* $k^{(i)} := (K^{(i)}, g_{K^{(i)}}(z_0^{(i)}))$*, i.e.* $d_0^{(i)} = h(z_0^{(i)})$*.*

Lemma 4.6.7 (Solving one chunk is difficult)**.** *Let* $p_c \in (\frac{1}{2}, 1]$*. Then, there exists no polynomial adversary* $\mathcal{A}$ *such that:*

$$\Pr\Big[\tilde{B}_1 = \beta_{t_c,p_c}(k^{(1)}, \ldots, k^{(t_c)}, y^{(1)}, \ldots, y^{(t_c)}) \\ \mid \forall i, d_0^{(i)} \xleftarrow{\$} \{0,1\}, k^{(i)} \leftarrow \mathtt{Gen}(1^\lambda, d_0^{(i)}), (y^{(1)}, \ldots, y^{(t_c)}, \tilde{B}_1) \leftarrow \mathcal{A}(k^{(1)}, \ldots, k^{(t_c)})\Big] > \eta$$

with β_{t_c,p_c} *is the basis computed by Alice for a single chunk in the non-negl-BB84-QFactory protocol—potentially random if too many aborts are present—*$\eta = \frac{1}{2}\left(1 + \frac{1}{2p_c}\right)$*, where the randomness is over the randomness of* β*,* $\mathcal{A}$*, and over the choice of* $(k^{(i)})_i$ *and* $(d_0^{(i)})_i$*.*

Proof. By contradiction, let us assume that there is an adversary $\mathcal{A}$ such that (we omit the parameters for readability)

$$\Pr\left[\tilde{B}_1 = \beta\right] > \eta$$

Then, if we define the abort bit $a_i := a(k^{(i)}, y^{(i)})$,

$$\begin{aligned}
\eta &< \Pr\left[\tilde{B}_1 = \beta\right] \\
&= \underbrace{\Pr\left[\sum_i a_i < p_c t_c\right]}_{\alpha} \times \frac{1}{2} + \Pr\left[\sum_i a_i \geq p_c t_c\right] \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right] \\
&= \alpha \times \frac{1}{2} + (1-\alpha) \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right] \\
&\leq \alpha \times \frac{1}{2} + (1-\alpha) = 1 - \frac{\alpha}{2}
\end{aligned}$$

so $\alpha \leq 2(1-\eta)$.

Now, we remark that we can bound also $(1-a) \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right]$. Indeed, if this value is too big then we can construct an adversary that could break the hardcore bit property of g_K. To do that, we define an adversary $\mathcal{A}'$ taking as input a k, and whose goal is to find the hardcore bit d_0 associated with k. This adversary will pick $t_c - 1$ public keys/trapdoors $(k^{(i)}, t_{k^{(i)}})$, and hide k in the middle of these trapdoors. Then, $\mathcal{A}'$ calls $\mathcal{A}$ with these t_c keys, and outputs $\tilde{d_0} := \tilde{B}_1 \oplus_i a^{(i)} d_0^{(i)}$, with $\tilde{B}_1$ the output of $\mathcal{A}$, and $a^{(i)}$ computed by using the $y^{(i)}$ provided by $\mathcal{A}$. We know that $\tilde{d_0} = d_0$ when the guess of $\mathcal{A}'$ was right, when $\sum_i a_i \geq p_c t_c$, and when the y corresponding to the function k has two preimages. But this event occurs with probability greater than $(1-\alpha) \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right] \times p_c$, and because d_0 is a hardcore bit, this probability is bounded by $1/2 + \mathsf{negl}(\lambda)$, or equivalently:

$$(1-\alpha) \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right] \leq \frac{1}{2p_c} + \mathsf{negl}(\lambda)$$

Now, let's come back to our probability to guess β:

$$\begin{aligned}
\Pr\left[\tilde{B}_1 = \beta\right] &= \alpha \times \frac{1}{2} + (1-\alpha) \times \Pr\left[\tilde{B}_1 = \beta \mid \sum_i a_i \geq p_c t_c\right] \\
&\leq \alpha \times \frac{1}{2} + \frac{1}{2p_c} + \mathsf{negl}(\lambda) \\
&\leq 1 - \eta + \frac{1}{2p_c} + \mathsf{negl}(\lambda)
\end{aligned}$$

But on the other side, $\Pr\left[\tilde{B}_1 = \beta\right] > \eta$, so

$$\eta < 1 - \eta + \frac{1}{2p_c} + \mathsf{negl}(\lambda)$$
$$\eta < \frac{1}{2}\left(1 + \frac{1}{2p_c}\right) + \mathsf{negl}(\lambda)$$

Because η and p_c are constants[7] that do not depend on n, this equality is also true without the $\mathsf{negl}(\lambda)$:

$$\eta < \frac{1}{2}\left(1 + \frac{1}{2p_c}\right)$$

which is absurd because $\eta = \frac{1}{2}\left(1 + \frac{1}{2p_c}\right)$. □

Theorem 4.6.8 (non-negl-BB84-QFactory is correct and secure)**.** *Assuming Conjecture 4.6.1, and by making sure that the probability for the family $\mathcal{F}$ to have two preimages for a random image is bigger than a constant $p_a > 1/2$, then there exists a set of parameters p_c, t_c and n_c such that Protocol 6 is correct with probability exponentially close to 1 and basis-blind, i.e. such that for all polynomial adversaries $\mathcal{A}$:*

$$\Pr\left[\tilde{B}_1 = B_1 \mid ((B_1, B_2), \tilde{B}_1) \leftarrow (\mathsf{Alice}_{\oplus} \leftrightsquigarrow \mathcal{A})\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

More precisely, we need $t_c \in (1/2, p_c)$ to be a constant, and both t_c and n_c need to be polynomial in n and $\Omega(n)$.

Proof. The proof of correctness is made in Lemma 4.6.5, and the security is a direct application of Conjecture 4.6.1: after using Lemma 4.6.7: this theorem provides a η such that it's not possible to solve one chunk with probability better than $\eta < 1$, so $\delta(n) := 1 - \eta$ is a constant (and $\delta(n) \geq \frac{1}{\mathsf{poly}(\lambda)}$). Therefore Conjecture 4.6.1 tells us that no adversary can get the XOR of n_c chunks with probability better than $\frac{1}{2} + \eta^{n_c} + \mathsf{negl}(\lambda)$. But $t_c = \Omega(n)$ and η is a constant, so no adversary can get the XOR of n_c chunks with probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$, i.e. no adversary can find B_1 with probability better then $\frac{1}{2} + \mathsf{negl}(\lambda)$. □

4.7 Unprovable Extensions and Open Questions

In this section, we discuss two potential improvements of the above protocols (one to produce a $|+_\theta\rangle$ state using a single superposition instead of 2, and one to optimize the

[7] note that if we give them a dependence on n, we can make sure that $\eta - \frac{1}{2}\left(1 + \frac{1}{2p_c}\right)$ is non negligible, but for simplicity we will keep them constant

protocol non-negl-BB84-QFactory shown in Section 4.6 relying on LWE with polynomial noise ratio). Unfortunately we are not able to prove their security. We also discuss the questions which are left open.

4.7.1 Producing $|+_\theta\rangle$ using a single superposition

The $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory protocol runs the BB84-QFactory protocol two times to produce a single $|+_\theta\rangle$ state (and we need three executions to also obtain $|0\rangle$ and $|1\rangle$). However, the BB84-QFactory protocol internally creates a very heavy superposition: it is therefore interesting to find a way to reduce the number of superpositions to prepare a $|+_\theta\rangle$. We propose here a protocol to generate $|+_\theta\rangle$, $|0\rangle$ and $|1\rangle$ using a single superposition. Unfortunately we are unable to prove its security in the general case. The idea is basically to generate a GHZ state of size 3 using the GHZ-QFactory protocol, and to measure it appropriately to reduce it to the appropriate one-qubit state.

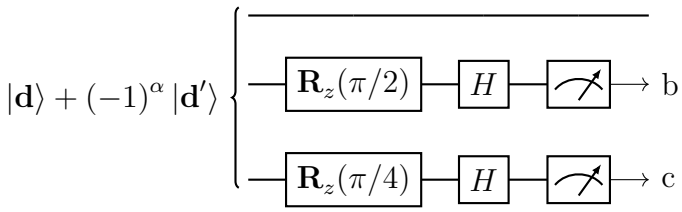


Figure 4.6: Circuit to implement Protocol 7

Theorem 4.7.1 (Correctness of Protocol 7)**.** *The protocol 10 states-QFactory and its particular case $\mathbb{Z}_{\frac{\pi}{4}}$-GHZ-QFactory (described in Protocol 7) are correct, in the sense that if both parties are honest and if Alice outputs ψ, then Bob's output is $|\psi\rangle$. Moreover, the protocol aborts only with probability δ (which is negligible by assumption). Moreover, in the $\mathbb{Z}_{\frac{\pi}{4}}$-GHZ-QFactory protocol, the produced state is a $|+_\theta\rangle$ with $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$ while in the 10 states-QFactory the produced state is in the computational basis if the input B_0 of Alice is 0, otherwise it is a $|+_\theta\rangle$ state with $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$.*

Proof. While we could use standard linear algebra to write this proof, we will use ZX-Calculus which provides not only succinct, intuitive and generalizable proofs, but also explains us why the above construction fundamentally works. However, we first need to see how hidden GHZ states can be represented in ZX-calculus (this representation may also be of independent interest, for instance to see how hidden GHZ states could be useful in other protocols). The first remark that we can make is that in ZX-calculus, an

Protocol 7 10 states-QFactory and its particular case $\mathbb{Z}_{\frac{\pi}{4}}$-GHZ-QFactory

Assumptions: There exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1). Moreover, we only need this construction to work for strings $\mathbf{d}_0$ of size $n = 3$.
Parties: A classical client (Alice) and a quantum server (Bob).
Alice's inputs: A bit $B_0 \in \{0, 1\}$: if $B_0 = 0$, Alice wants to prepare $|0\rangle$ or $|1\rangle$, otherwise she prepares a random $|+_\theta\rangle$ state. In the $\mathbb{Z}_{\frac{\pi}{4}}$-GHZ-QFactory protocol, Alice has no input: B_0 is set to 1.
Alice's outputs: Alice outputs a classical string $\psi \in \{0, 1, +_{\frac{\pi}{4}}, \ldots, +_{\frac{7\pi}{4}}\}$. In the $\mathbb{Z}_{\frac{\pi}{4}}$-GHZ-QFactory protocol, since we can only produce $|+_\theta\rangle$ states, Alice outputs θ directly.
Bob's output: A qubit $|\psi\rangle$.
Protocol:

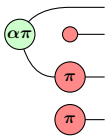
1. Alice computes $\mathbf{d}_0$ by assigning $\mathbf{d}_0[1] \coloneqq B_0$ and randomly sampling $\mathbf{d}_0[2] \xleftarrow{\$} \{0, 1\}$ and $\mathbf{d}_0[2] \xleftarrow{\$} \{0, 1\}$.
2. Alice and Bob run the protocol GHZ-QFactory, where Alice's input is $\mathbf{d}_0$. Alice obtains the description $(\mathbf{d}, \mathbf{d}', \alpha)$ of the (three qubits) hidden GHZ state $|\phi\rangle \coloneqq |\mathbf{d}\rangle + (-1)^\alpha \mathbf{d}'$ obtained by Bob.
3. If the bit string $\mathbf{d}$ is larger (in alphabetic order) than $\mathbf{d}'$, then Alice renames $(\mathbf{d}, \mathbf{d}')$ into $(\mathbf{d}', \mathbf{d})$.
4. Bob measures the second qubit of $|\phi\rangle$ in the basis $\{|+_{-\pi/2}\rangle, |-_{-\pi/2}\rangle\}$—to get the outcome b—and measures the third qubit in the basis $\{|+_{-\pi/2}\rangle, |-_{-\pi/2}\rangle\}$, getting an outcome c. The circuit is pictured in Figure 4.6. Bob sends both b and c to Alice, and outputs the remaining qubit $|\psi\rangle$.
5. If $B_0 = 0$, Alice outputs $\mathbf{d}[1]$ (the state obtained by Bob being $|\mathbf{d}[1]\rangle$). Otherwise Alice outputs $+_\theta$ (the state obtained by Bob being $|+_\theta\rangle$) where:

$$\theta \coloneqq \alpha\pi + \mathbf{d}_0[2](b\pi + (-1)^{\mathbf{d}[2]}\frac{\pi}{2}) + \mathbf{d}_0[3](c\pi + (-1)^{\mathbf{d}[3]}\frac{\pi}{4}) \tag{4.67}$$

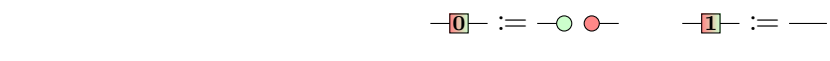
n qubit GHZ state $|0\ldots0\rangle + (-1)^\alpha |\ldots1\rangle$ can be represented using a single green spider with n outputs and an $\alpha\pi$ phase (this is just the definition of the green spider): 

.

A hidden GHZ state also has some $|0\rangle$ at various positions and local $\mathbf{X}$ gates applied to some of the qubits. This can be also applied on the ZX-calculus representation, making the set of entangled qubits more visible. For instance:

$$|0011\rangle + (-1)^\alpha |1001\rangle = \tag{4.68}$$


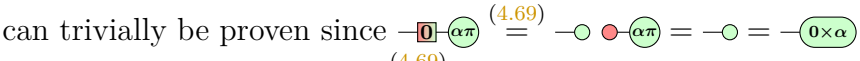
However, the ZX-calculus notation does not yet allow us to represent all hidden GHZ states on a single diagram since the connectivity of the ith qubit depends on the value of $\mathbf{d}_0$. To avoid this issue, we introduce a small notation (which turns out to be a particular case of the *transistors* introduced in [JPV19]) that allows us to "cut" the wires depending on the value of a variable, together with an equation which will turn out to be useful later:

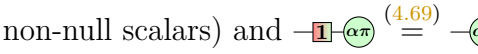


(4.69)

$$\forall d \in \{0,1\}, \alpha \in \mathbb{R},$$

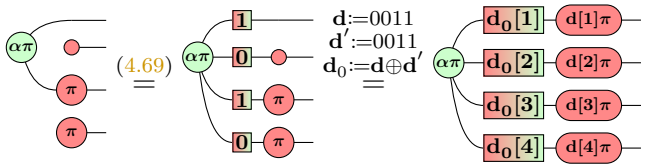
(4.70)

The property Eq. (4.70) can trivially be proven since 

(remember we remove all non-null scalars) and 

. Using this new notation, it is now possible to have a single notation to represent any hidden GHZ states $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle$ in term of its support $\mathbf{d}_0 = \mathbf{d} \oplus \mathbf{d}'$. For instance we can rewrite Eq. (4.68) as follows, and this notation trivially extend to hidden GHZ states of arbitrary size:

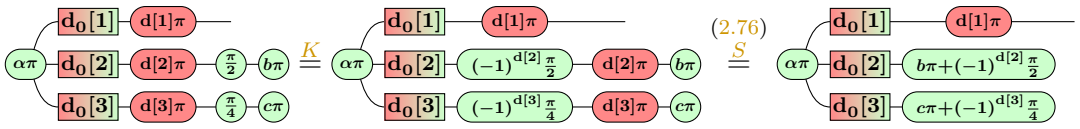
$$|0011\rangle + (-1)^\alpha |1001\rangle \overset{(4.68)}{=}$$



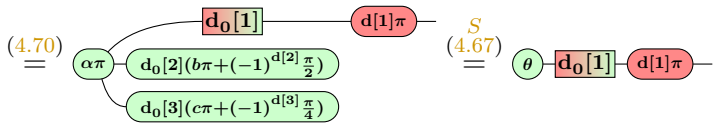
(4.71)

Note that again, we can use both $\mathbf{d}$ or $\mathbf{d}'$ in this equation since $|\mathbf{d}\rangle + (-1)^\alpha |\mathbf{d}'\rangle = |\mathbf{d}'\rangle + (-1)^\alpha |\mathbf{d}\rangle$. In particular we can rename $\mathbf{d}$ and $\mathbf{d}'$ to ensure $\mathbf{d}$ is smaller than $\mathbf{d}'$ in the alphabetic order (this is useful to slightly simplify the expression of θ). Using this notation, we can derive the correctness proof:

$$|\psi\rangle :=$$

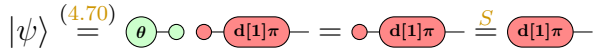


(4.72)

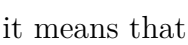


(4.73)

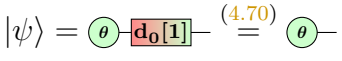
We have now two cases: if $B_0 = 0$, then $\mathbf{d}_0[1] = 0$ and



(4.74)

which means that the output is a qubit in the computational basis $|\mathbf{d}[1]\rangle$. Otherwise if $\mathbf{d}_0[1] = B_0 = 1$, then $\mathbf{d}[1] \neq \mathbf{d}'[1]$: because we chose to define $\mathbf{d}$ as the smallest (in the alphabetic order) string, it means that $\mathbf{d}[1] = 0$, so 

. Therefore:



(4.75)

which is a $|+_\theta\rangle$ state, ending the proof.

As a final remark, if we do not assume that $\mathbf{d} < \mathbf{d}'$, then we get the state $|+_{(-1)^{\mathbf{d}[1]}\theta}\rangle$ (this can be seen using Eq. (2.76)). □

We are able to prove, similarly to Theorem 4.4.6 that no adversary can learn $\theta \bmod \frac{\pi}{4}$ (this is quite direct since $\theta \bmod \frac{\pi}{4} = \mathbf{d}_0[3]$ and $\mathbf{d}_0[3]$ is supposed to be hard to find). Unfortunately, it is harder to prove that no information leaks about $\theta \bmod \pi$: to reduce it to finding an information about $\overrightarrow{d}_0$, we need to get information about one preimage $\mathbf{d}$. Unfortunately, it is not possible to obtain $\mathbf{d}$ without the trapdoor and without destroying the state. [GV19] seems to have a similar issue, and uses a cut-and-choose approach to solve it (some runs are tested, some runs are kept): unfortunately it fundamentally provides polynomial security and we aim to keep superpolynomial security. Another option would be to design f_k such that it has multiple independent trapdoors, so that we can learn partial information about $\mathbf{d}$ (it's basically what we do in Section 7.4). Unfortunately, as far as we know, adding n independent trapdoors multiplies by n the complexity of the function $f_k \ldots$ so this approach is not more efficient than what is already done in $\mathbb{Z}\frac{\pi}{4}$-QFactory.

4.7.2 Improving Protocol for Non-Negligible Delta

Similarly, one may want to optimize the protocol non-negl-BB84-QFactory (to deal with non-negligible δ) and/or get rid of our conjecture.

One option would be to simply say to the server when the BB84-QFactory protocol aborts, so that we can restart the protocol from scratch in that case. Unfortunately we have no proof of security for this approach: this abort bit could potentially leak additional information about $\mathbf{d}_0$ (even if we found no such attack). Actually, we have a construction in which we can make sure that the protocol can only abort before the value of $\mathbf{d}_0$ is fixed (this works only for single bit $\mathbf{d}_0$). The idea is to put no noise in the entry corresponding to $\mathbf{d}_0$ (this should still be secure [BLP+13, Sec. 4.1]). Then, the server would do the superposition only for the first M rows: if the protocol does not abort at that step, adding the last row would not make it abort either since there is no noise. Unfortunately, this is not enough to conclude the proof of security: the server may manage to learn enough information about $(\mathbf{s}_0, \mathbf{e}_0)$ via the abort bit—*a-priori* the server could learn a logarithmic amount of bits about them—so that he can learn $\mathbf{d}_0$. Even if this seems improbable, we still need to find a proof.

A second option would be to keep the protocol as it is, but to change the definition of h with a random oracle. Note that this approach does not have only advantages: we

lose the nice property that for one run of BB84-QFactory the basis bit is fixed before the start of the protocol (we won't have anymore $h(x) \oplus h(x') = \mathbf{d}_0$), but anyway this property was not true anymore in the non-negl-BB84-QFactory protocol. Also, it means that we need to rely not only on the hardness of LWE but also on the quantum random oracle model (but we do not need our conjecture anymore), and the circuit to implement the circuit may be slightly harder since we need to apply a more complicated function h. On the other side, the proof might be simpler to write. The major difficulty is to properly handle the quantum oracle (the oracle can be queried in superposition), but methods like [Zha19] may make the proof possible. However, I've not yet tried to solve the problem that way.

4.7.3 Other Open Questions

In this chapter we saw how to do classical-client remote state preparation, allowing us to obtain classical-client blind quantum computing. However, the question is still open of whether we can do both blind and *verifiable* blind quantum computing (meaning that the server cannot alter the output of the computation) with superpolynomial security ([GV19] provides polynomial security as discussed in Section 4.5). This may be done for instance by inserting our QFactory protocol into the VBQC protocol [FK17]. However, because our protocol is not verifiable, additional care must be taken. Notably, it may be required to add a testing round in our protocol, or to change the way traps are tested in VBQC (maybe by testing the distribution when traps are measured with random angles) not to be vulnerable to a "3-theta" attack.

Similarly, it could be tempting to use our QFactory protocol into other protocols, for instance to do Multiparty Quantum Computing [KKM+21]. Unfortunately, the security is not guaranteed directly for the same reasons. I am also working on replacing quantum communications in position-based verification (where the goal is to prove that we are at a given position in space) for which an impossibility results exists when all parties are classical [CGM+09]. This is discussed in Chapter 8.

4.8 Comparison With Related Works

In this section, we compare our approach with the related works. We summarize in Table 4.1 the strengths and weaknesses of each approach.

[8] For both verifiability and blindness.

	Related Works				Our Work	
	[Mah18a]	[Mah18b]	[Bra18]	[GV19]	Chapter 4	Section 4.3
Blind	Yes	No	Yes	Yes	Yes	Yes
Verifiable	No	Yes	No	Yes	No	No
Modular RSP	No	No	No	Yes	Yes	Yes
Security	SupPoly	SupPoly	SupPoly	Poly[8]	SupPoly	SupPoly
Composable	No	No	No	Yes	No	No
Costly testing	No	Yes	No	Yes	No	No
Assumptions	LWE SupPoly	LWE SupPoly	LWE Poly	LWE SupPoly	LWE SupPoly	LWE Poly + conjecture
Multi-qubits	No	No	No	No	Yes	No

Table 4.1: Comparison with the related works. By "Poly" me mean "polynomial", "SupPoly" means "superpolynomial". More specifically, in the "Security" line, we mean that the security scales polynomially or superpolynomially, and in the line "Assumptions", "LWE poly/SupPoly" means that the protocol is secure and correct assuming the hardness of the LWE problem with polynomial/superpolynomial noise ratio.

The groundbreaking work of Mahadev [Mah18a] was the first result achieving classical-client blind quantum computing and is complementary to our own independent approach. The protocol of Mahadev has the advantage of being non-interactive (a single round of message is required) but provides a monolithic protocol targeting only classical-client blind quantum computing (note also that this protocol hides the input but not the computation: in order to also hide the computation one should encode the circuit in the input and replace the actual circuit with a universal quantum machine). On the other hand, our QFactory protocol is modular and provides a fundamental and atomic functionality: faking quantum channels with classical communications. This allows our protocol to be potentially reused in many other protocols, including, maybe, in quantum multi-party computating or verifiable blind quantum computing. Note however that a separate proof a security must be written for each new protocol involving QFactory since we prove (see Chapter 6) that it is impossible to obtain general composable security for any classical-client RSP protocol. Moreover, for verifiable protocols, our protocol may need additional testing as discussed in Section 4.7.3.

Note also that the work of Mahadev is based on a cryptographic construction assuming the hardness of the *Learning With Error* (LWE) problem with a super-polynomial noise ratio, which is a less standard assumption compared to LWE with polynomial noise ratio.

Later, Brakerski [Bra18] improved the construction of Mahadev in order to allow the use of LWE with polynomial noise ratio. One protocol presented in this chapter also requires the assumption present in [Mah18a]—we will see in Chapter 5 that we need this in order to obtain a negligible δ—but we also provide a second construction that can be used to generalize our QFactory protocol assuming only the hardness of LWE with polynomial noise ratio[9] (corresponds to a non-negligible δ).

Remark 4.8.1. Concerning the modularity, one could make the remark that in [Mah18a] the server also ends up with a quantum state $|\phi\rangle = \mathbf{X^a Z^b} |\psi\rangle$, and therefore this protocol may be seen as an RSP protocol. We do not claim that we cannot turn [Mah18a] into an RSP protocol (similarly, we may also potentially be able to adapt our protocol to obtain Quantum Fully Homomorphic Encryption), however, the security proof regarding how $|\phi\rangle$ is hidden to the server is not direct and some additional work must be done. The reason is that [Mah18a] shows that the input of the circuit stays hidden to Bob, but Bob may be able to maliciously play with the value of $\mathbf{a}$ and $\mathbf{b}$ to learn additional information about the one-time padded state $|\phi\rangle$.

Mahadev also provides in another seminal paper [Mah18b] a protocol to achieve *verifiable*[10] quantum computing with a classical client by extending the post-hoc protocol [MF18] to a classical client setting. However, this protocol is not blind (the server learns the computation done by the client), while in this thesis we focus on blind quantum computing.

The more recent result of Alexandru Gheorghio and Thomas Vidick [GV19] (whose construction is based on [BCM+18]) also provides, similar to ours, a modular protocol for classical-client RSP protocol. They actually provide a *verifiable* classical-client RSP protocol (i.e. the client has some guarantees on the fact that server obtained the good state and not another state), and show that their protocol can be used in the Verifiable Blind Quantum Computing (VBQC [FK17]) protocol to obtain both blind and verifiable quantum computing. Actually, it is proven secure in the constructive cryptography framework, allowing general composability of the protocol (this does not rule out our own impossibility result presented in Chapter 6 since they require an additional assumption called *Measurement Buffer* effectively creating a quantum link between the simulator and the attacker). However, the security scales only polynomially with the security parameter: as a result, a polynomial distinguisher can break the security of the protocol (both at the level of the RSP functionality and when the protocol is used inside the VBQC protocol).

[9] Note however that the security relies on some unproven conjectures.

[10] *Verifiable* means that the client can verify that the result given by the server is correct.

This is true not only for the verifiable property—which is hard to avoid at the level of the RSP protocol, but may be avoidable at the level of the VBQC protocol—but also for the blindness property.

Remark 4.8.2. The reason behind this polynomial security for the blindness comes from the fact that [GV19] relies on an assumption known as "adaptive hardcore bit property", which informally states that it is hard to output y, a preimage x, a measurement b and the corresponding angle θ. However, in the actual protocol, accessing x is not possible as it would destroy the state: therefore [GV19] needs to apply a cut-and-choose approach, i.e. repeat the protocol N times, test $N-1$ of them by destroying the state to check that the adversary could know x, and outputs the remaining instance. However, there may exist a way to sample y and b such that θ is known but x cannot be recovered: this is not in contradiction with the adaptive hardcore property. If such a sampling method exists (which we do not know) then an attack would be to be honest in $N-1$ runs and to use the malicious sampling in the remaining run: if this malicious run is not tested (with probability $\frac{1}{N}$), Bob can learn θ. Note that we do not suffer from this issue since our reduction does not rely on the fact that x must be known.

In term of efficiency, [GV19] generates single qubit states, while we can also generate large hidden GHZ states using a single superposition. This allows us to produce an n qubits state using $O((n+M)N)$ operations, where M and N are very large constants describing the size of the matrix involved in the function description, while one would need $O(nMN)$ operations to create such a state using more standard methods. Moreover [GV19] uses internally a cut-and-choose approach (some runs are tested and destroyed, some other runs are used in practice), required for both the blindness and verifiable property, but explaining why the security scales polynomially. As a consequence, in order to get a decent security, [GV19] needs to create (and destroy) many times the superposition in order to obtain a single qubit.

Concerning the differences in term of method, note that all these works internally need to apply a function (notably *claw-free*, which is close to our 2-to-1 requirement) in superposition[11]. In our case, this superposition is uniform over the input set, while in the other approaches the superposition is not uniform but rather Gaussian. Having a uniform superposition allows us to have a state that is exactly the expected one (this turns out to be useful when considering LWE with polynomial noise ratio). In the other approaches, the state is never perfectly equal to the expected state, but superpolynomially close to it when relying on LWE with superpolynomial noise ratio. Of course, this is good enough in

[11] In [Mah18b] this is needed to hide a measurement and in [GV19] to hide a quantum state.

practice since it is superpolynomially hard to distinguish it from the expected quantum state.

CHAPTER 5

Function Construction

"A picture is worth a thousand words."

— A wise proverb

Figure 5.1: *Triptych of St Hippolyte* by Dieric Bouts and Hugo van der Goes (revisited)

In the previous sections, we saw how to build a protocol that fakes a quantum channel using a purely classical channel, and we saw how it turns out to be useful to do classical-client blind quantum computing. However, we left open the question of the construction of the δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family (Definition 4.2.1). In this chapter, we explain how to build such family assuming the hardness of the Learning With Errors (LWE): when

relying on the hardness of LWE with superpolynomial noise ratio we obtain a negligible δ, otherwise if we rely on the more standard assumption of LWE with polynomial noise ratio, we obtain $\delta = \frac{1}{\mathsf{poly}(\lambda)}$. We start with a quick overview of our method in Section 5.1 (introducing the LWE problem very briefly). We present in Section 5.2 the LWE problem in details, and we do a complete analysis of our construction in Section 5.3 in which we explicit a set of parameters usable for our construction depending on whether we rely on the hardness of LWE with polynomial or superpolynomial noise ratio.

5.1 Quick Overview

As a remainder, we wish to implement a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family $\{f_k\}_{k\in\mathcal{K}}$ having the following properties (see Definition 4.2.1 for the precise definition):

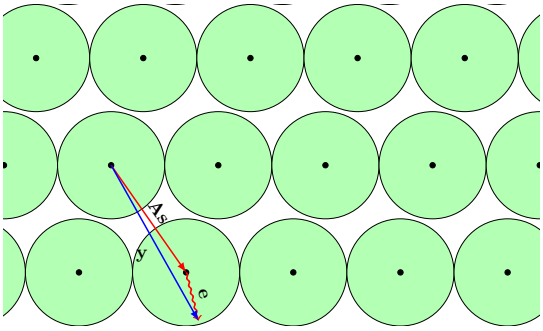
- For any $\mathbf{d}_0 \in \{0,1\}^n$ (the support) we can sample an index k and the corresponding trapdoor using $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$.
- f_k must be $\mathsf{negl}(\lambda)$-2-to-1 (i.e. an overwhelming fraction of the outputs have exactly two preimages).
- f_k can be efficiently computed given k, but should be hard to invert without t_k. Moreover, it should be hard to obtain any information on $\mathbf{d}_0$ given k.
- Given the trapdoor t_k, f_k can be efficiently inverted.
- For any $x \neq x'$ such that $f(x) = f(x')$, $h(x) \oplus h(x') = \mathbf{d}_0$.

To implement this family, we rely on the hardness assumption of the LWE problem (more details in Section 5.2): informally this problem states that given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{M\times N}$ and a vector $\mathbf{b} \in \mathbb{Z}_q^M$, it should be hard to know if $\mathbf{b}$ was sampled uniformly at random or if $b = \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{s} \in \mathbb{Z}_q^N$ was sampled uniformly at random and $\mathbf{e} \in \mathbb{Z}_q^M$ (the *noise* or *error*) was sampled according to a small discrete Gaussian[1].

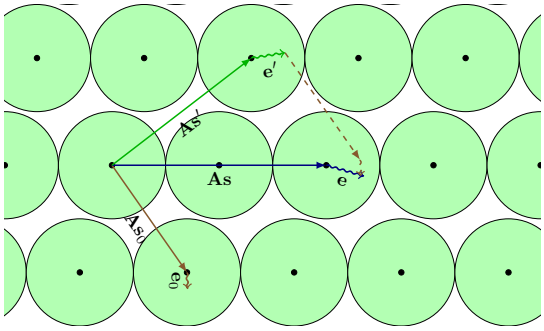
The starting point of our work is the trapdoor construction provided by [MP12]. They provide an algorithm to generate a matrix $\mathbf{A} \in \mathbb{Z}_q^{M\times N}$ (q will be a power of two) indistinguishable from a random matrix, together with a trapdoor matrix $\mathbf{R}$. If the noise $\mathbf{e} \in \mathbb{Z}_q^M$ is sufficiently small[2], the function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) \coloneqq \mathbf{A}\mathbf{s} + \mathbf{e}$ is injective. Moreover, given the trapdoor $\mathbf{R}$, one can easily invert the function $g_{\mathbf{A}}$, otherwise inverting this function is hard: $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is indistinguishable from a random vector given $\mathbf{A}$. This property is depicted in Figure 5.2a.

[1] This basically implies that $\mathbf{e}$ has small Euclidean norm.

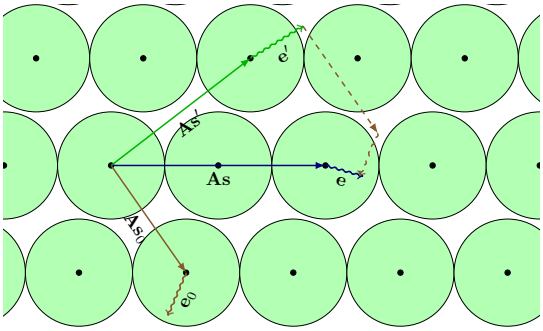
[2] In the actual construction, we also require $\mathbf{s}$ to be small because we rely on the equivalent but more efficient normal-form of LWE, but for simplicity we use the classic LWE problem in this overview.



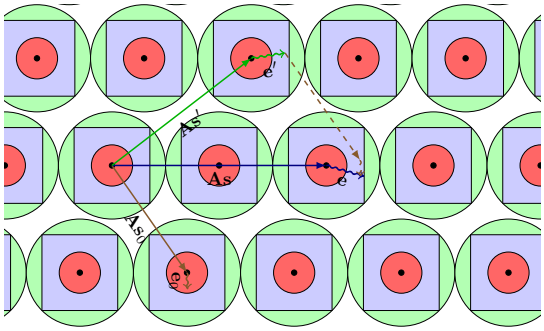
(a) Illustration of $g_{\mathbf{A}}$: given $\mathbf{y}$ it is hard to recover $\mathbf{s}$ and $\mathbf{e}$.



(b) Illustration of the 2-to-1 property: $f(\mathbf{s},\mathbf{e}) = \mathbf{As}+\mathbf{e} = \mathbf{As}'+\mathbf{e}'+\mathbf{y}_0 = f(\mathbf{s}',\mathbf{e}')$, with $\mathbf{s}' = \mathbf{s}-\mathbf{s}_0$, $\mathbf{e}' = \mathbf{e}-\mathbf{e}_0$ and $\mathbf{y}_0 = f(\mathbf{s}_0,\mathbf{e}_0)$.



(c) Some images only have a single preimage...



(d) To limit that, we sample $\mathbf{e}_0$ according to a small Gaussian (red) and $\mathbf{e}$ is defined on a much bigger hypercube (blue).

Figure 5.2: Graphical representation of the first version of the function in dimension 2. The black dots represent the lattice spanned by all points of the form $\mathbf{As}$, and the green circle is the noise domain in which one can easily invert g.

From that, we can first see how to get a δ-2-to-1 family of functions, and we will complete the construction later. Note that the larger the noise $\mathbf{e}$ and $\mathbf{e}_0$ are, the larger δ is. So a perfect (but insecure) 2-to-1 family would use $\mathbf{e} = \mathbf{e}_0 = \mathbf{0}$: therefore, to better understand this construction, it may be practical to imagine that $\mathbf{e} = \mathbf{e}_0 = \mathbf{0}$ during a first reading. The idea of the construction is to sample first a matrix $\mathbf{A}_u$ and its trapdoor $\mathbf{R}$ using the construction of [MP12][3], and then to sample an image vector $\mathbf{y}_0 \coloneqq \mathbf{A}_u\mathbf{s}_0 + \mathbf{e}_0 \in \mathbb{Z}_q^M$ (where $\mathbf{s}_0$ is sampled uniformly at random over $\mathbb{Z}_q^N$ and $\mathbf{e}_0 \in \mathbb{Z}_q^m$ is sampled according to a small discrete Gaussian). Intuitively, this $\mathbf{y}_0$ will correspond to the difference between two preimages. We define now define a first version of the

[3]This trapdoor allows us to invert our function f_k, but later, it can also be seen as a way to recover the randomness used when encrypt the bit string $\mathbf{d}_0$.

function f_k, where $k \coloneqq (\mathbf{A}_u, \mathbf{y}_0)$ as:

$$\begin{aligned} f_{\mathbf{A}_u,\mathbf{y}_0} \colon \mathbb{Z}_q^N \times E \times \{0,1\} &\longrightarrow \mathbb{Z}_q^M \\ (\mathbf{s}, \mathbf{e}, c) &\longmapsto \mathbf{A}_u\mathbf{s} + \mathbf{e} + c \times \mathbf{y}_0 \end{aligned} \tag{5.1}$$

where $E \subseteq \mathbb{Z}_q^M$ will be a set of small vectors. That way, if all vectors in E are small enough, $f_{\mathbf{A}_u,\mathbf{y}_0}$ has at most two preimages, one for $c = 0$ and one for $c = 1$ (as pictured in Figure 5.2b):

$$f(\mathbf{s}, \mathbf{e}, 0) = \mathbf{A}_u\mathbf{s} + \mathbf{e} \tag{5.2}$$

$$= \mathbf{A}_u(\mathbf{s} - \mathbf{s}_0) + (\mathbf{e} - \mathbf{e}_0) + \mathbf{y}_0 \tag{5.3}$$

$$= f(\mathbf{s} - \mathbf{s}_0, \mathbf{e} - \mathbf{e}_0, 1) \tag{5.4}$$

We remark that in order to have two preimages, we want to make sure that both $\mathbf{e} \in E$ and $\mathbf{e} - \mathbf{e}_0 \in E$ (otherwise we only get one preimage as pictured in Figure 5.2c), meaning that the intersection between E and $E - \mathbf{e}_0$ must be as big as possible. The size of this intersection will basically determine the value of δ. To have a negligible δ (and therefore a correct protocol) we want $\mathbf{e}_0$ to be as small as possible as illustrated in Figure 5.2d (so that $E - \mathbf{e}_0 \approx E$)... on the other hand if $\mathbf{e}_0$ is too small the function is not secure anymore (if $\mathbf{e}_0$ is really too small, we could for instance find $\mathbf{e}_0$ by doing an exhaustive search). These two constraints basically force us to have $\mathbf{e}_0/q$ superpolynomially small, hence forcing us to rely on LWE with superpolynomial modulus to noise ratio (or simply noise ratio). Note that the precise analysis will not fit in this first overview, and will be studied in depth in the next sections.

Now that we have a $\mathsf{negl}(\lambda)$-2-to-1 family, it is time to see how to improve it in order to obtain h such that for any two x, x' such that $f(x) = f(x')$, $h(x) \oplus h(x') = \mathbf{d}_0$. For that, we will update the previous construction and now sample $\mathbf{y}_0$ as follows: We will first sample additional lines $\mathbf{A}_l \xleftarrow{\$} \mathbb{Z}_q^{n \times N}$ to add to the matrix $\mathbf{A}_u$. Then, as before, we will sample $\mathbf{s}_0$ uniformly at random over $\mathbb{Z}_q^N$ and $\mathbf{e}_0 \in \mathbb{Z}_q^{M+n}$ will also be sampled according to a small discrete Gaussian. Finally, we compute

$$\mathbf{y}_0 \coloneqq \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right]\mathbf{s}_0 + \mathbf{e}_0 + \frac{q}{2}\left[\frac{\mathbf{0}^M}{\mathbf{d}_0}\right] \tag{5.5}$$

(we transparently interpret the bit string $\mathbf{d}_0$ as a binary vector in $\mathbb{Z}_q^N$). We update similarly our function f by adding a parameter $\mathbf{d} \in \{0,1\}^n$:

$$f_{\mathbf{A}_u,\mathbf{A}_l,\mathbf{y}_0}(\mathbf{s}, \mathbf{e}, c, \mathbf{d}) \coloneqq \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right]\mathbf{s} + \mathbf{e} + \frac{q}{2}\left[\frac{\mathbf{0}^M}{\mathbf{d}}\right] + c \times \mathbf{y}_0 \tag{5.6}$$

Intuitively this $\frac{q}{2}\mathbf{d}_0$ term adds a huge noise (much bigger than $\mathbf{e}_0$): the only way to cancel this huge noise (appearing only $c = 1$) is that a single preimage also adds such a term. More precisely, we can remark that because q is even:

$$f_{\mathbf{A}_u,\mathbf{A}_l,\mathbf{y}_0}(\mathbf{s},\mathbf{e},0,\mathbf{d}) = \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right]\mathbf{s} + \mathbf{e} + \frac{q}{2}\left[\frac{\mathbf{0}^M}{\mathbf{d}}\right] \tag{5.7}$$

$$= \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right](\mathbf{s}-\mathbf{s}_0) + (\mathbf{e}-\mathbf{e}_0) + \frac{q}{2}\left[\frac{\mathbf{0}^M}{\mathbf{d}\oplus\mathbf{d}_0}\right] + \mathbf{y}_0 \tag{5.8}$$

$$= f_{\mathbf{A}_u,\mathbf{A}_l,\mathbf{y}_0}(\mathbf{s}-\mathbf{s}_0,\mathbf{e}-\mathbf{e}_0,1,\mathbf{d}\oplus\mathbf{d}_0) \tag{5.9}$$

and that therefore (skipping a small technicality) for any two preimages $(\mathbf{s},\mathbf{e},0,\mathbf{d})$ and $(\mathbf{s}',\mathbf{e}',1,\mathbf{d}')$, we have $\mathbf{d}\oplus\mathbf{d}' = \mathbf{d}_0$. So by simply defining $h(\mathbf{s},\mathbf{e},c,\mathbf{d}) = \mathbf{d}$ we get the XOR property. The role of the $\frac{q}{2}$ in the above equations is now clearer, it allows to turn an addition modulo q into an addition modulo 2 (used in our XOR property) since $\frac{q}{2}(\mathbf{d}-\mathbf{d}_0) = \frac{q}{2}\mathbf{d}\oplus\mathbf{d}_0$.

The trapdoor property is a fairly direct consequence of the construction of [MP12]: using the trapdoor $\mathbf{R}$ on the upper part of f, we can learn $\mathbf{s}$ and the upper part of $\mathbf{e}$. Then, we can consider the lower part of the matrix, remove from it $\mathbf{A}_l\mathbf{s}$: we are left with $\mathbf{e}_l + \frac{q}{2}\mathbf{d}$. Because $\mathbf{e}_l$ is small, we can learn $\mathbf{d}$ by checking if the components are closer to 0 or to $\frac{q}{2}$. To obtain the other preimage, we compute $(\mathbf{s}-\mathbf{s}_0,\mathbf{e}-\mathbf{e}_0,\mathbf{d}\oplus\mathbf{d}_0)$.

Finally, we are left with the indistinguishability property: since $\mathbf{A}_u$ is indistinguishable from a random matrix, and $\mathbf{A}_l$ is actually a random matrix, thus $\mathbf{A} \coloneqq \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right]$ is indistinguishable from a random matrix. But, under the hardness assumption of LWE, $\mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$ is indistinguishable from a random vector. Therefore, since adding a constant vector to a uniformly sampled vector does not change its distribution, one cannot distinguish $\mathbf{A}\mathbf{s}_0 + \mathbf{e}_0$ from $\mathbf{A}\mathbf{s}_0 + \mathbf{e}_0 + \left[\frac{\mathbf{0}^M}{\mathbf{d}_0}\right]$, or from any vector of the form $\mathbf{A}\mathbf{s}_0 + \mathbf{e}_0 + \left[\frac{\mathbf{0}^M}{\mathbf{d}}\right]$.

We provide in the next sections a more in-depth analysis in order to properly handle the noise, we find an explicit set of parameters allowing f to be $\mathsf{negl}(\lambda)$-2-to-1, and we give a method to prove that a maliciously sampled f is indeed $\mathsf{negl}(\lambda)$-2-to-1 and has the XOR property. While this is not required right now, it will prove useful when considering Non-Interactive and Non-Destructive Zero-Knowledge proofs on Quantum States in Chapter 7.

We use in this work the LWE problem. However, one may wonder whether other post-quantum hardness problems can be used. Two others major post-quantum candidates are code-based and isogeny-based cryptography. I've not tried to use isogeny-based cryptography, but I tried to construct f_k using code-based cryptography. We discuss more on that in Section 5.4, but long story short, I was not able to find a code-based construction as the errors in code-based cryptography add-up to quickly compared to lattice-based cryptography.

5.2 Introduction to the Learning With Errors (LWE) problem

In this section, we formalize the definition of the LWE problem, derive several useful properties and describe the construction of [MP12].

5.2.1 Definitions

The Learning With Errors (LWE) problem was introduced in [Reg05].

Definition 5.2.1 (Learning With Errors (LWE) [Reg05])**.**
Let $N \in \mathbb{N}$ [4], $q = q(N) \in \mathbb{N}_{\geq 2}$ be a modulus and χ a distribution on $\mathbb{R}$ (χ may be continuous or discrete ($\chi \subseteq \mathbb{Z}$) and will always be reduced modulo q). For any $s \in \mathbb{Z}_q^N$, we define $A_{\mathbf{s},\chi}$ as the distribution on $\mathbb{Z}_q^N \times [0,q)$ (or $\mathbb{Z}_q^N \times \mathbb{Z}_q$ if χ is discrete) obtained by sampling $\mathbf{a} \in \mathbb{Z}_q^N$ uniformly at random, $e \leftarrow \chi$ and outputting $(\mathbf{a}, \mathbf{a}^T\mathbf{s} + e \bmod q)$.

We say that an algorithm solves the search-$\mathsf{LWE}_{q,\chi}$ *problem (in the worst case, with overwhelming probability) if for any $\mathbf{s} \in \mathbb{Z}_q^N$, given an arbitrary number of samples from $A_{\mathbf{s},\chi}$, it outputs $\mathbf{s}$ with overwhelming probability. We say that an algorithm solves the* decision-$\mathsf{LWE}_{q,\chi}$ *problem (on average, with non-negligible advantage) if it can distinguish with non-negligible advantage between the distribution $A_{\mathbf{s},\chi}$ where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^N$, and the uniform distribution $U := \mathcal{U}(\mathbb{Z}_q^N \times [0,q))$ (when χ is discrete, we consider instead the uniform distribution $U := \mathcal{U}(\mathbb{Z}_q^N \times \mathbb{Z}_q)$).*

We can also formulate this problem using matrices by grouping a fixed number $M \in \mathbb{N}$ of samples: For any $\mathbf{s} \in \mathbb{Z}_q^N$, we can sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{M \times N}$ and $\mathbf{e} \leftarrow \chi^M$ a (typically small) vector where each of its component is sampled according to χ. Then let $\mathbf{b} := \mathbf{A}\mathbf{s} + \mathbf{e}$. The search *problem consists in finding $\mathbf{s}$ given $(\mathbf{A}, \mathbf{b})$. The* decision *problem consists in*

[4] While usually the parameters N is written in lowercase, we will use this notation here since n already represents the size of the support.

deciding, when receiving a couple $(\mathbf{A}, \mathbf{b})$*, if* $\mathbf{b}$ *has been sampled uniformly at random (over* $[0, q)^M$ *if* $\mathcal{X}$ *is continuous, or over* $\mathbb{Z}_q^m$ *if* χ *is discrete) or if* $\mathbf{b}$ *has been sampled according to the procedure described above (and therefore* $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$*).*

Note that having access to less samples can only make the problem harder. On the other hand, one can show that having access to polynomially many samples is enough to generate arbitrary many further samples, with only a minor degradation in the error [GPV08, ACP+09, APS15]. Note also that the the worst-case and average-case decision problems are in fact equivalent: [Reg05, Lem. 4.1] shows how it is possible to turn a distinguisher that can solve the decision-LWE problem with non-negligible advantage into a better distinguisher that can solve the decision-LWE problem with overwhelming probability.

The distribution χ can be instantiated in many different ways: for example when χ is always equal to 0, these problems are trivial, and when χ is uniform, they are impossible. In practice, χ is usually a (discrete or continuous) Gaussian (or more rarely a rounded Gaussian [Reg05]):

Definition 5.2.2 (Continuous and discrete Gaussian)**.** *For any* $s \in \mathbb{R}_{>0}$ *and any vector* $\mathbf{x} \in \mathbb{R}^N$*, we define* $\rho_s(\mathbf{x}) \coloneqq \exp\left(-\pi \left(\frac{\|\mathbf{x}\|_2}{s}\right)^2\right) = \exp(-\pi \mathbf{x}^T\mathbf{x}/s^2)$*. By applying a linear transformation on* $\mathbf{x}$*, we can generalize this notion: for any positive-definite matrix* $\Sigma > \mathbf{0}$*, we define:*

$$\rho_{\sqrt{\Sigma}}(\mathbf{x}) \coloneqq \exp(-\pi \cdot \mathbf{x}^T\Sigma^{-1}\mathbf{x}) \tag{5.10}$$

In particular, if $\Sigma = s^2\mathbf{I}$*, we have* $\rho_s = \rho_{\sqrt{\Sigma}}$*. The normalization of the expression gives* $\int_{\mathbb{R}^n} \rho_{\sqrt{\Sigma}}(\mathbf{x}) = \sqrt{\det \Sigma}$ *and* $\int_{\mathbb{R}^n} \rho_s(\mathbf{x}) = s^N$*. We can now define the continuous Gaussian distribution:*

$$\mathcal{D}_s^N(\mathbf{x}) \coloneqq \rho_s(\mathbf{x})/s^N \qquad \mathcal{D}_{\sqrt{\Sigma}}(\mathbf{x}) \coloneqq \rho_{\sqrt{\Sigma}}(\mathbf{x})/\sqrt{\det \Sigma} \tag{5.11}$$

Note that sampling from $\mathcal{D}_s^N$ *is equivalent to sampling each component from* $\mathcal{D}_s \coloneqq \mathcal{D}_s^1$*. Moreover, due to our choice of normalization,* s *and* Σ *are not exactly equal to the standard deviation and to the covariance matrix:* $\mathcal{D}_s$ *has standard deviation* $\sigma \coloneqq s/(\sqrt{2\pi})$ *and the actual covariance of* $\mathcal{D}_{\sqrt{\Sigma}}$ *is* $\Sigma' \coloneqq \Sigma/(2\pi)$*.*

If $\Lambda \subseteq \mathbb{R}^N$ *is a lattice (i.e. a discrete additive subgroup of* $\mathbb{R}^N$*), we define for any* $\mathbf{c} \in \mathbb{R}^N$ *the coset* $\Lambda + \mathbf{c} = \{\mathbf{x} + \mathbf{c} \mid \mathbf{x} \in \Lambda\}$ *and* $\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c}) \coloneqq \sum_{\mathbf{x} \in \Lambda + \mathbf{c}} \rho_{\sqrt{\Sigma}}(\mathbf{x})$*. We can define now the discrete Gaussian on* $\Lambda + \mathbf{c}$ *by simply normalizing* $\rho_{\sqrt{\Sigma}}(\mathbf{x})$*. For any*

$\mathbf{x} \in \mathbb{R}^N$, *if* $\mathbf{x} \notin \Lambda + \mathbf{c}$, $\mathcal{D}_{\Lambda+\mathbf{c},\sqrt{\Sigma}}(\mathbf{x}) = 0$ *and if* $\mathbf{x} \in \Lambda + \mathbf{c}$*:*

$$\mathcal{D}_{\Lambda+\mathbf{c},\sqrt{\Sigma}}(\mathbf{x}) \coloneqq \frac{\rho_{\sqrt{\Sigma}}(\mathbf{x})}{\rho_{\sqrt{\Sigma}}(\Lambda + \mathbf{c})} \tag{5.12}$$

In the following, for $\alpha \in (0,1)$*, the discrete Gaussian on* $\mathbb{Z}$*,* $\mathcal{D}_{\mathbb{Z},\alpha q} = \mathcal{D}_{\mathbb{Z},\sqrt{(\alpha q)^2 \mathbf{I}}}$*, will be particularly important to sample the noise.*

Note that [Reg05] defined originally a different kind of discrete Gaussian (not equivalent to the "true" discrete Gaussians we just defined) that we call *rounded Gaussians*. To sample from such distribution, we first sample from a continuous Gaussian and round the result to the nearest integer modulo q. While reductions are easier to prove using rounded Gaussians, they do not benefit from some advantageous properties of discrete Gaussians: for instance the sum of two rounded Gaussians may not be a rounded Gaussian, and it is harder to bound the singular value of rounded Gaussians (this will be required in Lemma 5.2.10).

The next lemma is useful to bound the length of a vector sampled according to a discrete Gaussian.

Lemma 5.2.3 (Particular case of [Ban93, Lem. 1.5][MP12, Lem. 2.6])**.** *For any* $s > 0$*, we have:*

$$\Pr\left[\|\mathbf{x}\|_2 \geq s\sqrt{n} \;\middle|\; \mathbf{x} \leftarrow \mathcal{D}^n_{\mathbb{Z},s} \right] \leq 2^{-n} \tag{5.13}$$

5.2.2 Hardness of LWE

The LWE problems are widely supposed to be hard to solve even for quantum computers and are the basic building block of many post-quantum cryptographic protocols [Pei16]. In particular, it is on average as hard as worst-case problems on lattices (the precise definition of these problems is not important for this thesis).

Lemma 5.2.4 (Hardness of LWE [PRS17])**.** *Let* N, q *be integers and* $\alpha \in (0,1)$ *be such that* $\alpha q > 2\sqrt{N}$*. If there exists an algorithm that solves decision-*$\mathsf{LWE}_{q,\mathcal{D}_{\alpha q}}$*, then there exists an efficient quantum algorithm that approximates the decision version*[5] *of the shortest vector problem (*GapSVP_γ*) and the shortest independent vectors problem (*SIVP_γ*) to within* $\gamma \coloneqq \tilde{O}(N/\alpha)$*.*

[5] Note that the original reduction from [Reg05] targets the search version: however, the search-to-decision reductions typically add some small loses in the parameters, like [MP12]. This direct reduction to decision-LWE is therefore more efficient.

To have strong security guarantees, we usually want γ to be polynomial in N, but this is sometimes impossible. According to [Pei16] the best algorithm for solving these problems in polynomial time works for only slightly sub-exponential approximation factor $\gamma = 2^{\Theta(N \log\log N/\log N)}$. The algorithm [Sch87] also provides a tradeoff between the approximation γ and the running time: an approximation of $\gamma = 2^k$ can be obtained in time $2^{\tilde{\Theta}(N/k)}$. This suggest that there is no efficient algorithm for $\gamma = 2^{N^\epsilon}$ for $\epsilon \in (0, \frac{1}{2})$ (the algorithm provided by [Sch87] would indeed run in subexponential time $2^{N^{1-\epsilon}}$). In practice, many works rely on the security of LWE when γ is superpolynomial ([BGG+14] uses for example the above assumption that $\gamma = 2^{N^\epsilon}$ for some $\epsilon \in (0, \frac{1}{2})$), and we will sometimes need this same assumption.

While the hardness assumption given in Lemma 5.2.4 targets a continuous noise distribution, it can also be adapted to discrete Gaussians. In particular, [Pei10, Thm. 3.1] can be used to show that if decision-$\mathsf{LWE}_{q,\mathcal{D}_{\alpha q}}$ is hard, then the discrete version decision-$\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},s}}$ is hard for $s := \sqrt{(\alpha q)^2 + \omega\left(\sqrt{\log\lambda}\right)^2}$, where $\omega(\sqrt{\log N})$ denotes any function, fixed across all the thesis, such that $\lim_{N\to\infty} \sqrt{\log N}/\omega(\sqrt{\log N}) = 0$ (for instance, we can take $\omega(\sqrt{\log N}) = \log N$). More precisely, [Pei10] gives a method to turn any sample from $\mathsf{LWE}_{q,\mathcal{D}_{\alpha q}}$ into a sample (indistinguishable from a sample) of $\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},s}}$.

Corollary 5.2.5 (From continuous Gaussian to discrete Gaussian, corollary of Thm. 3.1, [Pei10])**.** *Let $\lambda, q \in \mathbb{N}$, $\alpha \in (0,1)$. If e_c is sampled according to $\mathcal{D}_{\alpha q}$ and $e \leftarrow e_c + \mathcal{D}_{\mathbb{Z}-e_c,\omega\left(\sqrt{\log\lambda}\right)}$, then the marginal distribution of e is within negligible statistical distance*

$$\Delta = \frac{1}{\exp(\pi\omega(\sqrt{\log\lambda})^2 - \ln(2\lambda)) - 1} = \mathsf{negl}(\lambda) \tag{5.14}$$

of $\mathcal{D}_{\mathbb{Z},s}$ with $s := \sqrt{(\alpha q)^2 + \omega\left(\sqrt{\log\lambda}\right)^2}$. Moreover, if there exists instead $x \in \mathbb{Z}_q$ (for example $x = \mathbf{a}^T\mathbf{s}$ for some $\mathbf{s} \in \mathbb{Z}_q^N$ and $\mathbf{a} \in \mathbb{Z}_q^N$) such that e_c is distributed according to $x + \mathcal{D}_{\alpha q} \bmod q$, then the statistical distance between the distribution $e \bmod q$ and $x + \mathcal{D}_{\mathbb{Z},s} \bmod q$ is $\Delta = \mathsf{negl}(\lambda)$. Finally, if e_c is uniformly sampled over $[0, q)$, the marginal distribution of e is uniform over $\mathbb{Z}_q$.

Proof. The first part of this corollary is a direct application of [Pei10, Thm. 3.1]: We define $\mathbf{c}_1 = \mathbf{0}$, $\Lambda_1 = \mathbb{Z}$, $\Sigma_2 = (\alpha q)^2\mathbf{I}_1$, $\Sigma_1 = \omega(\sqrt{\log\lambda})^2\mathbf{I}_1$ with $\sqrt{\Sigma_1} = \omega(\sqrt{\log\lambda}) \geq \eta_\varepsilon(\mathbb{Z})$, where ε is a negligible function of N (the last inequality comes from [Pei10, Lem. 2.5]). When $\mathbf{x}_2 (= e_c)$ is chosen according to a continuous Gaussian, the marginal distribution of e is within statistical distance $8\varepsilon = \mathsf{negl}(\lambda)$ of $\mathcal{D}_{\mathbb{Z},\sqrt{\Sigma}}$ (the constant can actually be

improved in this specific case), with $\Sigma = \Sigma_1 + \Sigma_2 = (\alpha q)^2\mathbf{I}_1 + \omega(\sqrt{\log\lambda})^2\mathbf{I}_1 = s^2\mathbf{I}_1$, which concludes the first part of the proof.

To see that the equality also holds when $e_c \leftarrow x + \mathcal{D}_{\alpha q} \bmod q$ for some $x \in \mathbb{Z}$, we remark that for any $\bar{e} \in \mathbb{Z}_q$ and for any $e_c = x + e'_c \in \mathbb{R}$:

$$p \coloneqq \Pr\left[\bar{e} = e \bmod q \;\middle|\; e \leftarrow e_c + \mathcal{D}_{\mathbb{Z}-e_c,\omega\left(\sqrt{\log\lambda}\right)}\right] \tag{5.15}$$

$$= \Pr\left[\bar{e} = e \;\middle|\; e \leftarrow e_c + \mathcal{D}_{\mathbb{Z}-e_c,\omega\left(\sqrt{\log\lambda}\right)} \bmod q\right] \tag{5.16}$$

$$= \Pr\left[\bar{e} = e \;\middle|\; e \leftarrow x + e'_c + \mathcal{D}_{\mathbb{Z}-e'_c,\omega\left(\sqrt{\log\lambda}\right)} \bmod q\right] \tag{5.17}$$

where the last equality comes from $\mathbb{Z} - e_c = \mathbb{Z} - (e'_c \bmod q)$. But we already know that $e'_c + \mathcal{D}_{\mathbb{Z}-e'_c,\omega\left(\sqrt{\log\lambda}\right)}$ is statistically close to $\mathcal{D}_{\mathbb{Z},s}$ from the first part of the corollary, which concludes this part of the proof.

Now, let us assume that e_c is sampled uniformly at random over $[0, q)$. Because e_c can be uniquely decomposed into $e_c = e_{c,1} + e_{c,2}$ where $e_{c,1} \in \{0, \ldots, q-1\}$ and $e_{c,2} \in [0, 1)$, and because $\mathbb{Z} - e_c = \mathbb{Z}^n - e_{c,2}$ we have:

$$\Pr\left[\bar{e} = e \bmod q \;\middle|\; e_c \leftarrow [0,q), e \leftarrow e_c + \mathcal{D}_{\mathbb{Z}-e_c,\omega\left(\sqrt{\log\lambda}\right)}\right]$$

$$= \frac{1}{q}\int_0^1 de_{c,2} \sum_{e_{c,1}=0}^{q-1} \sum_{\substack{e\in\mathbb{Z} \\ \bar{e}=e[q]}} \frac{\rho_{\omega\left(\sqrt{\log\lambda}\right)}(e - e_{c,1} - e_{c,2})}{\rho_{\omega\left(\sqrt{\log\lambda}\right)}(\mathbb{Z} - e_{c,2})}$$

Similarly, because any integer $\hat{e} \in \mathbb{Z}$ can be decomposed uniquely into $\hat{e} = e - e_{c,1}$ where $e_{c,1} \in \{0, \ldots, q-1\}$, and $e = \bar{e} \bmod q$, we can merge the two sums into a single sum over $\hat{e} \in \mathbb{Z}$, replace the $e - e_{c,1}$ with $\hat{e}$, and use the fact that $\sum_{\hat{e}\in\mathbb{Z}} \rho_\omega(\hat{e} - e_{c,2}) = \rho_\omega(\mathbb{Z} - e_{c,2})$ to conclude that the probability is equal to $1/q$: it corresponds to a uniform sampling over $\mathbb{Z}_q$. □

While in the usual LWE problem, $\mathbf{s}$ is sampled uniformly at random over $\mathbb{Z}_q^N$, it is also possible to sample $\mathbf{s}$ according to a small Gaussian. Since, as we will see, this can be seen as a reformulation of the problem in its Hermite Normal form (HNF), this new sampling method is actually at least as secure as the initial uniform sampling, and it appears to be more efficient. Moreover, the construction given in [MP12] will naturally be formulated in this form, so in this work we will also sample $\mathbf{s}$ according to a small

Gaussian (however, one can easily come back to the initial formulation as we will see later).

Lemma 5.2.6 (Normal LWE problem [ACP+09, Lem. 2]). *Let $q = p^k$ be a prime power. There is a deterministic polynomial-time transformation T that, for arbitrary $\mathbf{s} \in \mathbb{Z}_q^N$ and error distribution χ, maps $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{s}},\chi}$ where $\bar{\mathbf{s}} \leftarrow \chi^N$, and maps $U(\mathbb{Z}_q^N \times \mathbb{Z}_q)$ to itself.*

The idea of the proof given in [ACP+09, Lem. 2] (following [MR09], see also [Pei16, p. 23]) is to first obtain and select enough samples $(\bar{\mathbf{A}}, \bar{\mathbf{y}} \coloneqq \bar{\mathbf{A}}\mathbf{s} + \bar{\mathbf{s}})$, $\bar{\mathbf{s}}$ being sampled according to χ, to ensure that $\bar{\mathbf{A}} \in \mathbb{Z}_q^{N \times N}$ is invertible. Then, any new sample $(\mathbf{a}, y \coloneqq \langle \mathbf{a}, \mathbf{s} \rangle + e)$ can be updated into $(\mathbf{a}', b')$ where $\mathbf{a}' \coloneqq -(\bar{\mathbf{A}}^T)^{-1}\mathbf{a}$ and $b' \coloneqq y + \langle \mathbf{a}', \bar{\mathbf{y}} \rangle = \langle \mathbf{a}', \bar{\mathbf{s}} \rangle + e$.

5.2.3 The [MP12] Construction

We saw that the LWE problem can provide one way functions (because it is hard to recover $\mathbf{s}$ and $\mathbf{e}$ from $\mathbf{A}\mathbf{s} + \mathbf{e}$), but it turns out that LWE can also be used to introduce a trapdoor inside $\mathbf{A}$ such that this function can easily be inverted [GGH97, Ajt99, AP11, MP12]. In order to realize the primitives described in Definition 4.2.1, we will use the trapdoor system presented in [MP12]. This work introduced an algorithm `MP.Gen` that samples a matrix $\mathbf{A}$ and a trapdoor $\mathbf{R}$. In addition, $\mathbf{A}$ is indistinguishable from a random matrix (without $\mathbf{R}$), and $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) \coloneqq \mathbf{A}\mathbf{s} + \mathbf{e}$ is injective and can be inverted given $\mathbf{R}$ for any $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}$ ($\mathcal{X}$ will be defined later as sets of elements having a small norm).

In this thesis, we will focus on the (more efficient) computationally-secure construction presented in [MP12] (we also require the modulus $q \coloneqq 2^k$ to be a power of 2), but the same method should extend to other constructions with even q. We give in Definition 5.2.7 the construction we will use, and we explain after the intuition behind it. The reader not interested by the choice of parameters can safely skip the rest of the section.

Definition 5.2.7 ([MP12]). *Let $\lambda \in \mathbb{N}$ be a security parameter, and $\mathcal{P}_0 = (k, N, \alpha, r_{\max})$ with $(k, N) \in \mathbb{N}^2$, $\alpha \in (0, 1)$ and $r_{\max} \in \mathbb{R}$, be some parameters that can depend on λ. We define $M \coloneqq N(1 + k)$, $q \coloneqq 2^k$, $\mathcal{X}_g \coloneqq \left\{ (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^M \;\middle|\; \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_2 \leq r_{\max} \right\}$,*

$\mathbf{g} \coloneqq \begin{bmatrix} 1 & 2 & 4 & \dots & 2^{k-1} \end{bmatrix}^T \in \mathbb{Z}_q^k$, *and the gadget matrix* $\mathbf{G}$ *as:*

$$\mathbf{G} \coloneqq \mathbf{I}_n \otimes \mathbf{g} = \begin{bmatrix} \vdots & & \\ \mathbf{g} & & \\ \vdots & & \\ & \ddots & \\ & & \vdots \\ & & \mathbf{g} \\ & & \vdots \end{bmatrix} \in \mathbb{Z}_q^{Nk \times N} \tag{5.18}$$

We define now in Algorithm 2 a procedure to sample a public matrix and its trapdoor $(\mathbf{A}, \mathbf{R}) \leftarrow \mathit{MP.Gen}_{\mathcal{P}_0}(1^\lambda)$*, and for any* $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}_g$ *we define* $\mathbf{y} \coloneqq g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ *and its inversion procedure* $(\tilde{\mathbf{s}}, \tilde{\mathbf{e}}) \coloneqq \mathit{MP.Invert}(\mathbf{R}, \mathbf{A}, \mathbf{y})$.

Algorithm 2 Construction from [MP12]

$\mathtt{MP.Gen}_{\mathcal{P}_0}(1^\lambda)$

1 : $\hat{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{N \times N}$
2 : $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \mid \mathbf{R}_2 \end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}^{Nk \times 2N}$
3 : $\mathbf{A} \coloneqq \begin{bmatrix} \hat{\mathbf{A}} \\ \hline \mathbf{G} - \mathbf{R}_2\hat{\mathbf{A}} - \mathbf{R}_1 \end{bmatrix} \in \mathbb{Z}_q^{M \times N}$
4 : **return** $(\mathbf{A}, \mathbf{R})$

$g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$

1 : **return** $\mathbf{As} + \mathbf{e}$

$\mathtt{MP.Invert}_{\mathcal{P}_0}(\mathbf{R} \coloneqq \begin{bmatrix} \mathbf{R}_1 \mid \mathbf{R}_2 \end{bmatrix}, \mathbf{A}, \mathbf{y})$

1 : // Return $(\tilde{\mathbf{s}}, \tilde{\mathbf{e}}) \in \mathcal{X}_g$ s.t. $\mathbf{y} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$
2 : $\tilde{\mathbf{s}} \coloneqq \mathtt{InvertGadget}_{\mathcal{P}_0}(\begin{bmatrix} \mathbf{R}_2 \mid \mathbf{I}_{Nk} \end{bmatrix}\mathbf{y})$
3 : $\tilde{\mathbf{e}} \coloneqq \mathbf{y} - \mathbf{A}\tilde{\mathbf{s}}$
4 : **if** $\left\| \begin{bmatrix} \tilde{\mathbf{s}} \\ \hline \tilde{\mathbf{e}} \end{bmatrix} \right\|_2 > r_{\max}$ **return** $\perp$ **fi**
5 : **return** $(\tilde{\mathbf{s}}, \tilde{\mathbf{e}})$

$\mathtt{InvertSmallGadget}_{\mathcal{P}_0}(\mathbf{y} = \begin{bmatrix} y_0 & \dots y_{k-1} \end{bmatrix}^T)$

1 : // Return $s \in \mathbb{Z}_q$ such that $\mathbf{y} = \mathbf{g}s + \mathbf{e}$
2 : $s \coloneqq 0$
3 : **for** $i = k-1, \dots, 0$ **do**
4 : **if** $y_i - 2^i s \notin \left[-\frac{q}{4}, \frac{q}{4}\right) \bmod q$
5 : $s \coloneqq s + 2^{k-1-i}$ **fi endfor**
6 : **return** s

$\mathtt{InvertGadget}_{\mathcal{P}_0}(\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \dots \mathbf{y}_N^T \end{bmatrix}^T)$

1 : // Return $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\mathbf{y} = \mathbf{Gs} + \mathbf{e}$
2 : **for** $i = 1, \dots, N$ **do**
3 : $s_i \coloneqq \mathtt{InvertSmallGadget}_{\mathcal{P}_0}(\mathbf{y}_i)$
4 : **endfor**
5 : $\mathbf{s} \coloneqq \begin{bmatrix} s_1 & \dots & s_n \end{bmatrix}^T$
6 : **return s**

The idea of the construction given in Definition 5.2.7 is to use a gadget matrix $\mathbf{G}$ which is easy to invert even in the presence of noise, and then to hide this matrix inside a random looking matrix $\mathbf{A}$. $\mathbf{G}$ is easy to invert because $\mathbf{G}$ basically encodes all bits of the binary representation of each component of $\mathbf{s}$ in a different component (where a 1 is

encoded by $q/2$+noise, and 0 is encoded by 0+noise): the inversion of $\mathbf{G}$ is doable by a rounding operation, starting from the least significants bits of the components of $\mathbf{s}$. Then, as we will see, $\mathbf{R}$ can be used to invert $\mathbf{As}+\mathbf{e}$: with $\mathbf{R}$ we can obtain a vector of the form $\mathbf{Gs}+\mathbf{e}'$ ($\mathbf{e}'$ is small if $\mathbf{R}$ has sufficiently small singular values), and then since $\mathbf{G}$ is easy to invert we can obtain $\mathbf{s}$ easily. We formalize now these statements.

We describe now conditions that are sufficient to ensure that $g_{\mathbf{A}}$ is injective.

Lemma 5.2.8 ([MP12])**.** *If $\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},\alpha q}}$ is hard and if $Nk = \mathsf{poly}(\lambda)$, then the matrix $\mathbf{A}$ obtained via `MP.Gen` is indistinguishable from a uniform random matrix.*

Proof. For completeness, we sketch the proof given in [MP12]. Since $\mathbf{G}$ is a fixed matrix, it is easy to subtract $\mathbf{G}$ from $\mathbf{A}$ and transpose the matrices: $\mathbf{A}$ looks random iff $(\hat{\mathbf{A}}^T, \hat{\mathbf{A}}^T\mathbf{R}_2^T + \mathbf{R}_1^T)$ looks random. But this is nearly an exact LWE instance in its normal form ($\mathbf{R}_2$ is indeed sampled according to a small Gaussian). The only difference is that the $\hat{\mathbf{A}}$ samples are "reused" multiple times since $\mathbf{R}_i^T$ are matrices and not vectors. However, as shown in [PW08, Lem. 6.2], an hybrid argument can be made (by gradually replacing each column with random elements) to prove that it is still hard to distinguish it from a random matrix if $\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},\alpha q}}$ is hard (since we obtain Nk hybrid games, we need $Nk = \mathsf{poly}(\lambda)$). □

Remark 5.2.9. It is possible to easily translate the normal form into a more usual form in which $\mathbf{s}$ is sampled uniformly at random: one can sample a random invertible matrix $\mathbf{A}_r \in \mathbb{Z}_q^{N\times N}$, and define $\mathbf{A}' \coloneqq \left[\begin{array}{c}\mathbf{I}\\\hline\mathbf{A}\end{array}\right]\mathbf{A}_r$.

Lemma 5.2.10. *Let $\hat{\mathbf{A}} \in \mathbb{Z}_q^{N\times N}$, $(\mathbf{R}_1, \mathbf{R}_2) \in (\mathbb{Z}^{Nk\times N})^2$, $\mathbf{A} \coloneqq \left[\begin{array}{c}\hat{\mathbf{A}}\\\hline\mathbf{G}-\mathbf{R}_2\hat{\mathbf{A}}-\mathbf{R}_1\end{array}\right] \in \mathbb{Z}_q^{M\times N}$. If the highest singular value $\sigma_{\max}(\mathbf{R})$ of $\mathbf{R}$ is such that $\sqrt{\sigma_{\max}(\mathbf{R})+1} < \frac{q}{4r_{\max}}$, then $g_{\mathbf{A}} : \mathcal{X}_g \to \mathbb{Z}_q^M$ is injective and for all $(\mathbf{s},\mathbf{e}) \in \mathcal{X}_g$, `MP.Invert`$(\left[\mathbf{R}_1 \,\middle|\, \mathbf{R}_2\right], \mathbf{A}, \mathbf{As}+\mathbf{e}) = (\mathbf{s},\mathbf{e})$.*

Moreover, if we denote by $C \approx \frac{1}{\sqrt{2\pi}}$ the universal constant defined in [MP12, Lem. 1.9], and if we have for the parameters $\mathcal{P}_0$ defined in Definition 5.2.7:

$$\sqrt{\left(C\times\alpha q\times\sqrt{N}(\sqrt{k}+\sqrt{2}+1)\right)^2+1} \le \frac{q}{4r_{\max}} \tag{5.19}$$

then with overwhelming probability (on N) $\ge 1-2e^{-N}$, we have $\sqrt{\sigma_{\max}(\mathbf{R})^2+1} < \frac{q}{4r_{\max}}$ and therefore $g_{\mathbf{A}}$ is injective.

Proof. This proof can be obtained by combining different theorems from [MP12]. For completeness, we put the full proof of this lemma here. For the injectivity, we assume that there exist $(\mathbf{s}, \mathbf{s}') \in S$ and $(\mathbf{e}, \mathbf{e}') \in E$ such that $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = g_{\mathbf{A}}(\mathbf{s}', \mathbf{e}')$. So $\mathbf{A}(\mathbf{s} - \mathbf{s}') + (\mathbf{e} - \mathbf{e}') = \mathbf{0}$, and it is enough now to prove that $\mathbf{s} = \mathbf{s}'$ to obtain $\mathbf{e} = \mathbf{e}'$. We multiply (on the left) this equation by $\left[\mathbf{R}_2 \;\middle|\; \mathbf{I}_{Nk}\right]$: we obtain $\mathbf{G}(\mathbf{s}-\mathbf{s}') - \mathbf{R}_1(\mathbf{s}-\mathbf{s}') + \left[\mathbf{R}_2 \;\middle|\; \mathbf{I}_{Nk}\right](\mathbf{e}-\mathbf{e}') = \mathbf{0}$, i.e. if we define $\tilde{\mathbf{R}} \coloneqq \left[-\mathbf{R}_1 \;\middle|\; \mathbf{R}_2 \;\middle|\; \mathbf{I}_{Nk}\right]$, $\tilde{\mathbf{e}} \coloneqq \tilde{\mathbf{R}} \begin{bmatrix} \mathbf{s} - \mathbf{s}' \\ \mathbf{e} - \mathbf{e}' \end{bmatrix}$ and $\tilde{\mathbf{s}} = \mathbf{s} - \mathbf{s}'$, we have $\mathbf{G}\tilde{\mathbf{s}} + \tilde{\mathbf{e}} = \mathbf{0}$. First, we remark that:

$$
\begin{aligned}
\sigma_{\max}(\tilde{\mathbf{R}^T}) &= \|\tilde{\mathbf{R}^T}\|_2 \\
&= \max_{x, \|x\|_2 = 1} \left\| \begin{bmatrix} -\mathbf{R}_1^T \\ \mathbf{R}_2^T \\ \mathbf{I}_{Nk} \end{bmatrix} x \right\|_2 \\
&= \max_{x, \|x\|_2 = 1} \sqrt{\left\| \begin{bmatrix} -\mathbf{R}_1^T \\ \mathbf{R}_2^T \end{bmatrix} x \right\|_2^2 + \|x\|_2^2} \\
&= \sqrt{\max_{x, \|x\|_2 = 1} \left\| \begin{bmatrix} \mathbf{R}_1^T \\ \mathbf{R}_2^T \end{bmatrix} x \right\|_2^2 + 1} \\
&= \sqrt{\max_{x, \|x\|_2 = 1} \left\| \mathbf{R}^T x \right\|_2^2 + 1} \\
&= \sqrt{\sigma_{\max}\left(\mathbf{R}^T\right)^2 + 1} \\
&= \sqrt{\sigma_{\max}\left(\mathbf{R}\right)^2 + 1}
\end{aligned} \tag{5.20}
$$

We prove now that for all i, $\tilde{\mathbf{e}}[i] \in (-\frac{q}{2}, \frac{q}{2})$: If we denote by $\mathbf{u}_i$ the vector such that $\mathbf{u}_i[i] = 1$ and for all $j \neq i$, $\mathbf{u}_i[j] = 0$, we have

$$
|\tilde{\mathbf{e}}[i]| = |\mathbf{u}_i^T \tilde{\mathbf{e}}| = \left| \mathbf{u}_i^T \tilde{\mathbf{R}} \begin{bmatrix} \mathbf{s} - \mathbf{s}' \\ \mathbf{e} - \mathbf{e}' \end{bmatrix} \right| = \left\langle \tilde{\mathbf{R}}^T \mathbf{u}_i, \begin{bmatrix} \mathbf{s} - \mathbf{s}' \\ \mathbf{e} - \mathbf{e}' \end{bmatrix} \right\rangle \tag{5.21}
$$

Using the Cauchy-Schwarz inequality we get:

$$
|\tilde{\mathbf{e}}[i]| \leq \|\tilde{\mathbf{R}}^T \mathbf{u}_i\|_2 \left\| \begin{bmatrix} \mathbf{s} - \mathbf{s}' \\ \mathbf{e} - \mathbf{e}' \end{bmatrix} \right\|_2 \tag{5.22}
$$

Using $\|\mathbf{A}\|_2 = \sigma_{\max}(\tilde{\mathbf{R}}^T)$, and the definition of $\mathcal{X}_g$ (Definition 5.2.7), then Eq. (5.20) and the assumption on $\sigma_{\max}(\mathbf{R})$ we obtain:

$$
|\tilde{\mathbf{e}}[i]| \leq \sigma_{\max}(\tilde{\mathbf{R}^T}) \times (2r_{\max}) < \frac{q}{2} \tag{5.23}
$$

Now, if we write for all $i \in [N]$, $\tilde{\mathbf{s}}[i] = \sum_{j=0}^{k-1} 2^j \tilde{\mathbf{s}}[i]_j$, with for all j, $\tilde{\mathbf{s}}[i]_j \in \{0,1\}$ (we also use the same notation for $\tilde{\mathbf{e}}$), we can prove, following the same path as the `InvertSmallGadget` algorithm that $\forall i, j$, $\tilde{\mathbf{s}}[i]_j = 0$. Let $i \in [n]$. If we consider only the line $l \coloneqq (i-1)k + k$ of $\mathbf{G}\tilde{\mathbf{s}} + \tilde{\mathbf{e}} = \mathbf{0}$, we obtain $2^{k-1}\tilde{\mathbf{s}} + \tilde{\mathbf{e}}[l] = \frac{q}{2}\tilde{\mathbf{s}}[i]_0 + \tilde{\mathbf{e}}[i] \bmod q = \mathbf{0}$. Since $\tilde{\mathbf{e}}[l] \in (-\frac{q}{2}, \frac{q}{2})$, we cannot have $\tilde{\mathbf{s}}[i]_0 = 1$, so $\tilde{\mathbf{s}}[i]_0 = 0$. We can then iterate the same process for $m = 1 \dots k-1$ with the line $l \coloneqq (i-1)k + (k-m)$ to show that $\tilde{\mathbf{s}}[i]_m = 0$, i.e. $\tilde{\mathbf{s}} = 0$, which concludes the proof of the injectivity of $g_{\mathbf{A}}$. Because this proof follows exactly the algorithm `MP.Invert`, it is easy to see using the same argument that this algorithm correctly inverts any $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}_g$.

We prove now the second part of the theorem. Because $\mathbf{R}$ is sampled according to a discrete Gaussian of parameter αq, so according to [MP12, Lem. 2.8], this distribution is 0-subgaussian, and therefore we can apply [MP12, Lem. 2.9] with, for example, $t = \sqrt{N/\pi}$ (we divide by π just to simplify the probability, and to transform the $\leq$ into a $<$). So with probability $1 - 2\exp(-\pi t^2) = 1 - 2e^N$,

$$\sigma_{\max}(\mathbf{R}) \leq C \times \alpha q \times (\sqrt{Nk} + \sqrt{2N} + \sqrt{\frac{N}{\pi}}) < C \times \alpha q \times \sqrt{N}(\sqrt{k} + \sqrt{2} + 1)$$

To conclude the proof, we inject this equation inside Eq. (5.20) and use Eq. (5.27). □

We have now all the background necessary to build our function.

5.3 Function Construction and Analysis

5.3.1 Construction

In this section, we will explain how to derive a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family. See the Section 5.1 to get an intuitive explanation of our method. In the following, `MP.Gen` and `MP.Invert` are the functions defined in [MP12] (to, respectively, generate a couple public key/trapdoor $(\mathbf{A}, \mathbf{R})$ and to invert the function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) \coloneqq \mathbf{A}\mathbf{s} + \mathbf{e}$). Details are in Section 5.2.3.

Definition 5.3.1. *For the parameters $\mathcal{P} \coloneqq (k, N, \alpha, r_{max}, n, \mathcal{X})$ with $(k, N, n, r_{max}) \in \mathbb{N}$, $0 < \alpha < 1$, $\mathcal{X} \subseteq \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n}$ where $q \coloneqq 2^k$ and $M \coloneqq N(1+k)$, we define in Algorithm 3 the algorithms $\mathtt{Gen}_{\mathcal{P}}$, $\mathtt{Invert}_{\mathcal{P}}$, $\mathtt{Eval}_{\mathcal{P}}$ (to compute f_k) and h. We use $\mathbf{d}_0 \in \{0,1\}^n \subseteq \mathbb{Z}_q^n$ (same for $\mathbf{d}$), $\mathbf{s}_0 \in \mathbb{Z}_q^N$, $\mathbf{e}_0 \in \mathbb{Z}_q^M$, $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}$, $c \in \{0,1\}$, $\mathbf{A} \in \mathbb{Z}_q^{(M+n)\times N}$ and $\mathbf{R}$ is the trapdoor obtained via the [MP12] algorithm.*

Algorithm 3 Definition of the δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family

$\underline{\mathtt{Gen}_{\mathcal{P}}(1^{\lambda}, \mathbf{d}_0)}$

1 : $(\mathbf{A}_u, \mathbf{R}) \leftarrow \mathtt{MP.Gen}(1^{\lambda})$
2 : $\mathbf{A}_l \xleftarrow{\$} \mathbb{Z}_q^n$
3 : $\mathbf{A} \coloneqq \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right]$
4 : $\mathbf{s}_0 \leftarrow \mathcal{D}^N_{\mathbb{Z},\alpha q}$
5 : $\mathbf{e}_0 \leftarrow \mathcal{D}^{M+n}_{\mathbb{Z},\alpha q}$
6 : $\mathbf{y}_0 \coloneqq \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0 + \frac{q}{2}\left[\frac{\mathbf{0}^M}{\mathbf{d}_0}\right]$
7 : $k \coloneqq (\mathbf{A}, \mathbf{y}_0)$
8 : $t_k \coloneqq (\mathbf{R}, \mathbf{d}_0, \mathbf{s}_0, \mathbf{e}_0, \mathbf{A})$
9 : **return** (k, t_k)

$\underline{\mathtt{Invert}_{\mathcal{P}}(t_k \coloneqq (\mathbf{R}, \mathbf{d}_0, \mathbf{s}_0, \mathbf{e}_0, \mathbf{A}), \mathbf{y})}$

1 : $\left[\frac{\mathbf{y}_u \in \mathbb{Z}_q^M}{\mathbf{y}_l \in \mathbb{Z}_q^n}\right] \coloneqq \mathbf{y}; \quad \left[\frac{\mathbf{A}_u}{\mathbf{A}_l}\right] \coloneqq \mathbf{A}$
2 : $(\mathbf{s}, \mathbf{e}_u) \leftarrow \mathtt{MP.Invert}(\mathbf{R}, \mathbf{A}, \mathbf{y}_u)$
3 : **if** $\mathbf{s} = \perp$ **then return** $\perp$ **fi**
4 : $\mathbf{d} \coloneqq \mathrm{RoundMod}_q(\mathbf{y}_l - \mathbf{A}_l\mathbf{s})$
5 : $\mathbf{e} \coloneqq \left[\frac{\mathbf{e}_u}{\mathbf{y}_l - \mathbf{A}_l\mathbf{s} - \mathbf{d}}\right]$
6 : $\mathbf{s}' \coloneqq \mathbf{s} - \mathbf{s}_0; \quad \mathbf{e}' \coloneqq \mathbf{e} - \mathbf{s}_0$
7 : **if** $(\mathbf{s}, \mathbf{e}) \notin \mathcal{X}$ or $(\mathbf{s}', \mathbf{e}') \notin \mathcal{X}$ **then**
8 : **return** $\perp$ **fi**
9 : **return** $((\mathbf{s}, \mathbf{e}, 0, \mathbf{d}), (\mathbf{s}', \mathbf{e}', 1, \mathbf{d} \oplus \mathbf{d}_0))$

$\underline{\mathtt{Eval}_{\mathcal{P}}(k \coloneqq (\mathbf{A}, \mathbf{y}_0), x \coloneqq (\mathbf{s}, \mathbf{e}, c, \mathbf{d})) =: f_k(x)}$

1 : **return** $\mathbf{A}\mathbf{s} + \mathbf{e} + \left[\frac{\mathbf{0}^M}{\mathbf{d}}\right] + c \times \mathbf{y_0}$

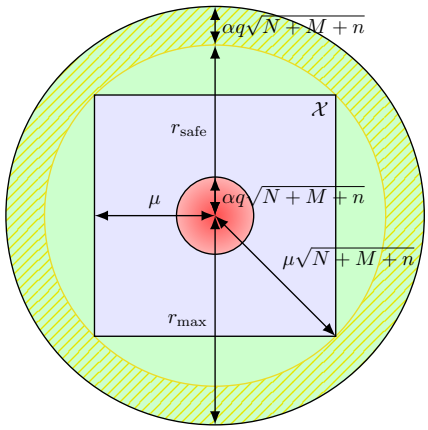
$\underline{h(x \coloneqq (\mathbf{s}, \mathbf{e}, c, \mathbf{d}))}$

1 : **return** $\mathbf{d}$

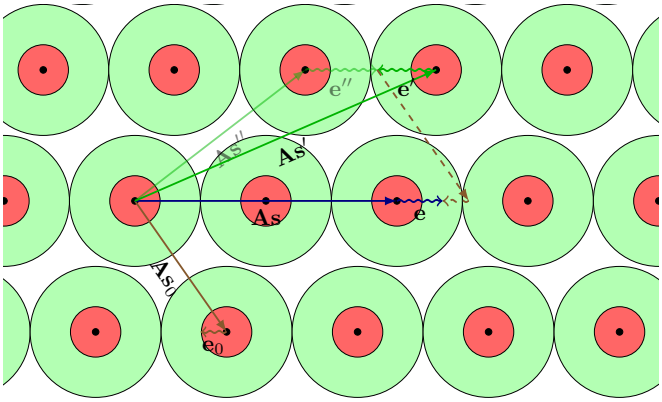
5.3.2 Analysis

In the rest of the section, we will prove that we can find appropriate set of parameters to implement this construction and obtain a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family.

We derive now conditions to check that a function is δ-2-to-1 and has the XOR property explained in Definition 4.2.1. Note that we do a worse case analysis to compute δ (so in average δ may be smaller than what we actually compute) in order to also deal with an adversarial scenario in which f_k is maliciously sampled: while this does not make a lot of sense right now, it will turn out to be useful in Chapter 7 when defining NIZKoQS to define a function `H.CheckTrapdoor`. Note that you can refer to Figure 5.3a in order to have a graphical representation of the different parameters used in the rest of this section (the need for r_{safe} is depicted in Figure 5.3b).



(a) Graphical representation of the parameters. The vector $(\mathbf{s}_0, \mathbf{e}_0)$ is sampled according to a very small Gaussian $\mathcal{D}^{N+M+n}_{\mathbb{Z},\alpha q}$ and its norm is with overwhelming probability smaller than $\alpha q\sqrt{N+M+n}$ (smaller red circle). The bigger green circle of radius r_{max} corresponds to the space in which $g_{\mathbf{A}}$ can be decrypted. The non-hatched part, delimited by the circle of radius r_{safe}, delimits the space in which the two preimages have the property that $s' = s - s_0$ and $e' = e - e_0$. The set in which we define our input is $\mathcal{X} = \mathcal{X}_{\blacksquare\mu}$, the hypercube of "radius" μ (it is easy to create a uniform superposition on a hypercube).



(b) Illustration of the issue if we use r_{max} instead of r_{safe}: the true preimage (s', e') may not be such that $s' = s - s_0$ and $e' = e - e_0$ (here this "wanted" preimage is denoted (s'', e'')), which is required for the correctness.

Figure 5.3: Graphical representation of the parameters and of the potential issues when using r_{max} in place of r_{safe}.

Lemma 5.3.2 (Conditions for f_k to be δ-2-to-1)**.** *Let $\mathcal{P}$ be like in Definition 5.3.1. We define $r_{\text{safe}} = r_{\text{max}} - \alpha q\sqrt{N+M+n}$, $\mathcal{X} + (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0) \coloneqq \{(\mathbf{s}+\hat{\mathbf{s}}_0, \mathbf{e}+\hat{\mathbf{e}}_0) \mid (\mathbf{s},\mathbf{e}) \in \mathcal{X}\}$ and*

$$\delta \coloneqq 1 - \min\left\{ \frac{|\mathcal{X} \cap (\mathcal{X} + (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0))|}{|\mathcal{X}|} \;\middle|\; (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n}, \left\| \begin{bmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_2 \leq \alpha q\sqrt{N+M+n} \right\} \tag{5.24}$$

Let $\mathbf{s}_0 \in \mathbb{Z}_q^N$, $\mathbf{e}_0 \in \mathbb{Z}_q^{M+n}$, $\mathbf{d}_0 \in \{0,1\}$, $\hat{\mathbf{A}} \in \mathbb{Z}_q^{N\times N}$, $\mathbf{A}_l \in \mathbb{Z}_q^{n\times n}$ and $\mathbf{R} = \left[\mathbf{R}_1 \;\middle|\; \mathbf{R}_2\right] \in \mathbb{Z}_q^{Nk\times 2N}$. We define as before:

$$\mathbf{A_u} \coloneqq \left[\begin{array}{c} \hat{\mathbf{A}} \\ \hline \mathbf{G} - \mathbf{R}_2\hat{\mathbf{A}} - \mathbf{R}_1 \end{array}\right] \qquad \mathbf{A} \coloneqq \left[\begin{array}{c} \mathbf{A}_u \\ \hline \mathbf{A}_l \end{array}\right] \in \mathbb{Z}_q^{(M+n)\times N} \tag{5.25}$$

together with $\mathbf{y}_0 \coloneqq \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0 + \frac{q}{2}\left[\begin{array}{c} \mathbf{0}^M \\ \hline \mathbf{d}_0 \end{array}\right]$ and $k \coloneqq (\mathbf{A}, \mathbf{y}_0)$.

If $\sqrt{\sigma_{\max}(\mathbf{R})^2+1} \leq \frac{q}{4r_{\max}}$, $\left\| \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{e}_0 \end{bmatrix} \right\|_2 \leq \alpha q \sqrt{N+M+n}$, *and*

$$\mathcal{X} \subseteq \left\{ (\mathbf{s},\mathbf{e}) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n} \;\middle|\; \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_2 \leq r_{\text{safe}} \right\} \tag{5.26}$$

then the function $f_k(x)$ *described in Definition 5.3.1 is* δ*-2-to-1, trapdoor, and for any* y *having exactly two preimages* x *and* x'*, we have* $x \neq x'$*,* $h(x) \oplus h(x') = \mathbf{d}_0$*.*

On the other hand, if (k, t_k) *is sampled according to* $\texttt{Gen}(1^\lambda, \mathbf{d}_0)$*, if Eq. (5.26) is true, and if:*

$$\sqrt{\left(C \times \alpha q \times \sqrt{N}(\sqrt{k}+\sqrt{2}+1)\right)^2+1} \leq \frac{q}{4r_{\max}} \tag{5.27}$$

then with overwhelming probability on N*, the function* f_k *is* δ*-2-to-1, trapdoor, and for any* y *having exactly two preimages* x *and* x'*, we have* $x \neq x'$*,* $h(x) \oplus h(x') = \mathbf{d}_0$*.*

Proof. Let us first prove that for all $c \in \{0,1\}$, the function $f_k(\cdot,\cdot,c,\cdot)$ is injective. Let $c \in \{0,1\}$, and $\mathbf{s},\mathbf{e},\mathbf{d},\mathbf{s}',\mathbf{e}',\mathbf{d}'$ be such that $f_k(\mathbf{s},\mathbf{e},c,\mathbf{d}) = f_k(\mathbf{s}',\mathbf{e}',c,\mathbf{d}')$. Then, if we consider only the upper part of this equation (and denote $\mathbf{e}_u$ the upper part of the vector $\mathbf{e}$), we get $\mathbf{A}_u\mathbf{s}+\mathbf{e}_u+c\times\mathbf{y}_{0,l} = \mathbf{A}_u\mathbf{s}'+\mathbf{e}'_u+c\times\mathbf{y}_{0,l}$, i.e. $\mathbf{A}_u\mathbf{s}+\mathbf{e}_u = \mathbf{A}_u\mathbf{s}'+\mathbf{e}'_u$. But because $\sqrt{\sigma_{\max}(\mathbf{R})^2+1} \leq \frac{q}{4r_{\max}}$ and due to the condition on $\mathcal{X}$ given in Eq. (5.26) and the fact that $r_{\text{safe}} \leq r_{\max}$, according to Lemma 5.2.10 the function $(\mathbf{s},\mathbf{e}) \to \mathbf{A}_u\mathbf{s}+\mathbf{e}$ is injective. So $\mathbf{s}=\mathbf{s}'$ and $\mathbf{e}_u = \mathbf{e}'_u$. Now, we focus on the lower part of the above equation: we have $\mathbf{A}_l\mathbf{s}+\mathbf{e}_l+\frac{q}{2}\mathbf{d}+c\times\mathbf{y}_{0,l} = \mathbf{A}_l\mathbf{s}'+\mathbf{e}'_l+\frac{q}{2}\mathbf{d}'+c\times\mathbf{y}_{0,l}$. Because $\mathbf{s}=\mathbf{s}'$, we obtain $\mathbf{e}_l+\frac{q}{2}\mathbf{d} = \mathbf{e}'_l+\frac{q}{2}\mathbf{d}'$. Because $1 \leq \sqrt{\sigma_{\max}(\mathbf{R})+1} < \frac{q}{4r_{\max}}$, we have $r_{\max} < \frac{q}{4}$. Therefore, we get for all i, $|\mathbf{e}_l[i]| < \frac{q}{4}$, so $\text{RoundMod}_q(\mathbf{e}_l[i]+\frac{q}{2}\mathbf{d}[i]) = \text{RoundMod}_q(\mathbf{e}'_l[i]+\frac{q}{2}\mathbf{d}'[i])$, i.e. $\mathbf{d}[i] = \mathbf{d}'[i]$. So $\mathbf{d}=\mathbf{d}$ and therefore we also get $\mathbf{e}_l = \mathbf{e}'_l$: the function $f_k(\cdot,\cdot,c,\cdot)$ is injective.

Therefore, f_k has at most two preimages, one for $c=0$ and one for $c=1$. We prove now that $f_k(\mathbf{s},\mathbf{e},0,\mathbf{d}) = f_k(\mathbf{s}',\mathbf{e}',1,\mathbf{d}')$, iff $(\mathbf{s}',\mathbf{e}',\mathbf{d}') = (\mathbf{s}-\mathbf{s}_0,\mathbf{e}-\mathbf{e}_0,\mathbf{d}\oplus\mathbf{d}_0)$. One implication is trivial: if $(\mathbf{s}',\mathbf{e}',\mathbf{d}') = (\mathbf{s}-\mathbf{s}_0,\mathbf{e}-\mathbf{e}_0,\mathbf{d}\oplus\mathbf{d}_0)$ then because q is even, $f_k(\mathbf{s},\mathbf{e},0,\mathbf{d}) = f_k(\mathbf{s}',\mathbf{e}',1,\mathbf{d}')$. We prove now the second implication. By definition of f_k, if we consider again the upper part of the equation and replace y_0 by its definition, we have $\mathbf{A}_u\mathbf{s}+\mathbf{e}_u = \mathbf{A}_u(\mathbf{s}'+\mathbf{s}_0)+(\mathbf{e}'_u+\mathbf{e}_{0,u})$. But the triangle inequality gives:

$$\left\| \begin{bmatrix} \mathbf{s}'+\mathbf{s}_0 \\ \mathbf{e}'_u+\mathbf{e}_{0,u} \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} \mathbf{s}' \\ \mathbf{e}'_u \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{e}_{0,u} \end{bmatrix} \right\|_2 \leq r_{safe} + \alpha q \sqrt{N+M+n} = r_{\max}$$

Therefore, we can use again the injectivity property given in Lemma 5.2.10, which gives $(\mathbf{s}, \mathbf{e}_u) = (\mathbf{s}' + \mathbf{s}_0, \mathbf{e}' + \mathbf{e}_{0,u})$. We can now analyse the lower part of the equation: $\mathbf{A}_l\mathbf{s} + \mathbf{e}_l + \frac{q}{2}\mathbf{d} = \mathbf{A}_l(\mathbf{s}' + \mathbf{s}_0) + (\mathbf{e}'_l + \mathbf{e}_{0,l}) + \frac{q}{2}(\mathbf{d}' + \mathbf{d}_0)$. Because $\mathbf{s} = \mathbf{s}' + \mathbf{s}_0$ and q is even, we have $\mathbf{e}_l + \frac{q}{2}\mathbf{d} = (\mathbf{e}'_l + \mathbf{e}_{0,l}) + \frac{q}{2}(\mathbf{d}' \oplus \mathbf{d}_0)$. Using again the triangle inequality, we prove the same way that $\|\mathbf{e}'_l + \mathbf{e}_{0,l}\|_2 < \frac{q}{2}$. As before, by rounding the previous equation using $\mathrm{RoundMod}_q$, we obtain $\mathbf{d} = \mathbf{d}' \oplus \mathbf{d}_0$ and $\mathbf{e}_l = \mathbf{e}'_l$ which concludes the proof. In particular, if x and x' are the two preimage, we have $h(x) \oplus h(x') = \mathbf{d} \oplus \mathbf{d}' = \mathbf{d}_0$. We remark that this proof follows exactly the algorithm `Invert`, therefore the correctness of `Invert` follows quite directly and thus the function is trapdoor.

Now, we prove that the function f_k is δ-2-to-1. The total number of elements in the domain of f_k is $2|\mathcal{X}| \times 2^n$. Let $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}$ and $\mathbf{d} \in \{0,1\}^n$. Then, using the result proven above, $f_k(\mathbf{s}, \mathbf{e}, 0, \mathbf{d})$ has a second preimages $(\mathbf{s}-\mathbf{s}_0, \mathbf{e}-\mathbf{e}_0, 1, \mathbf{d}\oplus\mathbf{d}_0)$ iff $(\mathbf{s}-\mathbf{s}_0, \mathbf{e}-\mathbf{e}_0) \in \mathcal{X}$, i.e. iff $(\mathbf{s}, \mathbf{e}) \in \mathcal{X} + (\mathbf{s}_0, \mathbf{e}_0)$. So the number of elements x such that $|f_k^{-1}(f_k(x))| = 2$ is equal to $2|\mathcal{X} \cap (\mathcal{X} + (\mathbf{s}_0, \mathbf{e}_0))|2^n$, therefore if we define:

$$\delta_k \coloneqq 1 - \frac{2|\mathcal{X} \cap (\mathcal{X} + (\mathbf{s}_0, \mathbf{e}_0))|2^n}{2|\mathcal{X}|2^n} = 1 - \frac{|\mathcal{X} \cap (\mathcal{X} + (\mathbf{s}_0, \mathbf{e}_0))|}{|\mathcal{X}|} \tag{5.28}$$

this function is δ_k-2-to-1. But $\left\| \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{e}_0 \end{bmatrix} \right\|_2 \leq \alpha q \sqrt{N + M + n}$, so by definition of δ, $\delta_k \leq \delta$. So f_k is also δ-2-to-1.

To prove the last part of the theorem, we use Eq. (5.27) and Lemma 5.2.10: with overwhelming probability (on N), $\sqrt{\sigma_{\max}(\mathbf{R})^2 + 1} < \frac{q}{4r_{\max}}$. Moreover, because $\begin{bmatrix} \mathbf{s}_0 \\ \mathbf{e}_0 \end{bmatrix}$ is sampled according to $\mathcal{D}_{\mathbb{Z},\alpha q}^{N+(M+n)}$, we get, using[6] Lemma 5.2.3, that with overwhelming probability (on $N + M + n$): $\left\| \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{e}_0 \end{bmatrix} \right\|_2 \leq \alpha q \sqrt{N + M + n}$. We can now end the proof using the first (already proven) part of the theorem. □

Note that we did not yet give an explicit definition of $\mathcal{X}$. The most natural way to define $\mathcal{X}$ may be to define it following Eq. (5.26) as:

$$\mathcal{X}_\bullet \coloneqq \left\{ (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n} \;\middle|\; \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_2 \leq r_{\text{safe}} \right\} \tag{5.29}$$

However, for our protocol to work, one needs to be able to create quantumly a uniform superposition over all elements in $\mathcal{X}$. A first naive method would be a *rejection sampling*

[6] Note that the original lemma applies to Gaussian distributions that are not reduced modulo q, but reducing the Gaussian distribution modulo q can only decrease the length of the vector.

method (see Remark 4.2.2) by creating a uniform superposition on the hypercube of length $2r_{safe}$ (see later how to do) and rejecting if the vectors have length bigger than r_{safe}. Unfortunately, this method is inefficient as the probability of not rejecting is negligible when N tends to the infinity. While some more efficient methods may exist, it will be easier to focus rather on $\mathcal{X}$ being an hypercube.

Definition 5.3.3. *For any $(\mu, N, M, n) \in \mathbb{N}^4$, we define the hypercube $\mathcal{X}_{\blacksquare\mu}$ as:*

$$\mathcal{X}_{\blacksquare\mu} := \left\{ (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n} \;\middle|\; \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_\infty \leq \mu \right\} \tag{5.30}$$

Remark 5.3.4. It is now easy to sample from $\mathcal{X}_{\blacksquare\mu}$: we can do a rejection sampling as explained above, except that we proceed coordinate per coordinate (this is much more efficient than doing a rejection sampling on the final high dimensional state): if we use the binary two's complement notation, we apply Hadamard gates on $\lceil \log_2(2\mu+1) \rceil$ qubits, and use an auxiliary qubit to check if the state is projected on the superposition of elements having size $\leq 2\mu+1$ (this should happen with probability $\frac{2\mu+1}{2^{\lceil \log_2(2\mu+1) \rceil}} \geq 1/2$). If the test passes, we add $|0\rangle$ "significants qubits" until having $k = \log_2(q)$ qubits, and run the quantum unitary that substracts μ modulo q. We repeat until having $N+M+n$ successful projections (this require therefore $O(N+M+n)$ samplings). Moreover, we can even get completely rid of the rejection sampling if we slightly change the definition of $\mathcal{X}_{\blacksquare\mu}$ and if we make it less symmetric by asking that there exists $k' \in \mathbb{N}$ such that for all i, $\begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix}[i] \in \left[-2^{k'}, 2^{k'} - 1\right]$. The superposition procedure is the same except that we work on $k'+1$ qubits, and we do not need the rejection sampling. However this notation slightly complicates the computations with no clear benefit (if simplifies slightly the sampling part, but it may decrease the value of δ since the term $2^{k'}$ must be a power of 2), so for simplicity we will keep our initial notation. Note also that in the following, for the sake of simplicity we won't try to give tight bounds.

The following lemma will be useful to bound δ when $\mathcal{X} = \mathcal{X}_{\blacksquare\mu}$.

Lemma 5.3.5. *Let $(N, M, n, \mu) \in \mathbb{N}^4$, $\mathcal{X} = \mathcal{X}_{\blacksquare\mu}$, $\alpha \in (0,1)$, δ be as in Eq. (5.24) and $\mu' := \left\lfloor \mu - \alpha q \sqrt{N+M+n} \right\rfloor$. Then if $\mu' \geq 0$:*

$$\delta \leq 1 - \left(\frac{2\mu'+1}{2\mu+1} \right)^{N+M+n} \leq \frac{(\alpha q + 1)(N+M+n)^{3/2}}{\mu + 1/2} \tag{5.31}$$

Proof. Let $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}_{\blacksquare_\mu}$ and $(\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0) \in \mathbb{Z}_q^N \times \mathbb{Z}_q^{M+n}$ such that $\left\| \begin{bmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_2 \leq \alpha q \sqrt{N+M+n}$. Then, $(\mathbf{s}, \mathbf{e}) \in \mathcal{X}_{\blacksquare_\mu} + (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0)$ iff $(\mathbf{s} - \hat{\mathbf{s}}_0, \mathbf{e} - \hat{\mathbf{e}}_0) \in \mathcal{X}$, i.e. iff $\left\| \begin{bmatrix} \mathbf{s} - \hat{\mathbf{s}}_0 \\ \mathbf{e} - \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_\infty \leq \mu$. If we assume that $\left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_\infty \leq \mu'$ (this will not be super tight, but it is good enough for our analysis), then

$$\left\| \begin{bmatrix} \mathbf{s} - \hat{\mathbf{s}}_0 \\ \mathbf{e} - \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_\infty \leq \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_\infty + \left\| \begin{bmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_\infty \leq \left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} \right\|_\infty + \left\| \begin{bmatrix} \hat{\mathbf{s}}_0 \\ \hat{\mathbf{e}}_0 \end{bmatrix} \right\|_2 \leq \mu \tag{5.32}$$

Therefore, $\mathcal{X}_{\blacksquare_{\mu'}} \subseteq \mathcal{X}_{\blacksquare_\mu} \cap (\mathcal{X}_{\blacksquare_\mu} + (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0))$. So $|\mathcal{X}_{\blacksquare_\mu} \cap (\mathcal{X}_{\blacksquare_\mu} + (\hat{\mathbf{s}}_0, \hat{\mathbf{e}}_0))| \geq |\mathcal{X}_{\blacksquare_{\mu'}}| = (2\mu' + 1)^{N+M+n}$, and because $|\mathcal{X}_{\blacksquare_\mu} = (2\mu+1)^{N+M+n}|$, we get:

$$\begin{aligned}
\delta &\leq 1 - \left(\frac{2\mu'+1}{2\mu+1}\right)^{N+M+n} = 1 - \left(1 + \left(-1 + \frac{2\mu'+1}{2\mu+1}\right)\right)^{N+M+n} \\
&\leq 1 - \left(1 + (N+M+n)\left(-1 + \frac{2\mu'+1}{2\mu+1}\right)\right) && \text{(Bernouilli's inequality)} \\
&= (N+M+n)\left(1 - \frac{2\mu'+1}{2\mu+1}\right) \\
&\leq (N+M+n)\left(1 - \frac{2\mu - 2\alpha q\sqrt{N+M+n} + 3}{2\mu+1}\right) && \text{(Remove } \lfloor\cdot\rfloor\text{)} \\
&= (N+M+n)\left(\frac{\alpha q\sqrt{N+M+n} + 1}{\mu + 1/2}\right) \\
&\leq \frac{(\alpha q + 1)(N+M+n)^{3/2}}{\mu + 1/2} && \square
\end{aligned}$$

We can derive now a more simple set of conditions on the parameters ensuring that the function created from these parameters is $\delta_m - \mathsf{GHZ}^{\mathsf{H}}$*capable*.

Lemma 5.3.6 (Conditions on parameters)**.** *Let $\lambda \in \mathbb{N}$ be a security parameter, let $(k, n) \in \mathbb{N}$ and $\alpha \in (0, 1)$ be parameters that depend on λ, and $C \approx \frac{1}{\sqrt{2\pi}}$ (see Lemma 5.2.10). We define $N \coloneqq \lambda$, $q = 2^k$, $M \coloneqq N(1+k)$,*

$$r_{\max} \coloneqq \frac{q}{4\sqrt{\left(C \times \alpha q \times \sqrt{N}(\sqrt{k} + \sqrt{2} + 1)\right)^2 + 1}} \tag{5.33}$$

$$r_{\text{safe}} \coloneqq r_{\max} - \alpha q \sqrt{N+M+n} \tag{5.34}$$

$$\mu \coloneqq \left\lfloor \frac{r_{\text{safe}}}{\sqrt{N+M+n}} \right\rfloor \tag{5.35}$$

$$\mathcal{X} \coloneqq \mathcal{X}_{\blacksquare\mu} \tag{5.36}$$

$$\delta_m \coloneqq \frac{(\alpha q+1)(N+M+n)^{3/2}}{\mu+1/2} \tag{5.37}$$

Then, if $\left\lfloor \mu - \alpha q\sqrt{N+M+n} \right\rfloor \geq 0$*, the construction given in Definition 5.3.1 is* δ_m $-$ $\mathsf{GHZ}^{\mathsf{H}}$*capable (see Definition 4.2.1) assuming the security of decision-*$\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},\alpha q}}$*. Moreover, if we define:*

$$\alpha_0 \coloneqq \frac{1}{q}\sqrt{(\alpha q)^2 - \omega(\sqrt{\log N})^2} \tag{5.38}$$

$$\gamma \coloneqq \tilde{O}\left(\frac{N}{\alpha_0}\right) \qquad \text{(See constants in [PRS17])}$$

and if $\alpha_0 q > 2\sqrt{N}$*, then the construction is secure if* GapSVP_γ *is hard. In particular, we are interesting in the regime in which* δ_m *is negligible (correctness) and in which* $\gamma = \tilde{O}(2^{N^\varepsilon})$ *for some* $\varepsilon \in (0, 1/2)$ *(security).*

Proof. For the first part of the theorem, the efficient generation and computation properties are trivial to check. The fact that the function is trapdoor and the property on the XOR is a direct consequence of Lemma 5.3.2. The δ_m-2-to-1 property comes from Lemma 5.3.2 and Lemma 5.3.5. The method to efficiently create a uniform superposition of elements in $\mathcal{X}$ is given in Remark 5.3.4.

To prove the indistinguishability property on game IND-DO, we assume that there exists an adversary $\mathcal{A}$ that can win this game with a non-negligible advantage. Because $\mathcal{A}$ has only access to $\left(\mathbf{A}, \mathbf{y}_0 = \mathbf{A}\mathbf{s}_0 + \mathbf{e}_0 + \frac{q}{2}\begin{bmatrix}\mathbf{0}^M \\ \mathbf{d}_0^{(c)}\end{bmatrix}\right)$, we can use $\mathcal{A}$ to break the decision-LWE problem: given a challenge $(\mathbf{A}', \mathbf{y})$, we run the adversary and send to $\mathcal{A}$ the couple $(\mathbf{A}', \mathbf{y} + \frac{q}{2}\begin{bmatrix}\mathbf{0}^M \\ \mathbf{d}_0^{(c)}\end{bmatrix})$. If the guess $\tilde{c}$ of $\mathcal{A}$ equals c, we guess that we get the non-uniform distribution normal-$A_{\mathbf{s},\mathcal{D}_{\mathbb{Z},\alpha q}}$ (i.e. the distribution where $\mathbf{s}$ is also sampled according to $\mathcal{D}_{\mathbb{Z},\alpha q}$), otherwise we guess that we get the uniform distribution U. We remark that if $\mathbf{y}$ is a vector chosen uniformly at random, then the distribution of $\mathbf{y} + \frac{q}{2}\begin{bmatrix}\mathbf{0}^M \\ \mathbf{d}_0^{(c)}\end{bmatrix}$ is statistically uniform. So in that case, $\mathcal{A}$ cannot guess c with probability better than $1/2$. Now, if y is sampled from normal-$A_{\mathbf{s},\mathcal{D}_{\mathbb{Z},\alpha q}}$, then $\mathcal{A}$ must guess correctly the value of c with non-negligible advantage, otherwise it means that we can distinguish a matrix obtained by `MP.Gen` from a uniform matrix (we already know it is not possible, see Lemma 5.2.8). Therefore, the probability p of guessing the correct distribution is:

$$p \geq \frac{1}{2} \times \frac{1}{2} + \frac{1}{2}\left(\frac{1}{2} + \mathsf{negl}(\lambda)\right) = \frac{1}{2} + \mathsf{negl}(\lambda) \tag{5.39}$$

which is absurd since we assumed the hardness of LWE. Note that we do not exactly have an instance of decision-LWE, because it is an instance of the normal version of decision-LWE ($\mathbf{s}_0$ is sampled according to the same distribution as $\mathbf{e}_0$). However, Lemma 5.2.6 shows that the normal problem is harder, and keeping only K samples can also only make the problem harder. Therefore, no adversary can win IND-DO with non-negligible advantage.

For the second part, if we assume GapSVP_γ to be hard, then using Lemma 5.2.4 we get that decision-$\mathsf{LWE}_{q,\mathcal{D}_{\alpha_0 q}}$ is also hard ($\alpha_0 q > 2\sqrt{N}$, and $0 < \alpha_0 \leq \alpha < 1$). We can now discretize the distribution using Corollary 5.2.5 ($\lambda = N$) to obtain that decision-$\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},\alpha q}}$ is hard. Indeed, if we assume the existence of an adversary $\mathcal{A}$ that can distinguish with non-negligible advantage the distribution U from $A_{\mathbf{s},\mathcal{D}_{\mathbb{Z},\alpha q}}$ for a vector $\mathbf{s}$ chosen uniformly at random, then $\mathcal{A}$ has also a non-negligible advantage in distinguishing $\mathcal{A}_{\mathbf{s},\chi}$ where χ is the marginal distribution of $\mathbf{e}$ in Corollary 5.2.5 (otherwise we could use $\mathcal{A}$ to distinguish χ from $\mathcal{D}_{\mathbb{Z},\alpha q}$, which is impossible because they are statistically negligibly close). But it also means that $\mathcal{A}$ can also be used to break $\mathsf{LWE}_{q,\mathcal{D}_{\alpha_0 q}}$ by first discretizing $\mathcal{D}_{\alpha_0 q}$ (it works because the transformation given in Corollary 5.2.5 also maps the uniform distribution on itself). Since we already prove the security when we assume the hardness of decision-$\mathsf{LWE}_{q,\mathcal{D}_{\mathbb{Z},\alpha q}}$ above, the proof is finished. □

5.3.3 Explicit Instantiation of the Parameters

We prove now that there exists an instantiation that fulfills the requirements of Lemma 5.3.6. Note that for simplicity, we only verify the properties asymptotically. Moreover, we do not attempt to give any particularly optimized construction (note that there is are tradeoffs between security, correctness, efficiency, and simplicity of the quantum superposition preparation circuit).

Theorem 5.3.7 (Existence of a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family)**.** *Let $\varepsilon \in (0, \frac{1}{2})$ be a constant, and $\lambda \in \mathbb{N}$ be a security parameter. Let $n = \mathsf{poly}(\lambda) \in \mathbb{N}$ and $N := \lambda$. If we assume the hardness of the GapSVP_γ problem for any $\gamma = \tilde{O}(2^{N^\varepsilon})$, then there exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions. More precisely, if we define the fixed function $\omega(\sqrt{\log N}) := \log N$, $k := \lfloor N^\varepsilon \rfloor$, $q := 2^k$,*

$$\alpha := \frac{\sqrt{4N + \omega(\sqrt{\log N})^2 + 1}}{q} \tag{5.40}$$

and $M, r_{\max}, r_{\text{safe}}, \mu, \mathcal{X}, \delta_m$ *as in Lemma 5.3.6, the construction given in Definition 5.3.1 is* δ_m-$\mathsf{GHZ}^{\mathsf{H}}$ *capable (for sufficiently large*[7] λ*), with* $\delta_m = \mathsf{negl}(\lambda)$.

Proof. We just need to check that for sufficiently large λ the properties of Lemma 5.3.6 are respected. Because $1/q = \mathsf{negl}(N)$, it is easy to see that for sufficiently large λ, $a \in (0,1)$ since $\alpha = \mathsf{poly}(N)/q$. Then, $\alpha q > \sqrt{4N + \omega(\sqrt{\log N})^2}$, so using notation from Lemma 5.3.6, we directly get $\alpha_0 q > 2\sqrt{N}$. Moreover, multiplying Eq. (5.40) by q, we get $\alpha q = \mathsf{poly}(N)$, so it means that $\alpha_0 = \mathsf{poly}(N)/q = \mathsf{negl}(\lambda)$. Therefore $\gamma = \tilde{O}(N/\alpha_0) = \tilde{O}(\sqrt{N}q) = \tilde{O}(\sqrt{N}2^{N^\varepsilon}) = \tilde{O}(2^{N^\varepsilon})$ which is assumed to be hard. Next, let us study μ and δ_m. Because $\alpha q = \mathsf{poly}(n)$ and $q = 1/\mathsf{negl}(\lambda)$ is superpolynomial, that $r_{\max}$ is also superpolynomial, and same for r_{safe}, μ and $\lfloor \mu - \alpha q\sqrt{N+M+N} \rfloor$ (we only subtract or divide by terms that are $\mathsf{poly}(N)$). Therefore, for a large enough λ $(= N)$, we have $\lfloor \mu - \alpha q\sqrt{N+M+N} \rfloor \geq 0$. Finally, $\delta_m = \mathsf{poly}(N)/(\mu + 1/2)$. But we showed that μ is superpolynomial, so $\frac{1}{\mu}$ is negligible, and therefore δ_m is also negligible, which ends the proof. □

We also derive a similar statement where we only rely on LWE with polynomial noise ratio. In that case, it is possible to get δ arbitrarily small, but it decreases only polynomially fast with the security parameter λ.

Remark 5.3.8. There are two possible reasons to base our security on LWE with polynomial noise ratio: first, this assumption is more standard than the hardness of LWE with superpolynomial noise ratio. Secondly, it allows us to store the modulus q on a logarithmic (in λ) number of bits instead of on a polynomial number of bits, which should allow us to implement quantumly f_k more efficiently. On the other hand, the value of δ decreases, and we need to run a different protocol to generate $|+_\theta\rangle$ with overwhelming probabilities, involving multiple quantum evaluation of f_k (the function is therefore simpler to evaluate, but must be evaluated more).

Theorem 5.3.9 (Existence of a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable based on LWE with polynomial noise ratio)**.** *Let* $\lambda \in \mathbb{N}$ *be a security parameter. Let* $n = O(\lambda) \in \mathbb{N}$ *and* $N := \lambda$*. If we assume the hardness of the* GapSVP_γ *problem for* $\gamma = \mathsf{poly}(N)$*, then there exists a* δ-$\mathsf{GHZ}^{\mathsf{H}}$ *capable*

[7] The function may not be well defined for too small λ because the input set may be empty. For example, with this instantiation, if we take $\varepsilon = \frac{1}{3}$, the function is well defined for $N \geq 7 \times 10^5$. Moreover, when $N = 6 \times 10^6$, we get $k = 181$ and $\delta_m < 2^{-80}$. There is surely place for optimisation, but only existence matters here. Also, it is possible to change slightly the definition of k by adding a constant term (it won't change the analysis at all) to allow f_k to be defined for smaller values.

family of functions with $\delta = \tilde{O}\left(\frac{1}{\sqrt{\lambda}}\right) \xrightarrow{\infty} 0$. *More precisely, if we define the fixed function*[8] $\omega(\sqrt{\log N}) := \log N$, $k := 20 + 4\lceil \log N \rceil$, $q := 2^k$,

$$\alpha := \frac{\sqrt{4N + \omega(\sqrt{\log N})^2 + 1}}{q} \tag{5.41}$$

and $M, r_{\max}, r_{\text{safe}}, \mu, \mathcal{X}, \delta_m$ *as in Lemma 5.3.6, the construction given in Definition 5.3.1 is* δ-$\mathsf{GHZ}^{\mathsf{H}}$ *capable (for sufficiently large* λ*), with* $\delta = \tilde{O}\left(\frac{1}{\sqrt{\lambda}}\right) \xrightarrow{\infty} 0$.

Proof. As before, we need to check that for sufficiently large λ the properties of Lemma 5.3.6 are respected. First, we can check that $\alpha_0 q \geq 2\sqrt{N}$: this is a direct consequence of the definition of α_0 and α since

$$\alpha_0 q \overset{(5.38)}{=} \sqrt{(\alpha q)^2 - \omega(\sqrt{\log N})^2} \overset{(5.41)}{=} \sqrt{4N + \omega(\sqrt{\log N})^2 + 1 + \omega(\sqrt{\log N})^2} \tag{5.42}$$

$$= \sqrt{4N+1} > 2\sqrt{N} \tag{5.43}$$

Then, we approximate some terms using the notation $f(x) = \tilde{\Theta}(g(x))$ to denote the fact that there exist $k > 0$ such that:

$$\lim_{x\to\infty} \frac{f(x)}{g(x)\log^k x} < \infty \qquad \text{and} \qquad \lim_{x\to\infty} \frac{f(x)\log^k x}{g(x)} > 0 \tag{5.44}$$

In particular, if $f(x) = f_0(x) \pm f_1(x)$ where $f_0(x) = \tilde{\Theta}(x^a)$, $f_1(x) = \tilde{\Theta}(x^b)$ and $a > b$, then $f = \tilde{\Theta}(x^a)$.

This simple fact is trivial to prove. We just define $k = \max(k_0, k_1)$ where k_0 and k_1 are the term k appearing in the $\tilde{\Theta}$ definition of both terms. Then

$$\lim_{x\to\infty} \frac{f(x)}{x^a \log^k x} = \lim_{x\to\infty} \frac{f_0(x) \pm f_1(x)}{x^a \log^k x} = \lim_{x\to\infty} \frac{f_0(x)}{x^a \log^k x} \pm \lim_{x\to\infty} \frac{f_1(x)}{x^a \log^k x} \tag{5.45}$$

$$\leq \lim_{x\to\infty} \frac{f_0(x)}{x^a \log^{k_0} x} \pm \lim_{x\to\infty} \frac{f_1(x)}{x^b \log^{k_1} x} < \infty \tag{5.46}$$

similarly,

$$\lim_{x\to\infty} \frac{f(x)\log^k x}{x^a} = \lim_{x\to\infty} \frac{(f_0(x) \pm f_1(x))\log^k x}{x^a} = \lim_{x\to\infty} \frac{f_0(x)\log^k x}{x^a} \pm \lim_{x\to\infty} \frac{f_1(x)\log^k x}{x^a} \tag{5.47}$$

$$> \lim_{x\to\infty} \frac{f_0(x)\log^{k_1} x}{x^a} \pm \lim_{x\to\infty} \frac{1}{x^{a-b}} \frac{f_1(x)\log^{k_2} x}{x^b} \tag{5.48}$$

Because $a > b$ and using the assumption, the last term tends to zero and the first term is greater than 0 so $\lim_{x\to\infty} \frac{f(x)\log^k x}{x^a} > 0$.

[8] In the expression of k, the 20 plays no particular role and could be removed or changed, it only allows us to evaluate the function for small values of λ. The 4 on the other hand is important to ensure that δ converges to 0.

Similarly, if $f(x) = f_1(x)f_2(x)$ (using the above notation), $f(x) = \tilde{\Theta}(N^{(a+b)})$. Then, we have:

$$q = 2^k = \tilde{\Theta}(N^4) \tag{5.49}$$

$$M = N(1+k) = \tilde{\Theta}(N) \tag{5.50}$$

$$\alpha q \overset{(5.41)}{=} \sqrt{4N + \omega(\sqrt{\log N})^2 + 1} = \tilde{\Theta}(\sqrt{N}) \tag{5.51}$$

$$\alpha_0 \overset{(5.43)}{=} \frac{\sqrt{4N+1}}{q} \overset{(5.49)}{=} \tilde{\Theta}(N^{-7/2}) \tag{5.52}$$

$$r_{\max} \overset{(5.33)}{=} \frac{q}{4\sqrt{\left(C \times \alpha q \times \sqrt{N}(\sqrt{k}+\sqrt{2}+1)\right)^2 + 1}} \overset{(5.51)}{=} \frac{\tilde{\Theta}(N^4)}{\tilde{\Theta}(N)} = \tilde{\Theta}(N^3) \tag{5.53}$$

$$r_{\text{safe}} \overset{(5.34)}{=} r_{\max} - \alpha q\sqrt{N+M+n} \overset{(5.53)}{=} \tilde{\Theta}(N^3) - \tilde{\Theta}(N) = \tilde{\Theta}(N^3) \tag{5.54}$$

$$\mu \overset{(5.35)}{=} \left\lfloor \frac{r_{\text{safe}}}{\sqrt{N+M+n}} \right\rfloor \overset{(5.54)}{=} \tilde{\Theta}(N^{5/2}) \tag{5.55}$$

$$\delta_m \overset{(5.37)}{=} \frac{(\alpha q+1)(N+M+n)^{3/2}}{\mu + 1/2} = \frac{\tilde{\Theta}(\sqrt{N})\tilde{\Theta}(N^{3/2})}{\tilde{\Theta}(N^{5/2})} = \tilde{\Theta}\left(\frac{1}{\sqrt{N}}\right) \tag{5.56}$$

$$\mu - \alpha q\sqrt{N+M+n} \overset{(5.55)}{=} \tilde{\Theta}(N^{5/2}) - \text{`}\tilde{\Theta}(N) = \tilde{\Theta}(N^{5/2}) \tag{5.57}$$

Therefore, for sufficiently large n, we have $\left\lfloor \mu - \alpha q\sqrt{N+M+n} \right\rfloor \overset{(5.57)}{\geq} 0$, $\gamma = \tilde{O}(\frac{N}{\alpha_0}) \overset{(5.52)}{=} \mathsf{poly}(N)$ and $\delta_m = \tilde{\Theta}\left(\frac{1}{\sqrt{N}}\right)$, meaning that the probability of not having two preimages converges towards 0. □

5.4 Discussions and Open Questions

In this chapter, we saw constructions that fulfill the requirements needed by our various QFactory protocols: when relying on LWE with superpolynomial noise ratio, we can obtain a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family, which is exactly what is required in protocols described in Sections 4.3 and 4.4. If we prefer to rely on the more standard assumption of LWE with polynomial noise ratio, we can obtain a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family with $\delta = \frac{1}{\mathsf{poly}(\lambda)}$, which is enough for the construction provided in Section 4.6.

However, the open question described in Section 4.7 also apply to the cryptographic constructions (it is often hard to completely decouple the design of the protocols, the design of the cryptographic families, and the security proofs). Notably, it would be particularly interesting to see if revealing to the server the abort bit in the BB84-QFactory protocol (which translates to revealing of a given y has exactly 2 preimages or not)

weaken the security of our construction. For more details, we refer to the discussion in Section 4.7.2.

There is also many more questions concerning the cryptographic constructions. For instance we may want to rely on different cryptographic hardness assumption than on lattice-based cryptography. Two others major post-quantum candidates are code-based and isogeny-based cryptography. I've not tried to use isogeny-based cryptography (and I am not aware of any work that did), but I tried to construct f_k using code-based cryptography. Unfortunately, I was unable to find any working construction[9]. Intuitively, what goes wrong in code-based cryptography is that it is hard to find a set of errors E such that $E \approx E - \mathbf{e}_0$. The problem is that the Hamming distance scales poorly compared to the L_2 norm. More precisely, in code-based cryptography the errors are typically vectors having many zeros and a few ones (the weight being the number of ones): so if we define E as the set of errors having weight smaller than k and pick a random error $\mathbf{e}$ inside E, the weight w of $\mathbf{e}$ is likely to be very close to k. But if we add/substract to it a vector $\mathbf{e}_0$ of weight w_0, the sum has roughly a weight of $w + w_0 \ldots$ which is typically bigger than k, unless w_0 is very small. Unfortunately, if w_0 is too small, the security vanishes, making it hard (impossible?) to find a good balance between security and correctness using code-based cryptography. Our difficulty to find a code-based construction for QFactory may actually be linked with the difficulty to find a code-based Homomorphic Encryption scheme [Bra13, Hal17].

Another important question concerns the efficiency of the construction. For now, our non-optimized construction is likely to use a very large amount of qubits and it is of great importance to reduce this amount to make the protocol more accessible (of course, for the security proof to hold we need enough qubits to make sure it is not classically simulable). Construction based on LWE with polynomial noise ratio may in particular be interesting to reduce this cost.

The question of the construction of appropriate cryptographic families also applies for new potential applications for classical-clients as discussed in Section 4.7.3. As already shown, in can have some extremely unusual and surprising properties, indicating that classical-client quantum cryptography may lead to new interesting questions, constructions and security proofs in classical lattice-based cryptography.

[9] Of course it does not mean that there is none: also, I tried to find such a construction a long time ago, while I had a less accurate view on the problem than now. But nevertheless, I still think that finding a code-based construction would be challenging, if not impossible, and likely to be much less efficient.

Chapter 6

Impossibility Results in Composable Security

"People say nothing[1] *is impossible, but I do nothing*[1] *every day."*

— A. A. Milne, *Winnie-the-Pooh*

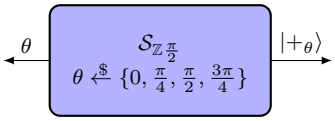
In Chapter 4 we developed a protocol to achieve classical-client Remote State Preparation (RSP), allowing us to obtain a classical-client blind quantum computing protocol based on the UBQC protocol. However, all the proofs of security were derived in the game-based security model, following the spirit of the IND-CPA definition. Unfortunately, this model of security does not guarantee that our protocol stays secure when it is used inside other protocols (a new proof of security must be written). It would therefore be much more interesting to prove the security of QFactory in a composable model—such as the Constructive Cryptography (CC) framework—to save us from re-proving the security for each new application.

However, we show in this chapter that it is impossible to prove the security of *any* classical-client RSP protocol (even noisy ones) in the CC framework. As it, this result does not rule out the possibility of a secure and composable classical-client UBQC protocol: it may be possible that a given protocol is not composable but that the usage of this protocol inside another one gives a composable protocol. Unfortunately, we also prove that it is impossible to prove the composable security of any protocol that is made of UBQC where quantum communications are replaced with a classical-client RSP protocol.

[1] Or was it "RSP"?

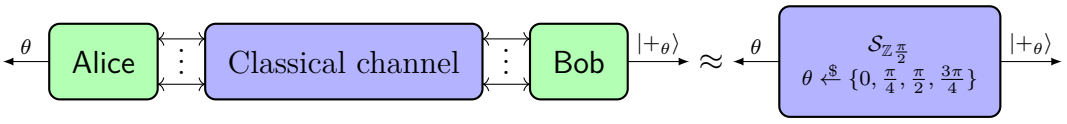
6.1 Quick Overview

In this section we give an informal overview of our approach. As explained in Section 6.2, in the CC framework the abstract specification of any protocol is characterized by a *resource*—an interactive machines with one interface per party—defining its security. Intuitively, at the end of an RSP protocol Bob gets a quantum state and Alice gets a description corresponding to this quantum state. So one of the most simple RSP resource could be:



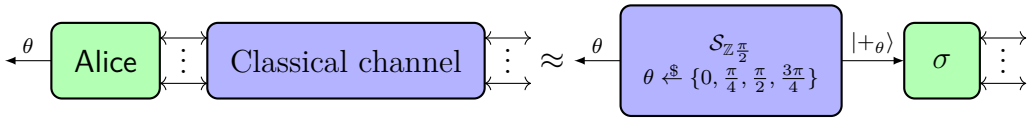
(6.1)

(We will later generalize significantly this definition, including noisy, leaky or interactive resources.) We will say that a protocol is an $\mathsf{RSP}_{\mathsf{CC}}$ protocol if it (computationally) realizes an RSP resource using a purely classical channel. In term of correctness it means that these two systems cannot be distinguished by the *distinguisher*[2]:



(6.2)

And in term of security, it means that there exists a simulator σ such that these two systems are indistinguishable (we refer to Section 6.2 for more details on the role of simulators):



(6.3)

Impossibility of $\mathsf{RSP}_{\mathsf{CC}}$ protocols. We show in Section 6.3 a wide-ranging limitation to the universally composable guarantees that any $\mathsf{RSP}_{\mathsf{CC}}$ protocol can achieve. We prove that if an $\mathsf{RSP}_{\mathsf{CC}}$ protocol realizes an RSP resource, then this resource is *describable*: roughly speaking, this notion measures how leaky an RSP resource is, i.e. what amount of information about the classical description of the final state can be extracted by an unbounded malicious server. We emphasize that even if this specific property is an information-theoretic notion, our final impossibility result also targets computational security. In this way, it rules out a wide set of desirable resources, even against computationally bounded distinguishers.

[2]Technically we should add a filter, but this is not required in this simple example.

Theorem 6.3.6 (Security Limitations of $\mathsf{RSP}_{\mathsf{CC}}$, informal). *Any RSP resource, realizable by an $\mathsf{RSP}_{\mathsf{CC}}$ protocol secure against QPT distinguishers, must leak an encoded, but complete description of the generated quantum state to the server.*

The importance of Theorem 6.3.6 lies in the fact that it is drawing a connection between the composability of an $\mathsf{RSP}_{\mathsf{CC}}$ protocol—a *computational* notion—with the statistical leakage of the ideal functionality it is constructing—an *information-theoretic* notion. This allows us to use fundamental physical principles such as no-cloning or no-signaling in the security analysis of *computationally* secure $\mathsf{RSP}_{\mathsf{CC}}$ protocols. As one direct application of this powerful tool, we show that secure implementations of the ideal resource in Eq. (6.1) give rise to the construction of a quantum cloner, and are hence impossible.

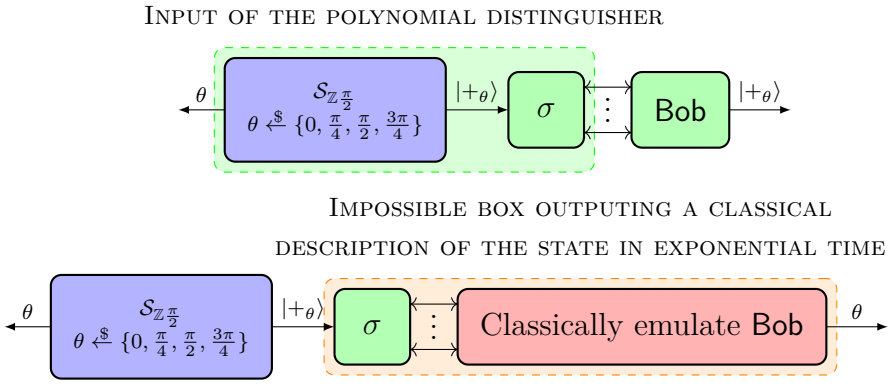


Figure 6.1: Idea of the proof of impossibility of composable $\mathsf{RSP}_{\mathsf{CC}}$, exemplified by the $\mathcal{S}_{\mathbb{Z}\frac{\pi}{2}}$ primitive from Eq. (6.1).

Proof Sketch. While Theorem 6.3.6 applies to much more general RSP resources having arbitrary behavior at its interfaces and targeting any output quantum state, for simplicity we exemplify the main ideas of our proof for the ideal resource $\mathcal{S}_{\mathbb{Z}\frac{\pi}{2}}$ (the proof is pictured in Figure 6.1). The composable security of a protocol realizing $\mathcal{S}_{\mathbb{Z}\frac{\pi}{2}}$ implies, by definition, the existence of a simulator σ which turns the right interface of the ideal resource into a completely classical interface as depicted in Eq. (6.3). Running the protocol of the honest server with access to this classical interface allows the distinguisher to reconstruct the quantum state $|+_\theta\rangle$ received by the simulator from the ideal resource. Since the distinguisher also has access to θ via the left interface of the ideal resource, it can perform a simple measurement to verify the consistency of the state obtained after interacting with the simulator. By the correctness of the protocol, the obtained quantum state $|+_\theta\rangle$

must therefore indeed comply with θ. We emphasize that this consistency check can be performed efficiently, i.e. by *polynomially-bounded* quantum distinguishers.

Since the quantum state $|+_\theta\rangle$ is transmitted from σ to the distinguisher over a classical channel, the ensemble of exchanged classical messages must contain a complete encoding of the description of the state, θ. A (possibly computationally unbounded) algorithm can hence extract the actual description of the state by means of a classical emulation of the honest server. This property of the ideal resource is central to our proof technique, and corresponds to describability.

Having a full description of the quantum state produced by $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ would allow us to clone it, a procedure prohibited by the no-cloning theorem. We conclude that the resource $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ cannot be constructed from a classical channel only. □

Remark 6.1.1. One could attempt to modify the ideal resource, to incorporate such an extensive leakage, which is necessary as the above proof implies. However, this yields an ideal resource that is actually not a useful idealization or abstraction of the real world (because it is fully leaky, i.e. reveals to a malicious server the full classical description of the state) which puts in question whether they are at all useful in a composable analysis. Indeed, usually in CC the resources considered are "trivially secure" (i.e. statistically secure, in the sense that the guarantees that we obtain does not depend on the power of the adversary), and we can then claim that a protocol is secure because it is (for any computationally bounded distinguisher) indistinguishable from the information-theoretically secure resource. Consider for example constructions of composite protocols that utilize a (non-leaky) ideal resource as a sub-module, say that leaks only the size of an encrypted message. These constructions require a fresh security analysis if the sub-module is replaced by any leaky version of it (like a resource leaking a specific encrypted form of the message), but since the modified resource is very specific and not trivially secure, it appears that this replacement does not give any benefit compared to directly using the implementation as a subroutine and then examining the composable security of the combined protocol as a whole.

Impossibility of composable $\mathsf{UBQC}_{\mathsf{CC}}$. The previous impossibility result does not prevent us from using an $\mathsf{RSP}_{\mathsf{CC}}$ protocols as a subroutine in other specific protocols—including in UBQC—and expect the overall protocol to still construct a useful ideal functionality. Unfortunately, as we show in Section 6.4, $\mathsf{UBQC}_{\mathsf{CC}}$ fails to provide the expected composable security guarantees once classical remote state preparation is used

to replace the quantum channel. This holds even if the distinguisher is computationally bounded.

Theorem 6.4.9 (Impossibility of $\mathsf{UBQC}_{\mathsf{CC}}$, informal). *No $\mathsf{RSP}_{\mathsf{CC}}$ protocol can replace the quantum channel in the UBQC protocol while preserving composable security.*

Proof Sketch. The proof proceeds in three steps. Firstly, we realize that the possibility of a composable $\mathsf{UBQC}_{\mathsf{CC}}$ protocol, which delegates arbitrary quantum computation, can be reduced to the possibility of any composable $\mathsf{UBQC}_{\mathsf{CC}}$ protocol that delegates single-qubit quantum computation. The latter protocol is much simpler to analyse. Next, we show that the single-qubit resource corresponding to $\mathsf{UBQC}_{\mathsf{CC}}$ can be seen as an RSP resource. This step allows us to employ the toolbox we developed for our previous result (Theorem 6.3.6). Finally, we show that the existence of a simulator for such an RSP functionality (that leaks the classical description, even in the form of an encoded message) would violate the no-signaling principle. Therefore, via this series of reduction, we show that the UBQC functionality, as defined in [DFP+14], cannot be realized by the combinaison of a classical-client RSP protocol and the UBQC protocol. □

Therefore, the two above results show that our QFactory protocols (GHZ-QFactory, BB84-QFactory, $\mathbb{Z}_{\frac{\pi}{4}}$-QFactory...) and our $\mathsf{UBQC}_{\mathsf{CC}}$ protocol could not have been proven secure in this composable framework. But it does not mean that they are not secure: the game-based proofs are still valid. Note also that an intermediate model known as *standalone* also exists, allowing only sequential composition: it is an open question of whether it is possible to prove the security of any $\mathsf{RSP}_{\mathsf{CC}}$ protocol in this framework. Our impossibility results also do not rule out the possibility of other composable blind quantum computing protocols. Notably, it may be possible that the verifiable protocol VBQC of [FK17] does not suffer from this issue[3].

6.2 The Constructive-Cryptography Framework

We describe now the Constructive-Cryptography Framework. Compared to game-based security, composable security takes a different approach to phrase the guarantees achieved by a protocol. Loosely speaking, a protocol is composable when it is shown to be "secure"

[3]Note that even if [GV19] proves the security of a classical-client VBQC (with polynomial security), they use an additional assumption called *Measurement Buffer* (discussed later in Remark 6.3.11) which re-creates quantum communication between the simulator and the distinguisher. Therefore, this is not in contradiction with the results and questions presented in this chapter.

in an arbitrarily adversarial environment[4]. There are several approaches which provide a general framework to study this cryptographic definitions [Can01, BPW03, MR11, Mau12, Unr10], but we will focus in this work on *Constructive Cryptography* (CC) (also known under the term Abstract Cryptography (AC)) [MR11, Mau12].

Note on the Universal Composability framework. CC is very close to the *Universal Composability* (UC) framework [Can01] which is more famous in classical cryptography. However, UC takes a "bottom-up" approach (i.e. it first defines Turing Machines, how they communicate formally, and finally defines the security) and is therefore not suitable for quantum cryptography since it was designed for classical Turing Machines from the beginning.

On the other hand, CC takes a "top-down" approach, i.e. it abstracts the notion of party and only defines a set of properties that the model of computation used by the parties must respect in order to obtain security. It is then possible to implement the model of computation of the parties using Turing Machines, but also using QPT interactive machines, unbounded machines, or even model protocol involving time such as relativistic protocols. This abstraction also allows clearer proofs since we do not need to go too deep into the low-level description of the parties.

Note also that a quantum version of UC was developed in [Unr10]. Because of the reasons mentioned above, combined with the fact that the results of interest for this work was stated in CC [DFP+14, DK16], the choice of using CC was more natural. Note however that we expect our results to also apply to Quantum UC: when considering only two QPT parties, both frameworks are mostly equivalent.

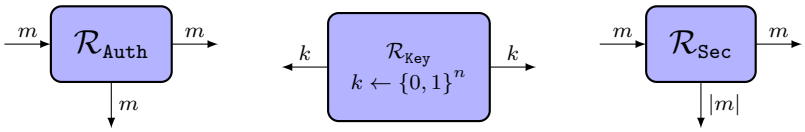
Introduction to Constructive Cryptography. In CC—and more generally in simulation-based frameworks—the security of a protocol is defined by saying that a "real world" (basically running the protocol) is indistinguishable from an "ideal world", which runs an idealized, trivially secure version of the protocol. This means that the real world is at least as secure as the ideal world, since if an attack were possible in the real world, then it would have been possible to apply this same attack in the ideal world (otherwise it would be easy to distinguish both worlds)... which is impossible because the ideal world is trivially secure. We will now formalize this first intuition.

The basic elements of CC are *systems*: objects with well-distinguished and labeled interfaces. For instance, in a two-party protocol, we can define $\mathcal{I} = \{A, B\}$ as the two

[4]Of course, the environment may still be limited to "efficient" computations.

interfaces accessible to the first and the second party: each interface represents the actions that are accessible by that agent. The system uses interfaces to exchange information with the outside world and/or other systems, and can internally do some computations (depending on how we instantiate CC, it could be unbounded or restricted to PPT or BQP computations). Systems are grouped in distinct classes: resources, converters and distinguisher.

Resources Φ are the central elements of CC[5]. They have three (related) roles: first they are used at the abstract level to specify the relevant properties of a protocol (in term of security and correctness). Secondly, they can be used to model the fundamental requirements of a protocol, like channels. Thirdly, they will be used as a container to the actual protocol to describe the real world. For instance, we can define the following three resources[6] having three interfaces $\mathcal{I} = \{A, B, E\}$, where A is represented on the left of the resource, B on the right, and E behind:



(6.4)

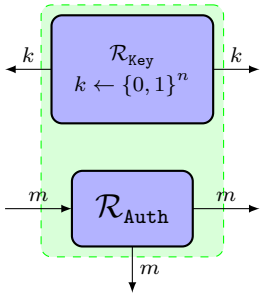
It is easy to see that $\mathcal{R}_{\texttt{Auth}}$ corresponds to an authenticated channel (readable but unalterable by Eve), $\mathcal{R}_{\texttt{Key}}$ is a key distribution mechanism (not even readable by Eve), and $\mathcal{R}_{\texttt{Sec}}$ is a secure channel (unalterable by Eve, who can learn at most the size of the transmitted message). Note how the security properties are trivially derivable from the resources: a resource will be used as the definition of the security of a protocols. You can also see that the security properties we derived above are true in an information-theoretic sense, meaning that we do not need to bound the power of Eve to derive these statements (of course, the final protocol may be only computationally secure). Note also that it is possible to group multiple resources into a bigger resource by merging their interfaces (the precise addressing mechanism is not of interest at this level of abstraction, we just need to know that there exists a parallel composition operation $\|: \Phi \times \Phi \to \Phi$ that has

[5]They roughly correspond to Ideal Functionalities in UC.

[6]When the operations done by the resource are simple enough, we write the computation directly on the wires instead of inside the resource.

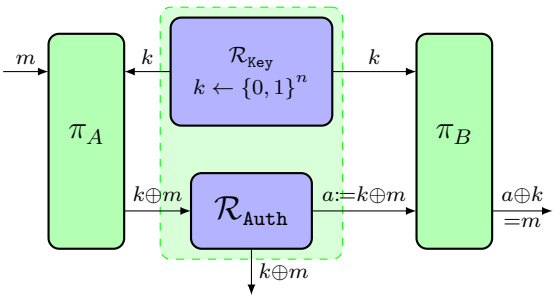
some properties described later):

$$\mathcal{R} \coloneqq \mathcal{R}_{\texttt{Key}} \| \mathcal{R}_{\texttt{Auth}} = \tag{6.5}$$



Converters Σ, on the other hand, are systems limited to two interfaces, an inside one (typically attached to a resource, potentially grouping multiple sub-resources) and an outside one (in order to take the inputs of the protocol and deliver outputs). A converter represents the actions performed by a given party and the name reflects the fact that a converter *converts* the functionality of the resource's interface it is attached to into a new functionality on the outside. A resource having a converter attached to one of its interfaces continues to qualify as a resource. For instance, we can attach to the above resource two converters π_A (playing the role of Alice) and π_B (playing the role of Bob) to obtain a new resource $\mathcal{S}$ doing the One-Time-Pad (OTP) encryption protocol (it will play the role of the "real world"):

$$\mathcal{S} \coloneqq \pi_A^A \pi_B^B \mathcal{R} = \tag{6.6}$$



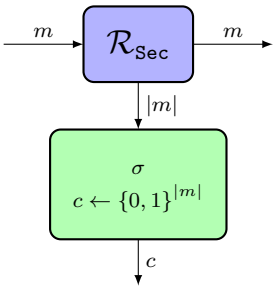
Usually, if $\pi_A \in \Sigma$ is a converter and $\mathcal{R}$ a resource, we write $\pi_A^i \mathcal{R}$ to denote new resource where the inner interface of π_A is connected to the interface i of $\mathcal{R}$, the outer interface of π_A being the new interface i (or we just use a graphical diagram as we did until now to keep simple notations). This gives a star-shaped topology where the resource is at the center and a chain of converters is attached to each interface, resulting in a new resource with the same set of interfaces. In this theses we will often have only two interfaces $\mathcal{I} = \{A, B\}$, so we will write the converter on the left of the resource when it is plugged on the interface A, and we will put the converter on the right of the resource when it is plugged on the interface B: $\pi_A^A \mathcal{R}$ is denoted $\pi_A \mathcal{R}$ while $\pi_A^B \mathcal{R}$ is $\mathcal{R} \pi_A$.

The abstraction used by the Constructive Cryptography framework allows us to instantiate resources, converters and distinguishers (see below) in different manners

(they can be unbounded, PPT or QPT interactive machines, or even be more complex to describe relativistic protocols). For the security properties to apply, we just need to have some general properties on them. Notably we require (Φ, Σ) to be a *cryptographic algebra* [Mau12, Def. 1] for the interfaces $\mathcal{I}$: resources Φ must be equipped with a parallel composition $\|: \Phi \times \Phi \to \Phi$ and a mapping $\Sigma \times \Phi \times \mathcal{I} \to \Phi$ (corresponding to our notation $\pi_A^i \mathcal{R}$ for $\pi_A \in \Sigma$, $i \in \mathcal{I}$ and $\mathcal{R} \in \Phi$) defining the resource obtained when converter π_A is attached to the interface $\mathcal{I}$ of $\mathcal{R}$. Moreover for any resource $\mathcal{R}$, converters π_A, π_B and interfaces $i \neq j$ we must have $\pi_A^i \pi_B^j \mathcal{R} = \pi_B^j \pi_A^i \mathcal{R}$ (composition order does not matter), and there should exist a special neutral converter $\mathbf{1} \in \Sigma$ such that $\mathbf{1}^i \mathcal{R} = \mathcal{R}$.

To define composition of protocols, it will also be useful to define a sequential composition law $\circ: \Sigma \times \Sigma \to \Sigma$ and a parallel composition law $\|: \Sigma \times \Sigma \to \Sigma$ on converters such that $(\pi_2 \circ \pi_1)^i \mathcal{R} \coloneqq \pi_2^i (pi_1^i \mathcal{R})$ and $(\pi_1 \| \pi_2)^i (\mathcal{R} \| \mathcal{S}) \coloneqq (\pi_1^i \mathcal{R}) \| (\pi_2^i \mathcal{S})$[7] for all $i, \mathcal{R}, \mathcal{S}, \pi_1, \pi_2$.

A *simulator* is a converter used when doing the security proof in order to define our ideal world (which is supposed to be indistinguishable from the real world). For instance, here we would like to say that the OTP protocol *realizes* the secure channel $\mathcal{R}_{\mathsf{Sec}}$. So we can define our simulator σ and attach it to $\mathcal{R}_{\mathsf{Sec}}$ as follows:

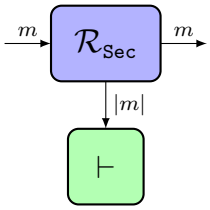


(6.7)

A *filter* (usually denoted $\vdash$) is a special converter used to force a honest behaviour on a given interface of a resource. They are usually used to prove the correctness of a protocol, as they describe what can be done in an honest run. They are removed to provide full power to a cheating adversary or to a simulator. Because an honest evedropper should not learn any information about the transmitted message, in our example the simulator is simply blocking all incoming messages on Eve's interface (there is no need for a simulator on the other interfaces since during an honest protocol we do

[7] We do not need to define the notation $(\pi_1 \| \pi_2)\mathcal{T}$ for resources $\mathcal{T}$ that are not of the form $\mathcal{T} = \mathcal{R} \| \mathcal{S}$.

want to send and receive m and do not want to filter it):

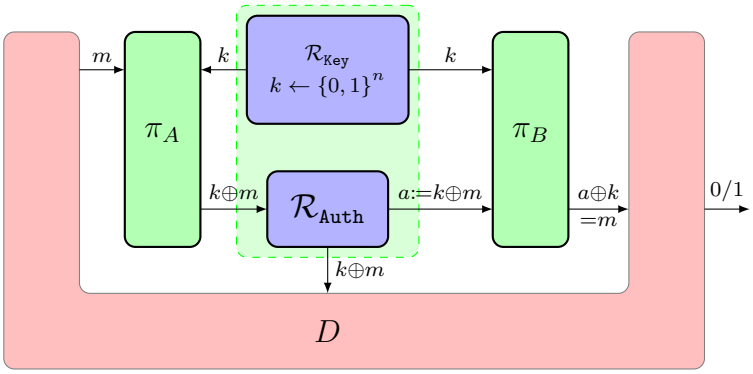


(6.8)

Sometimes, in order to keep the filter simple, the functionality accepts as a first message a bit c which says if the party wants to behave honestly ($c = 0$) or maliciously ($c = 1$). That way, the filter $\vdash^{c=0}$ (or simply $\vdash$) just sends $c = 0$ to the resource, and then forwards all the messages between it's inner and outer interface.

A *distinguisher*[8] is a system that helps to quantify the distance between resources. Given an n-interface resource $\mathcal{R}$, a distinguisher $D \in \mathcal{D}$ outputs a bit determined after interacting with the n interfaces of $\mathcal{R}$ (we denote by $D\mathcal{R}$ this random variable). You can imagine that this bit is set to, say, 0 when the distinguisher thinks it is interacting with the first resource, and 1 otherwise. With our example, this can be drawn as follows:

$D\mathcal{S} :=$



(6.9)

Then the distance (actually it is a pseudo-metric) between two resources $\mathcal{R}$ and $\mathcal{S}$ is defined by the best advantage ε a distinguisher $D \in \mathcal{D}$ can achieve when trying to determine which resource it is interacting with. This leads to the following definition:

$$\mathcal{R} \approx_\varepsilon \mathcal{S} :\Longleftrightarrow \Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) \leq \varepsilon \tag{6.10}$$

with $\Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) = \sup_{D \in \mathcal{D}} \Delta(D\mathcal{R}, D\mathcal{S})$, where $\Delta(D\mathcal{R}, D\mathcal{S})$ is the statistical distance between the distributions $D\mathcal{R}$ and $D\mathcal{S}$. Note that $\Delta^{\mathcal{D}}$ defines a pseudo-metric: $\forall \varepsilon > 0, (\mathcal{R}, \mathcal{S}, \mathcal{T}) \in \Phi^3$,

$$\Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{R}) = 0 \tag{6.11}$$

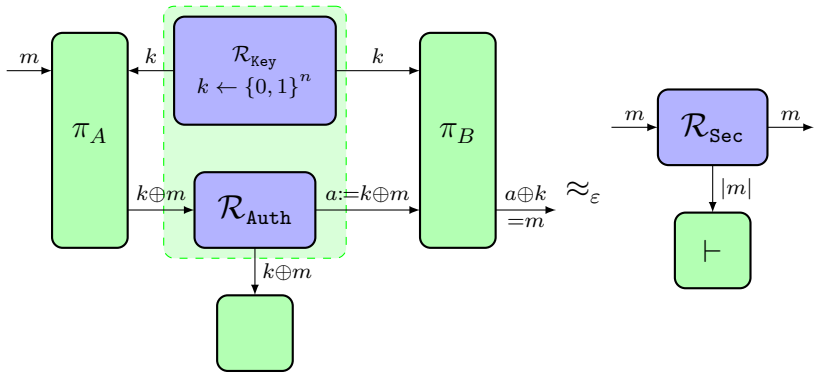
$$\Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) = \Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) \tag{6.12}$$

$$\Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) \leq \Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{T}) + \Delta^{\mathcal{D}}(\mathcal{T}, \mathcal{R}) \tag{6.13}$$

[8] In UC the distinguisher corresponds to the environment.

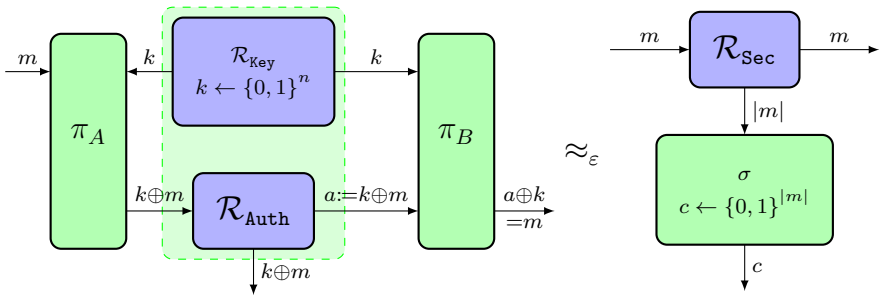
For the security proofs to hold, we expect this metric to be *compatible* [Mau12, Def. 2] with the cryptographic algebra (Φ, Σ), which means that we expect to have $\Delta^{\mathcal{D}}(\mathcal{R}\|\mathcal{R}', \mathcal{S}\|\mathcal{S}') \leq \Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S}) + \Delta^{\mathcal{D}}(\mathcal{R}', \mathcal{S}')$ and $\Delta^{\mathcal{D}}(\pi_A^i\mathcal{R}, \pi_A^i\mathcal{S}) \leq \Delta^{\mathcal{D}}(\mathcal{R}, \mathcal{S})$ for all resources $\mathcal{R}, \mathcal{R}', \mathcal{S}, \mathcal{S}'$, converter π_A and interface i. This is always the case if distinguishers can "absorb" converters and resources [Mau12, Lem. 1]: $\mathcal{D}\Sigma^i \subseteq \mathcal{D}$, $\mathcal{D}[\cdot\|\Phi] \subseteq \mathcal{D}$ and $\mathcal{D}[\Phi\|\cdot] \subseteq \mathcal{D}$ (it will always be the case).

We now have all the tools to describe how a protocol (specified as a tuple of converters, one converter per party) using internally a fundamental resource $\mathcal{R}$ can securely construct an (ideal) resource $\mathcal{S}$. Informaly, we want to check two properties: correctness and security. For the correctness, we just need to verify that the honestly performed protocol is indistinguishable from the filtered resource $\mathcal{S}$ (in our case, the honest behavior of Eve is to forget everything):



(6.14)

For the security, we need to check that for each subset of potentially malicious party, if we remove them from the protocol, there exists simulators (one per interface of the resource $\mathcal{S}$) that make the two worlds indistinguishable. In our example, only Eve can be malicious, so it gives:



(6.15)

The two above equations can be checked easily by realizing that the input-output distributions are statistically identical (the probability of observing a given string on Eve's interface is even independent of the input m). It means that the OTP protocol is statistically secure, and realizes a secure channel when implemented over an authenticated

channel. The intuition behind this is that if no distinguisher can know whether it is interacting with an ideal resource or with the real protocol, then it means that any attack done in the "real world" can also be done in the "ideal world". Because the ideal world is secure by definition, so is the real world. To give a more concrete example, if it were possible to extract m from Eve's interface in the OTP protocol, it would also be possible to extract m from Eve's interface in the ideal world—which corresponds to the output of the simulator—otherwise it would be easy to distinguish both world by checking if the input m corresponds to the extracted m. But this is of course impossible since the simulator only has access to the size of m, and therefore does not output enough information to recover m.

We also emphasis that even if this example showcases statistical security, the same argument also applies for computationally secure protocols (we could replace $k \oplus m$ with the encryption of m, and ask to π_B to decrypt the message). The only difference would be that Eq. (6.15) would be true only for QPT distinguishers: it would then be computationally hard to distinguish a computationally secure protocol from a statistically secure resource.

We give now the more formal definition, directly stated for the special case we are interested in in the rest of this thesis, namely in two-party protocols between a client A and a server B, where A is always considered to be honest. The definition can be simplified as follows:

Definition 6.2.1 ([MR11, Mau12])**.** *Let $\mathcal{I} = \{A, B\}$ be a set of two interfaces (A being the left interface and B the right one), and let $\mathcal{R}, \mathcal{S}$ be two resources. Then, we say that for the two converters π_A, π_B, the protocol $\boldsymbol{\pi} := (\pi_A, \pi_B)$* (securely) constructs *$\mathcal{S}$ from $\mathcal{R}$ within ε, or that $\mathcal{R}$* realizes *$\mathcal{S}$ within ε, denoted:*

$$\mathcal{R} \xrightarrow[\varepsilon]{\boldsymbol{\pi}} \mathcal{S} \tag{6.16}$$

if the following two conditions are satisfied:

1. *Availability (i.e. correctness):*

$$\pi_A \mathcal{R} \pi_B \approx_\varepsilon \mathcal{S} \vdash \tag{6.17}$$

2. *Security: there exists $\sigma \in \Sigma$ (called a simulator) such that:*

$$\pi_A \mathcal{R} \approx_\varepsilon \mathcal{S}\sigma \tag{6.18}$$

As promised, it is now possible to compose protocols sequentially or in parallel to realize more complex resources while ensuring the overall protocol is still secure:

Theorem 6.2.2 ([MR11, Mau12]). *If the metric $\Delta^{\mathcal{D}}$ is compatible with the cryptographic algebra (Φ, Σ), the construction $\mathcal{R} \xrightarrow[\varepsilon]{\pi} \mathcal{S}$ is generally composable, i.e. we have:*

- $\mathcal{R} \xrightarrow[\varepsilon]{(\pi_A,\pi_B)} \mathcal{S} \wedge \mathcal{S} \xrightarrow[\varepsilon']{(\pi'_A,\pi'_B)} \mathcal{T} \Rightarrow \mathcal{R} \xrightarrow[\varepsilon+\varepsilon']{(\pi'_A\circ\pi_A,\pi'_B\circ\pi_B)} \mathcal{T}$
- $\mathcal{R} \xrightarrow[\varepsilon]{(\pi_A,\pi_B)} \mathcal{S} \wedge \mathcal{R}' \xrightarrow[\varepsilon']{(\pi'_A,\pi'_B)} \mathcal{S}' \Rightarrow \mathcal{R}\|\mathcal{R}' \xrightarrow[\varepsilon+\varepsilon']{(\pi_A\|\pi'_A,\pi_B\|\pi'_B)} \mathcal{S}$
- $\mathcal{R} \xrightarrow[0]{(\mathbf{1},\mathbf{1})} \mathcal{R}$

As already discussed, it is possible to instantiate the resources Φ, converters Σ and distinguishers $\mathcal{D}$ in different ways to obtain different security guarantees. In this thesis, we will focus mostly on two instantiations. When all the systems (resources, converters, and distinguishers) are run in polynomial time on an interactive quantum machine (we say that the systems are *feasible*, denoted as $(\Phi^f, \Sigma^f, \mathcal{D}^f)$), we will say that the security is computational. If the systems are unbounded $(\Phi^u, \Sigma^u, \mathcal{D}^u)$ we will refer to statistical or information-theoretic security. The precise mathematical formalism used to describe the system models is not relevant in this work, but as discussed in [PR21] it is possible to use combs (see Figure 3.1) for simple 2-party protocols (which is the case of this paper), while more complex quantum protocols (involving time, relativistic cryptography...) can be modelled using [PMM+17] (classical systems also have their own formalisms [MMP+18, LSB+19]).

6.3 Impossibility of Composable Classical RSP

In this section, we formalize and generalize the proof sketched in the overview Section 6.1 concerning the impossibility of $\mathsf{RSP}_{\mathsf{CC}}$ protocols: first we define what RSP achieve in terms of resources and subsequently quantify the amount of information that an ideal RSP resource must leak to the server. We show that this leakage is basically complete and that the most meaningful and natural RSP resources cannot be realized from a classical channel alone. We finally conclude the section by looking at the class of imperfect (describable) RSP resources which avoid the no-go result at the price of being "fully-leaky", not standard, and having an unfortunately unclear composable security.

6.3.1 Remote State Preparation and Describable Resources

We first introduce, based on the standard definition in the Constructive Cryptography framework, the notion of *correctness* and *security* of a two-party protocol which constructs (realizes) a resource from a *classical* channel $\mathcal{C}$.

Definition 6.3.1 (Classically-Realizable Resource)**.** *An ideal resource $\mathcal{S}$ is said to be* ε-classically-realizable *if it is realizable (in the sense of Theorem 6.2.2) from a* classical *channel $\mathcal{C}$, i.e. if there exists a protocol $\boldsymbol{\pi} = (\pi_A, \pi_B)$ between two parties (interacting classically) such that:*

$$\mathcal{C} \xrightarrow[\varepsilon]{\boldsymbol{\pi}} \mathcal{S} \tag{6.19}$$

A simple ideal prototype that captures the goal of an RSP protocol could be phrased as follows: the resource outputs a quantum state (chosen from a set of states) on one interface and a classical description of that state on the other interface to the client. For our purposes, this view is too narrow and we want to generalize this notion. For instance, a resource could accept some inputs from the client or interact with the server, and it may still be possible to use this resource to come up with a quantum state and its description. More precisely, if there is an efficient way to convert the client and server interfaces to comply with the basic prototype above, then such a resource can be understood as RSP resource, too. To make this idea formal, we need to introduce some converters that witness this:

1. A converter $\mathcal{A}$ will output, after interacting with the ideal resource[9], a classical description $[\rho]$ which is one of the following:

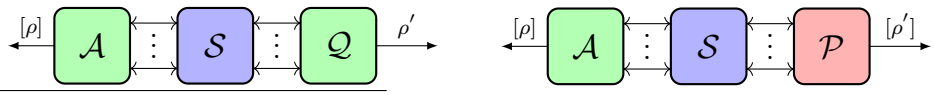
   a) A density matrix (positive and with trace 1) corresponding to a quantum state ρ.

   b) The null matrix, which is useful to denote the fact that we detected some deviation that should not happen in an honest run.

2. An (efficient) converter $\mathcal{Q}$, whose goal is to output a quantum state ρ' as close as possible to the state ρ output by $\mathcal{A}$.

3. A adversarial (potentially inneficient) converter $\mathcal{P}$ (for "photocopy"), whose goal is to output a classical description $[\rho']$ of a quantum state ρ' which is close to ρ (cf. Definition 6.3.2).

Which can be pictured as follows:



(6.20)

[9] $\mathcal{A}$ is allowed to interact with the (ideal) resource in a non-trivial manner. However, $\mathcal{A}$ will often be the trivial converter in the sense that it simply forwards the output of the ideal resource, or—when the resource waits for a simple activation input—picks some admissible value as input to the ideal resource and forwards the obtained description to its outer interface.

Then, in order to define properly our RSP resources, we start by realizing that an RSP resource must meet at least one central criteria: *accuracy*. More precisely, we want that the quantum state ρ described by $\mathcal{A}$'s output is close to $\mathcal{Q}$'s output ρ' in terms of trace distance. However, this first requirement is not enough for our impossibility result: indeed, a completely noisy resource sending a completely mixed state could easily be accurately reproduced using only classical communication (or even no communication at all) by simply outputting random states. Therefore, we intuitively want to add a restriction to avoid too noisy resources. A first solution would be to quantify the purity of the resource, asking for $\mathrm{Tr}\left(\rho'^2\right)$ to be close to 1. It turns out that these two conditions can be unified and equivalently captured requiring that the quantity $\mathrm{Tr}(\rho\rho')$ is close to one (a rigorous formulation of this claim and its proof is provided in Section 6.5).

This formula may seem a bit arbitrary. But we can also gain a more operational intuition of the notion of RSP by considering that an RSP resource (together with $\mathcal{A}$ and $\mathcal{Q}$) can be seen, not only as a box that produces a quantum state together with its description but also as a box whose accuracy can be easily *tested*[10]. For example, if such a box produces a state ρ', and pretends that the description of that state corresponds to $|\phi\rangle$ (i.e. $[\rho] = [|\phi\rangle\langle\phi|]$), then the natural way to test it would be to measure ρ' by doing a projection on $|\phi\rangle$. This test would pass with probability $p_s := \langle\phi|\rho'|\phi\rangle$, and therefore if the box is perfectly accurate (i.e. if $\rho' = |\phi\rangle\langle\phi|$), the test will always succeed. However, when ρ' is far from $|\phi\rangle\langle\phi|$, this test is unlikely to pass, and we will have $p_s < 1$. We can then generalise this same idea for arbitrary (eventually not pure) states by remarking that $p_s = \langle\phi|\rho'|\phi\rangle = \mathrm{Tr}(|\phi\rangle\langle\phi|\rho') = \mathrm{Tr}(\rho\rho')$. Indeed, this last expression corresponds[11] exactly to the probability of outputting E_0 when measuring the state ρ' according to the POVM $\{E_0 := \rho, E_1 := I - \rho\}$, and since the classical description of ρ is known, it is possible to perform this POVM and test the (average) accuracy of our box. This motivates the following definition, which characterizes the set of RSP resources.

Definition 6.3.2 (RSP resources)**.** *A resource $\mathcal{S}$ is said to be a* remote state preparation resource *within ε with respect to converters $\mathcal{A}$ and $\mathcal{Q}$ if the following three conditions hold: (1) both converters output a single message at the outer interface, where the output $[\rho]$ of $\mathcal{A}$ is classical and is either a density matrix or the null matrix, and the output ρ' of $\mathcal{Q}$ can be any quantum state of same dimension as ρ; (2) the equation:*

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\mathcal{S}\vdash\mathcal{Q}} \left[\, \mathrm{Tr}(\rho\rho') \,\right] \geq 1 - \varepsilon \tag{6.21}$$

[10] This testable property will be of great importance in our argument later.

[11] Note that it also turns out to be equal to the (squared) fidelity between ρ and ρ' when ρ is pure.

is satisfied, where the probability is taken over the randomness of $\mathcal{A}$, $\mathcal{S}$ and $\mathcal{Q}$, and finally, (3) for all the possible outputs $[\rho]$ of $([\rho], \rho') \leftarrow \mathcal{AS} \vdash \mathcal{Q}$, if we define $E_0 = \rho$, $E_1 = I - \rho$, then the POVM $\{E_0, E_1\}$ must be efficiently implementable[12] *by any distinguisher.*

Whenever we informally speak of a resource $\mathcal{S}$ as being an RSP resource, this has to be understood always in a context where the converters $\mathcal{A}$ and $\mathcal{Q}$ are fixed.

Describable resources. So far, we have specified that a resource qualifies as an RSP resource if, when all parties follow the protocol, we know how to compute a quantum state on the right interface and classical description of a "close" state on the other interface. A security-related question now is, if it is also possible to extract (possibly inefficiently) from the right interface a *classical* description of a quantum state that is close to the state described by the client. If we find a converter $\mathcal{P}$ doing this, we would call the (RSP) resource *describable*. This is captured by the following definition.

Definition 6.3.3 (Describable Resource)**.** *Let $\mathcal{S}$ be a resource and $\mathcal{A}$ a converter outputting a single classical message $[\rho]$ on its outer interface (either equal to a density matrix or the null matrix). Then we say that $(\mathcal{S}, \mathcal{A})$ is* ε-describable *(or, equivalently, that $\mathcal{S}$ is describable within ε with respect to $\mathcal{A}$) if there exists a possibly unbounded converter $\mathcal{P}$—outputting a single classical message $[\rho']$ on its outer interface representing a density matrix—such that:*

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{ASP}} [\,\mathrm{Tr}(\rho\rho')\,] \geq 1 - \varepsilon \tag{6.22}$$

(the expectation is taken over the randomness of $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{P}$).

Reproducible converters. In the proof of our first result, we will encounter a crucial decoding step. Roughly speaking, the core of this decoding step is the ability to convert the classical interaction with a client, which can be seen as an arbitrary encoding of a quantum state, back into an explicit representation of the state prepared by the server. The ability of such a conversion can be phrased by the following definition.

Definition 6.3.4 (Reproducible Converter)**.** *A converter π that outputs (on the right interface) a quantum state ρ is said to be reproducible if there exists a (possibly inefficient) converter $\tilde{\pi}$ such that:*

[12] We could also state a similar definition when this POVM can only be approximated (for example when assuming that distinguishers only perform quantum circuits using a finite set of gates). Our results hold analogously for such an approximation.

1. *the outer interface of $\tilde{\pi}$ outputs only a classical message $[\rho']$*

2. *the converter π is perfectly indistinguishable from $\tilde{\pi}$ against any unbounded distinguisher $D \in \mathcal{D}^u$, up to the conversion of the classical messages $[\rho']$ into a quantum state ρ'. More precisely, if we denote by $\mathcal{T}$ the converter that takes as input on its inner interface a classical description $[\rho']$ of a quantum state and outputs that quantum state ρ' (as depicted in Figure 6.2), we have:*

$$\mathcal{C}\pi \approx_0^{\mathcal{D}^u} \mathcal{C}\tilde{\pi}\mathcal{T} \tag{6.23}$$

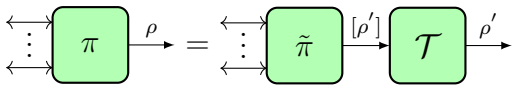


Figure 6.2: Reproducible converter.

Classical communication and reproducibility. We see that in general, being reproducible is a property that stands in conflict with the quantum no-cloning theorem. More precisely, the ability to reproduce implies that there is a way to extract knowledge of a state sufficient to clone it. We will now show that when communication is classical, converters are always reproducible: the idea is to compute classically (and inefficiently) the operations that are supposed to be done quantumly. Note that this is possible only because there is no unknown quantum input state since communication is classical.

Later, we just need to assume that the converter π interacts (classically) with the inner interface first, and finally outputs a quantum state on the outer interface, so for simplicity we will stick to that setting. In this way we can decompose $\pi = (\pi_i)_i$ as a sequence of quantum instruments[13] forming a quantum comb (see Figure 3.1) using the following notation[14]:

$$\pi := (\pi_i)_i \tag{6.24}$$

Now, we can prove that a party that produces a quantum state at the end of a protocol with exclusively classical communication is reproducible:

[13] Quantum instruments are the most general operation one can do quantumly, and are generalizations of CPTP maps to classical outputs, see Definition 2.1.5 for the formal definition.

[14] In order to deal with protocols in which the number of exchanged message is not fixed and can be arbitrarily large, we can also consider infinite sequences of π_i's, where the protocol stops when π_i outputs $\perp$.

Lemma 6.3.5. *Let $\pi = (\pi_i)_i$ (using the notation introduced Eq. (6.24)) be a converter such that:*

1. *it receives and sends only classical messages from the inner interfaces*
2. *it outputs at the end a quantum state on the outer interface*
3. *each π_i is a quantum instrument*

then π is reproducible.

Proof. The intuition behind the proof is to argue that because the only interactions with the outside world are classical as seen from Figure 3.1, the internal state of π can always be computed (in exponential time) manually.
More precisely, for all i, because π_i is a quantum instrument, there exists a set $\{\mathcal{E}_{y_i}\}$ of maps having the properties defined in Definition 2.1.5. And because for all y_i, $\mathcal{E}_{y_i}$ is completely positive, there exists a finite set of matrices $\{B_k^{(i,y_i)}\}_k$, known as Kraus operators, such that we have for all ρ (and in particular for $\rho = |x_i\rangle\langle x_i| \otimes \rho_i$):

$$\mathcal{E}_{y_i}(\rho) = \sum_k B_k^{(i,y_i)} \rho B_k^{(i,y_i)\dagger} \tag{6.25}$$

Therefore, for all x_i, ρ_i and y_i, we have with probability $p_{y_i} := \mathrm{Tr}(\mathcal{E}_{y_i}(|x_i\rangle\langle x_i| \otimes \rho_i))$:

$$\pi_i(x_i, \rho_i) = (y_i, \mathcal{E}_{y_i}(|x_i\rangle\langle x_i| \otimes \rho_i)) \tag{6.26}$$

$$= (y_i, \underbrace{\sum_k B_k^{(i,y_i)}(|x_i\rangle\langle x_i| \otimes \rho_i)B_k^{(i,y_i)\dagger}}_{\rho_{i+1}}) \tag{6.27}$$

We remark that if we know $[\rho_i]$, the coefficients of the matrix ρ_i, then for all y_i we can compute the probability p_{y_i} of outputting y_i, and the corresponding $[\rho_{i+1}]$, (the coefficients of the matrix ρ_{i+1}) by just doing the above computation. So to construct $\tilde{\pi}$ (using notations from Definition 6.3.4) we do as follows:

1. For all i we construct $\tilde{\pi}_i$, which on input $(x_i, [\rho_i])$ outputs $(y_i, [\rho_{i+1}])$ with probability p_{y_i} using the formula Eq. (6.27).
2. We define $\tilde{\pi}$ as $(\tilde{\pi}_i)$ with $[\rho_0] = (1)$.

Then, we trivially have $\mathcal{C}\pi \approx_0 \mathcal{C}\tilde{\pi}\mathcal{T}$, even for unbounded distinguishers, because $\tilde{\pi}$ is exactly the same as π, except that the representations of the quantum states in $\tilde{\pi}$ are matrices, while they are actual quantum states in π. Therefore, adding $\mathcal{T}$ (which turns any $[\rho_i]$ into ρ_i) on the outer interface (which is the only interface that sends a classical state $[\rho_i]$) gives us $\pi \approx_0 \mathcal{C}\tilde{\pi}\mathcal{T}$. □

6.3.2 Classically-Realizable RSP are Describable

In this section we show our main impossibility result about remote state preparation resources, which interestingly links a computational notion (*composability*) with an information theoretic property (*describability*).

Theorem 6.3.6 (Classically-Realizable RSP are Describable)**.** *If an ideal resource $\mathcal{S}$ is both an ε_1-remote state preparation with respect to some $\mathcal{A}$ and $\mathcal{Q}$ and ε_2-classically-realizable (including against only polynomially bounded distinguishers), then it is $(\varepsilon_1 + 2\varepsilon_2)$-describable with respect to $\mathcal{A}$. In particular, if $\varepsilon_1 = \mathsf{negl}(\lambda)$ and $\varepsilon_2 = \mathsf{negl}(\lambda)$, then $\mathcal{S}$ is describable within a negligible error $\varepsilon_1 + 2\varepsilon_2 = \mathsf{negl}(\lambda)$.*

Proof. Let $\mathcal{S}$ be an ε_1-remote state preparation resource with respect to $(\mathcal{A}, \mathcal{Q})$ which is ε_2-classically-realizable. Then there exist π_A, π_B, σ, such that:

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\mathcal{S} \vdash \mathcal{Q}} [\operatorname{Tr}(\rho\rho')] \geq 1 - \varepsilon_1 \tag{6.28}$$

$$\pi_A \mathcal{C} \pi_B \approx_{\varepsilon_2} \mathcal{S} \vdash \tag{6.29}$$

and

$$\pi_A \mathcal{C} \approx_{\varepsilon_2} \mathcal{S}\sigma \tag{6.30}$$

Now, using (6.29), we get:

$$\mathcal{A}\pi_A \mathcal{C} \pi_B \mathcal{Q} \approx_{\varepsilon_2} \mathcal{A}\mathcal{S} \vdash \mathcal{Q} \tag{6.31}$$

So it means that we cannot distinguish between $\mathcal{A}\mathcal{S} \vdash \mathcal{Q}$ and $\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}$ with an advantage better than ε_2 (i.e. with probability better than $\frac{1}{2}(1+\varepsilon_2)$). But, if we construct the following distinguisher, that runs $([\rho], \rho') \leftarrow \mathcal{A}\mathcal{S} \vdash \mathcal{Q}$ and then measures ρ' using the POVM $\{E_0, E_1\}$ (possible because this POVM is assumed to be efficiently implementable by distinguishers in $\mathcal{D}$), with $E_0 = [\rho]$ and $E_1 = I - [\rho]$ (which is possible because we know the classical description of ρ, which is positive and smaller than I, even when $[\rho] = 0$), we will measure E_0 with probability $1 - \varepsilon_1$. So it means that by replacing $\mathcal{A}\mathcal{S} \vdash \mathcal{Q}$ with $\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}$, the overall probability of measuring E_0 needs to be close to $1 - \varepsilon_1$. More precisely, we need to have:

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}} [\operatorname{Tr}(\rho\rho')] \geq 1 - \varepsilon_1 - \varepsilon_2 \tag{6.32}$$

Indeed, if the above probability is smaller than $1 - \varepsilon_1 - \varepsilon_2$, then we can define a distinguisher that outputs 0 if it measures E_0, and 1 if it measures E_1, and his

probability of distinguishing the two distributions would be equal to:

$$\frac{1}{2}\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\vdash\mathcal{Q}}[\operatorname{Tr}(\rho\rho')] + \frac{1}{2}\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{A}\pi_A\mathcal{C}\pi_B\mathcal{Q}}[\operatorname{Tr}((I-\rho)\rho')] \tag{6.33}$$

$$> \frac{1}{2}\left((1-\varepsilon_1)+1-(1-\varepsilon_1-\varepsilon_2)\right) \tag{6.34}$$

$$= \frac{1}{2}(1+\varepsilon_2) \tag{6.35}$$

So this distinguisher would have an advantage greater than ε_2, which is in contradiction with Eq. (6.31).

Using a similar argument and Eq. (6.29), we have:

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\sigma\pi_B\mathcal{Q}}[\operatorname{Tr}(\rho\rho')] \geq 1-\varepsilon_1-2\varepsilon_2 \tag{6.36}$$

We will now use $\pi_B\mathcal{Q}$ to construct a $\mathcal{B}$ that can describe the state given by the ideal resource. To do that, because $\pi_B\mathcal{Q}$ interacts only classically with the inner interface and outputs a single quantum state on the outer interface, then according to Lemma 6.3.5, $\pi_B\mathcal{Q}$ is reproducible, i.e. there exists[15] $\mathcal{B}$ such that $\mathcal{C}\pi_B\mathcal{Q} \approx_0 \mathcal{CBT}$. Therefore[16], we have:

$$\mathop{\mathbb{E}}_{([\rho],\rho')\leftarrow\mathcal{AS}\sigma\mathcal{BT}}[\operatorname{Tr}(\rho\rho')] \geq 1-\varepsilon_1-2\varepsilon_2 \tag{6.37}$$

But because $\mathcal{T}$ simply converts the classical description $[\rho']$ into ρ', we also have:

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{AS}\sigma\mathcal{B}}[\operatorname{Tr}(\rho\rho')] \geq 1-\varepsilon_1-2\varepsilon_2 \tag{6.38}$$

After defining $\mathcal{P}=\sigma\mathcal{B}$, we have that $\mathcal{S}$ is $(\varepsilon_1+2\varepsilon_2)$-describable, which ends the proof. □

While the above theorem does not rule out all the possible RSP resources, it shows that most "useful" RSP resources are impossible. Indeed, the describable property is usually not a desirable property, as it means that an (potentially unbounded) adversary could learn the description of the state it received from an ideal resource. To illustrate this theorem, we will see in the Section 6.3.3 some examples showing how this result can be used to prove the impossibility of classical protocols implementing some specific resources, and in Section 6.3.4 we will see some example of "imperfect" resources escaping the impossibility result.

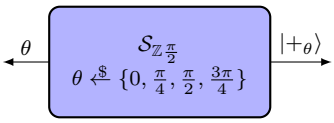
[15] Note that here $\mathcal{B}$ is not efficient anymore, so that's why in the describable definition we do not put any bound on $\mathcal{B}$, but of course the proof does apply when the distinguisher is polynomially bounded.

[16] Indeed, we also have in particular $\mathcal{AS}\sigma\mathcal{C}\pi_B\mathcal{Q} \approx_0 \mathcal{AS}\sigma\mathcal{CBT}$, and because $\mathcal{C}$ is a neutral resource [MR11, Sec. C.2] we can remove $\mathcal{C}$.

6.3.3 RSP Resources Impossible to Realize Classically

In the last section we proved that if an RSP functionality is classically-realizable (secure against polynomial quantum distinguishers), then this resource is describable by an unbounded adversary having access to the right interface of that resource. In this section we present some of these RSP resources that are impossible to classically realize.

Definition 6.3.7 (Ideal Resource $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$)**.** *$\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is the verifiable RSP resource (RSP which does not allow any deviation from the server), that receives no input, that internally picks a random $\theta \leftarrow \mathbb{Z}_{\frac{\pi}{2}}$, and that sends θ on the left interface, and $|+_\theta\rangle$ on the right interface as shown in Eq. (6.39):*



(6.39)

Lemma 6.3.8. *There exists a universal constant $\eta > 0$, such that for all $0 \leq \varepsilon < \eta$ the resource $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is not ε-classically-realizable.*

Proof. This proof is at its core a direct consequence of quantum no-cloning: If we define $\mathcal{A}(\theta) := [|+_\theta\rangle\langle+_\theta|]$ ($\mathcal{A}$ just converts θ into its classical density matrix representation) and $\mathcal{Q}$ the trivial converter that just forwards any message, then $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is a 0-remote state preparation resource with respect to $\mathcal{A}$ and $\mathcal{Q}$ because:

$$\mathop{\mathbb{E}}_{([\rho],\rho') \leftarrow \mathcal{A}\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}} \vdash \mathcal{Q}} [\,\mathrm{Tr}(\rho\rho')\,] = \frac{1}{4} \sum_{\theta \in \mathbb{Z}_{\frac{\pi}{2}}} \mathrm{Tr}(|+_\theta\rangle\langle+_\theta||+_\theta\rangle\langle+_\theta|) = 1 \geq 1 - 0 \tag{6.40}$$

Then, we remark also that there exists a constant $\eta > 0$ such that for all $\delta < \eta$, $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is not δ-describable with respect to $\mathcal{A}$.

Indeed, it is first easy to see that $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is not 0-describable with respect to $\mathcal{A}$. Indeed, we can assume by contradiction that there exists $\mathcal{P}$ such that:

$$\mathop{\mathbb{E}}_{([\rho],[\rho']) \leftarrow \mathcal{A}\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}} \mathcal{P}} [\,\mathrm{Tr}(\rho\rho')\,] = 1 \tag{6.41}$$

Then, because $\rho = |+_\theta\rangle\langle+_\theta|$ is a pure state, $\mathrm{Tr}(\rho\rho')$ corresponds to the fidelity of ρ and ρ', so $\mathrm{Tr}(\rho\rho') = 1 \Leftrightarrow \rho = \rho'$. But this is impossible because $\mathcal{P}$ just has a quantum state ρ as input, and if it can completely describe this quantum state then it can actually clone perfectly the input state with probability 1. But because the different possible values of ρ are not orthogonal, this is impossible due to the no-cloning theorem.

Moreover, it is also not possible to find a sequence $(\mathcal{P}^{(n)})_{n \in N}$ of CPTP maps that produces two copies of ρ with a fidelity arbitrary close to 1 (when $n \to \infty$), because

CPTP maps are compact and the fidelity is continuous.
Therefore, there exists a constant $\eta > 0$,[17] such that:

$$\mathop{\mathbb{E}}_{([\rho],[\rho'])\leftarrow\mathcal{A}\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}\mathcal{P}} [\,\mathrm{Tr}(\rho\rho')\,] < 1-\eta \tag{6.42}$$

Now, by contradiction, we assume that $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is ε-classically-realizable. Because $\lim_{n\to\infty}\varepsilon(n) = 0$, there exists $N \in \mathbb{N}$ such that $\varepsilon(N) < \eta/2$. So, using Theorem 6.3.6, $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ is $2\varepsilon(N)$-describable with respect to $\mathcal{A}$, which contradicts $2\varepsilon(N) < \eta$. □

Next, we describe RSP_V, a variant of $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$ introduced in [GV19]: more precisely, the set of prepared states is bigger, the adversary can make the resource abort and the client can partially choose the basis of the output state. Similar to the $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$, we prove that classically-realizing RSP_V is not possible.

Definition 6.3.9 (Ideal Resource RSP_V, See [GV19])**.** *The ideal verifiable remote state preparation resource, RSP_V, takes an input $W \in \{X, Z\}$ on the left interface, but no honest input on the right interface. The right interface has a filtered functionality that corresponds to a bit $c \in \{0, 1\}$. When $c = 1$, RSP_V outputs error message ERR on both the interfaces, otherwise:*

1. *if $W = Z$ the resource picks a random bit b and outputs $b \in \mathbb{Z}_2$ to the left interface and a computational basis state $|b\rangle\langle b|$ to the right interface;*
2. *if $W = X$ the resource picks a random angle $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$ and outputs θ to the left interface and a quantum state $|+_\theta\rangle\langle+_\theta|$ to the right interface.*

Corollary 6.3.10. *There exists a universal constant $\eta > 0$, such that for all $0 \le \varepsilon < \eta$ the resource RSP_V is not ε-classically-realizable.*

Proof. The proof is quite similar to the proof of impossibility of $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$. The main difference is that we need to address properly the abort case when $c = 1$. The main idea is to define $\mathcal{A}$ a bit differently: $\mathcal{A}$ picks always $W = X$, and outputs as ρ the classical density matrix corresponding to s when $s \neq \mathsf{ERR}$, and when $s = \mathsf{ERR}$, $\mathcal{A}$ outputs the null matrix $\rho = 0$ ($\mathcal{Q}$ is still the trivial converter). It is easy to see again that this resource is a 0-remote state preparation resource, and it is also impossible to describe it with arbitrary small probability: indeed, when $c = 1$, $\rho = 0$, so the trace $\mathrm{Tr}(\rho\rho')$ (that appears in Eq. (6.22))

[17] Note that for finding a more precise bound for η, it is possible to use Semidefinite Programming (SDP), or the method presented in [KRK13, p. 2]. However in our case it is enough to say that $\varepsilon > 0$ as we are interested only in asymptotic security.

is equal to 0. Therefore, from a converter $\mathcal{P}$ that (sometimes) inputs $c = 1$, we can always increase the value of $\mathrm{Tr}(\rho\rho')$ by creating a new converter $\mathcal{P}'$ turning c into 0. And we are basically back to the same picture as $\mathcal{S}_{\mathbb{Z}_{\frac{\pi}{2}}}$, where we have a set of states that is impossible to clone with arbitrary small probability, which finishes the impossibility proof. □

Remark 6.3.11. Note that our impossibility of classically-realizing RSP_V does not contradict the result of [GV19]. Specifically, in their work they make use of an additional assumption (the so called "Measurement Buffer" resource), which "externalizes" the measurement done by the distinguisher onto the simulator. In practice, this allows the simulator to discretely change the operation that was initially supposed to be performed by the server, or, equivalently, to temper the device used by the server. However, what our result shows is that it is impossible to realize this Measurement Buffer resource with a protocol interacting purely classically. Intuitively, the Measurement Buffer re-creates a quantum channel between the simulator and the server: when the simulator is not testing that the server is honest, the simulator replaces the state of the server with the quantum state sent by the ideal resource. This method has however a second drawback: it is possible for the server to put a known state as the input of the Measurement Buffer, and if he is not tested on that run (occurring with probability $\frac{1}{n}$), then he can check that the state has not been changed, leading to polynomial security (a polynomially bounded distinguisher can distinguish between the ideal and the real world). And because in CC, the security of the whole protocol is the sum of the security of the inner protocols, any protocol using this inner protocol will not be secure against polynomial distinguishers.

6.3.4 Accepting the Limitations: Fully Leaky RSP resources

As explained in the previous section, Theorem 6.3.6 rules out all resources that are impossible to describe with unbounded power: therefore the only type of classically-realizable RSP resources would be the one leaking the full classical description of the output quantum state to an unbounded adversary, which we will refer to as being *fully-leaky* RSP. Fully-leaky RSP resources can be separated into two categories:

1. If the RSP is describable in quantum polynomial time, then the adversary can get the secret in polynomial time. This is obviously not an interesting case as the useful properties that we know from quantum computations (such as UBQC) cannot be preserved if such a resource is employed to prepare the quantum states.

2. If the RSP are only describable inefficiently (for instance in exponential time), then these *fully-leaky* RSP resources are not trivially insecure, but their universally composable security remains unclear. Indeed, it defeats the purpose of aiming at a nice ideal resource where the provided security should be clear "by definition" and it becomes hard to quantify how the additional leakage could be used when composed with other protocols. Actually, we claim that the security we obtain is not better than game-based security. A possible remedy would be to show restricted composition following [JM17] which we discuss at the end of this paragraph.

For completeness, we present an example of a resource (producing a BB84 state corresponding to the set of states produced by the simpler QFactory protocol) that stands in this second category when assuming that post-quantum encryption schemes exist (based on the hardness of the LWE problem). As explained before, this resource needs to completely leak the description of the classical state, which in our case, is done by leaking an encryption of the description of the output state. The security guarantees therefore rely on the properties of the encryption scheme, and not on an ideal privacy guarantee as one would wish for, which is an obvious limitation.

Definition 6.3.12 (Ideal Resource $\mathsf{RSP}_{CC}^{BB,\mathcal{F}}$)**.** *Let $\mathcal{F} = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Invert})$ be a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family (Definition 4.2.1). Then, we define $\mathsf{RSP}_{CC}^{BB,\mathcal{F}}$ as pictured in Figure 6.3. B_1 represents the basis of the output state, and is guaranteed to be random even if the right interface is malicious. B_0 represents the value bit of the output state when encoded in the basis B_1, and in the worst case it can be chosen by the right interface in a malicious scenario[18]. Note however that in a malicious run, the adversary does not have access (at least not directly from the ideal resource) to the quantum state whose classical description is known by the classical client.*

Lemma 6.3.13. *The BB84-QFactory protocol (Protocol 3) securely constructs $\mathsf{RSP}_{CC}^{BB,\mathcal{F}}$ from a classical channel, where $\mathcal{F}$ is the δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family used in the protocol.*

Proof. We already know that the BB84-QFactory protocol (π_A, π_B) is correct with superpolynomial probability if the parameters are chosen accordingly (Corollary 4.4.1), therefore

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathsf{RSP}_{CC}^{BB,\mathcal{F}} \vdash \tag{6.43}$$

for some negligible ε. We now need to find a simulator σ such that

$$\pi_A \mathcal{C} \approx_{\varepsilon'} \mathsf{RSP}_{CC}^{BB,\mathcal{F}} \sigma \tag{6.44}$$

[18]Note that here the right interface can have (in a malicious scenario) full control over B_0, but in the BB84-QFactory Protocol it is not clear what an adversary can do concerning B_0.

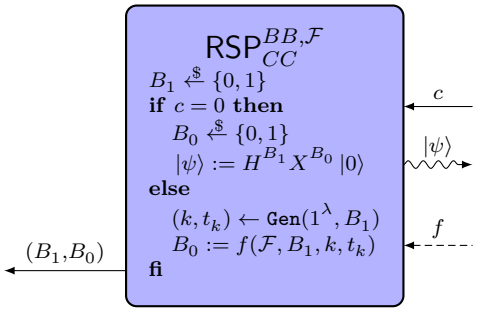


Figure 6.3: Ideal resource $\mathsf{RSP}^{BB,\mathcal{F}}_{CC}$, which prepares one of the four BB84 states. The "snake" arrow is sent only in the honest case ($c = 0$), and the dashed arrow is received only in the malicious case ($c = 1$).

The simulator is trivial here: it sends $c = 1$ to ideal resource then, it just forwards the k given by the resource to its outer interface, and when it receives the (y, b) corresponding to the measurements performed by the server, it just sets the deviation f to be the same function as the one computed by π_A. Therefore, $\pi_A \mathcal{C} \approx_0 \mathsf{RSP}^{BB,\mathcal{F}}_{CC} \sigma$, which ends the proof. □

Concluding remarks. We see that using this kind of leaky resource is not desirable: the resources are non-standard and it seems hard to write a modular protocol with this resource as an assumed resource. The resource is very specific and mimics its implementation. As such, we cannot really judge its security.

On the other hand however, if a higher-level protocol did guarantee that the value B_0 always remains hidden, i.e., a higher level protocol's output does not depend on B_0, it is easy to see that we could simulate y_0 without knowledge about B_1 thanks to the semantic security of the encryption scheme. If we fix this restricted context, the ideal resource in Figure 6.3 could be re-designed to not produce the output k at all and therefore, by definition, leak nothing extra about the quantum state (note that in such a restricted context, the simulator can simply come up with a fake encryption that is indistinguishable). This can be made formal following [JM17], but it is interesting future research to see whether it is possible to come up with restricted yet useful contexts that admit nice ideal resources for RSP following this framework.

6.4 Impossibility of Composable Classical-Client UBQC

In the previous section, we showed that it was impossible to get a (useful) composable RSP_{CC} protocol. A (weaker) RSP protocol, however, could still be used internally in other protocols, hoping for the overall protocol to be composably secure. To this end, we analyze the composable security of any $\mathsf{UBQC}_{\mathsf{CC}}$ protocol, which is following the UBQC (Protocol 1) protocol except that the quantum communication are replaced by any classical-client RSP protocol. Here, we assume we have a *correct* RSP protocol (except with negligible error), but we make *no assumption about the security* of this protocol.

6.4.1 Impossibility of Composable $\mathsf{UBQC}_{\mathsf{CC}}$ on 1 Qubit

In order to prove that there exists no $\mathsf{UBQC}_{\mathsf{CC}}$ protocol, we will first focus on the simpler case when the computation is described by a single measurement angle. The resource that performs a blind quantum computation on one qubit ($\mathcal{S}_{UBQC1}$) is defined as below, following the definition introduced in [DFP+14].

Definition 6.4.1 (Ideal resource of single-qubit UBQC [DFP+14])**.** *The definition of the ideal resource $\mathcal{S}_{UBQC1}$, depicted in Figure 6.4, achieves blind quantum computation specified by a single angle ϕ. The input (ξ, ρ) is filtered when $c = 0$. The ξ can be any deviation (specified for example using the classical description of a CPTP map) that outputs a classical bit, and which can depend on the computation angle ϕ and on some arbitrary quantum state ρ.*

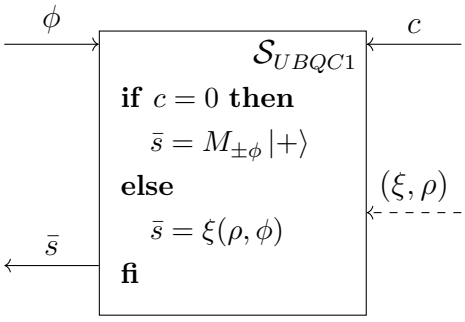


Figure 6.4: Ideal resource $\mathcal{S}_{UBQC1}$ for UBQC with one angle, with a filtered (dashed) input. In the case of honest server the output $\bar{s} \in \{0, 1\}$ is computed by measuring the qubits $|+\rangle$ in the $\{|+_\phi\rangle, |-_\phi\rangle\}$ basis. On the other hand if $c = 1$ any malicious behaviour of server can be captured by (ξ, ρ), i.e. the output $\bar{s}$ is computed by applying the CPTP map ξ on the input ϕ and on another auxiliary state ρ chosen by the server.

Theorem 6.4.2 (No-go composable classical-client single-qubit UBQC)**.** *Let (P_A, P_B) be a protocol interacting only through a classical channel $\mathcal{C}$, such that $(\theta, \rho_B) \leftarrow (P_A \mathcal{C} P_B)$ with $\theta \in \mathbb{Z}\frac{\pi}{4}$, and such that (by correctness) the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible[19] with overwhelming probability[20]. Then, if we define π_A and π_B as the UBQC_{CC} protocol on one qubit that makes use of (P_A, P_B) as a sub-protocol to replace the quantum channel (as pictured in Figure 6.5), (π_A, π_B) is not composable, i.e. there exists no simulator σ such that:*

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^{c=0} \tag{6.45}$$

$$\pi_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{6.46}$$

for some negligible $\varepsilon = \mathsf{negl}(\lambda)$.

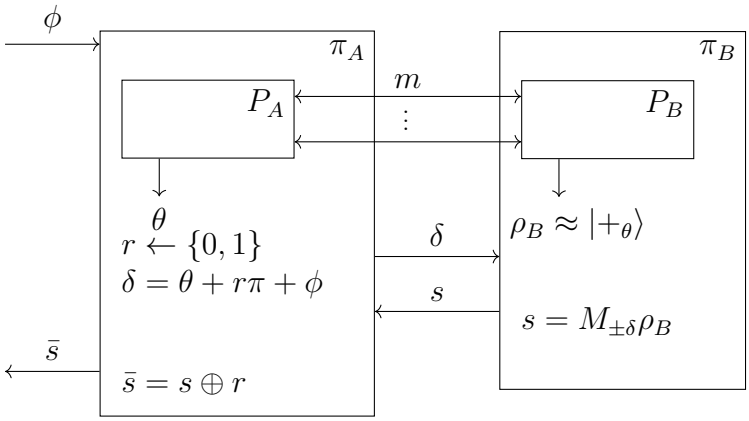


Figure 6.5: UBQC with one qubit when both Alice and Bob follows the protocol honestly (see Protocol 5)

Proof. The first proof we obtained for this theorem was quite long and hard to follow: we were basically studying all the possible strategies of σ, proving that none of them could lead to a secure protocol[21]: the last step in our proof was to say that if σ was able to trick the distinguisher, then σ would be able to be used to do an impossible measurement. However, we found later a much more elegant method to prove this theorem—that will be presented in the rest of this section—by remarking that the $\mathcal{S}_{UBQC1}$ resource can

[19] In the following, the parties P_A and P_B (and therefore π_A and π_B) and the simulator σ depend on some security parameter λ, but, in order to simplify the notations and the proof, this dependence will be implicit. We are as usual interested only in the asymptotic security, when $\lambda \to \infty$.

[20] Note that here ρ_B is different at every run: it corresponds to the density matrix of the state obtained after running P_B, when tracing out the environment and the internal registers of P_B and P_A.

[21] The proof has not been included in this thesis not to exhaust my reporters, but in case you are interested I can share the proof. Slightly more details are included in Section 6.6.

be seen as a RSP resource. We can therefore use the machinery introduced in our first impossibility result to treat this case.

In order to prove this theorem, we will proceed by contradiction. Let us assume that there exists (P_A, P_B), and a simulator σ having the above properties. Then, for the same resource $\mathcal{S}_{UBQC1}$ we will derive a different protocol $\boldsymbol{\pi}' = (\pi'_A, \pi'_B)$ that realizes it (the exact definition will be given later), but using a different filter[22] $\vdash^{\sigma}$ and a different simulator σ':

$$\pi'_A \mathcal{C} \pi'_B \approx_{\varepsilon} \mathcal{S}_{UBQC1} \vdash^{\sigma} \tag{6.47}$$

$$\pi'_A \mathcal{C} \approx_{\varepsilon} \mathcal{S}_{UBQC1} \sigma' \tag{6.48}$$

More specifically, the new filter $\vdash^{\sigma}_{UBQC1}$ will depend on σ defined in Eq. (6.46). Then our main proof can be described in the following steps:

1. We first show in Lemma 6.4.4 that $\mathcal{S}_{UBQC1}$ is also ε-classically-realizable by (π'_A, π'_B) with the filter $\vdash^{\sigma}$.

2. We then prove in Lemma 6.4.5 that the resource $\mathcal{S}_{UBQC1}$ is an RSP within $\mathsf{negl}(\lambda)$, with respect to some well chosen converters $\mathcal{A}$ and $\mathcal{Q}$ (see Figure 6.6) and this new filter $\vdash^{\sigma}$.

3. Then, we use the main result about RSP (Theorem 6.3.6) to show that $\mathcal{S}_{UBQC1}$ is describable within $\mathsf{negl}(\lambda)$ with respect to $\mathcal{A}$ (Corollary 6.4.6).

4. Finally, in Lemma 6.4.8 we prove that if $\mathcal{S}_{UBQC1}$ is describable then we could achieve *superluminal signaling*, which concludes the contradiction proof. □

Definition 6.4.3. *Let $\pi' = (\pi'_A, \pi'_B)$ be the protocol realizing $\mathcal{S}_{UBQC1}$ described in the following way (as pictured Figure 6.6):*

- *$\pi'_A = \pi_A$ (Figure 6.5)*
- *π'_B: runs P_B, obtains a state ρ_B, then uses the angle δ received from its inner interface to compute $\tilde{\rho} := R_Z(-\delta)\rho_B$, and finally outputs $\tilde{\rho}$ on its outer interface and $s := 0$ on its inner interface.*

Then we define $\vdash^{\sigma} = \sigma\pi'_B$ depicted in Figure 6.7 (with σ being the simulator from Eq. (6.46) above). We further let the converters $\mathcal{A}$ and $\mathcal{Q}$ be as described in Figure 6.6:

Lemma 6.4.4. *If $\mathcal{S}_{UBQC1}$ is ε-classically-realizable by (π_A, π_B) with the filter $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is also ε-classically-realizable by (π'_A, π'_B) with the filter $\vdash^{\sigma}$.*

[22] Note that we could include this new filter inside $\mathcal{S}_{UBQC1}$ and use a more traditional filter $\vdash^{c=0}$ but for simplicity we will just use a different filter.

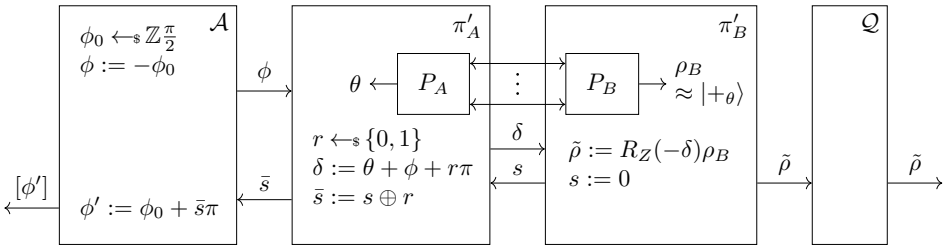


Figure 6.6: Definition of $\mathcal{A}$, π'_A, π'_B and $\mathcal{Q}$.

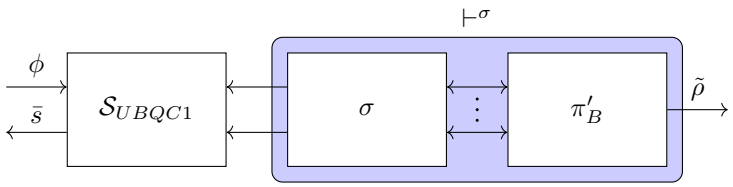


Figure 6.7: Description of $\vdash^\sigma$.

Proof. If S_{UBQC1} is ε-classically-realizable with $\vdash^{c=0}$, then as seen in Theorem 6.4.2, we have:

$$\pi_A \mathcal{C} \pi_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^{c=0} \tag{6.49}$$

$$\pi_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{6.50}$$

Now we can show that $\mathcal{S}_{UBQC1}$ is ε-classically-realizable by (π'_A, π'_B) with $\vdash^\sigma$, i.e. that there exists a simulator σ' such that:

$$\pi'_A \mathcal{C} \pi'_B \approx_\varepsilon \mathcal{S}_{UBQC1} \vdash^\sigma \tag{6.51}$$

$$\pi'_A \mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC1} \sigma' \tag{6.52}$$

For the correctness condition, we have:

$$\pi'_A \mathcal{C} \pi'_B = (\pi_A \mathcal{C}) \pi'_B \tag{6.53}$$

$$\approx_\varepsilon (\mathcal{S}_{UBQC1} \sigma) \pi'_B \tag{6.54}$$

$$= \mathcal{S}_{UBQC1} \vdash^\sigma \tag{6.55}$$

For the security condition, we define $\sigma' = \sigma$. Then, we have:

$$\pi'_A \mathcal{C} = \pi_A \mathcal{C} \tag{6.56}$$

$$\approx_\varepsilon \mathcal{S}_{UBQC1} \sigma \tag{6.57}$$

Which concludes our proof. □

Lemma 6.4.5. *If $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(\lambda)$-classically-realizable with $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is a $\mathsf{negl}(\lambda)$-*remote state preparation resource *with respect the converters $\mathcal{A}$ and $\mathcal{Q}$ and filter $\vdash^{\sigma}$ defined in Figure 6.6.*

Proof. We need to prove that:

$$\mathop{\mathbb{E}}_{([\rho],\rho_B)\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\vdash^{\sigma}\mathcal{Q}}[\,\mathrm{Tr}(\rho\rho_B)\,]\geq 1-\varepsilon \tag{6.58}$$

First, we remark that due to Lemma 6.4.4:

$$\mathcal{A}\mathcal{S}_{UBQC1}\vdash^{\sigma}\mathcal{Q}\approx_{\varepsilon}\mathcal{A}\pi'_A\mathcal{C}\pi'_B\mathcal{Q} \tag{6.59}$$

However, from the protocol description it is easy to check that in the real world $\bar{s}=0\oplus r=r$, and therefore $\phi':=\phi_0+\bar{s}\pi=\phi_0+r\pi$ and $\rho=|+_{\phi'}\rangle\langle+_{\phi'}|$. And because the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible with overwhelming probability (by the correctness of (P_A,P_B)), then we also have that $\tilde{\rho}=R_Z(-\delta)\rho_B R(-\delta)^\dagger$ is negligibly close in trace distance to $|+_{\theta-\delta}\rangle\langle+_{\theta-\delta}|=|+_{-\phi_0+r\pi}\rangle\langle+_{-\phi_0+r\pi}|=|+_{\phi'}\rangle\langle+_{\phi'}|$. Therefore, we have:

$$\mathop{\mathbb{E}}_{([\rho],\tilde{\rho})\leftarrow\mathcal{A}\pi'_A\mathcal{C}\pi'_B\mathcal{Q}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,]\geq 1-\mathsf{negl}(\lambda) \tag{6.60}$$

Then it also means that:

$$\mathop{\mathbb{E}}_{([\rho],\tilde{\rho})\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\vdash^{\sigma}\mathcal{Q}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,]\geq 1-\mathsf{negl}(\lambda) \tag{6.61}$$

otherwise we could (using a similar argument to the one given in the proof of Theorem 6.3.6) distinguish between the ideal and the real world, contradicting Eq. (6.59), which concludes the proof. □

Now, using our main Theorem 6.3.6 we obtain directly that if $\mathcal{S}_{UBQC1}$ is classically-realizable and RSP with respect to filter $\vdash^{\sigma}$, then it is also describable:

Corollary 6.4.6. *If $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(\lambda)$-classically-realizable with respect to filter $\vdash^{c=0}$ then $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(\lambda)$-describable with respect to the converter $\mathcal{A}$ described above.*

The problem of describability is that we only know that the state are close. However, in our reduction we need to say that the states are actually identical (so that we can transmit information faster than light). We prove now that if a resource is $\mathsf{negl}(\lambda)$-describable, then by "rounding" the classical description given by $\mathcal{P}$ to the nearest state that $\mathcal{A}$ can output, then we have with overwhelming probability the output of $\mathcal{A}$ and our new $\mathcal{P}$ are identical.

Lemma 6.4.7. *Let $\Omega = \{[\rho_i]\}$ be a set of (classical descriptions of) density matrices, such that $\forall i \neq j$, $\mathrm{Tr}(\rho_i\rho_j) \leq 1-\eta$. Then let $([\rho],[\tilde{\rho}])$ be two random variables (representing the classical description of the density matrices), such that $[\rho] \in \Omega$ and $\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1-\varepsilon$, with $\eta > 6\sqrt{\varepsilon}$. Then, if we define the following "rounding" operation that rounds $\tilde{\rho}$ to the closest $\tilde{\rho}_r \in \Omega$:*

$$[\tilde{\rho}_r] := \mathrm{Round}_\Omega([\tilde{\rho}]) := \arg\max_{[\tilde{\rho}_r]\in\Omega} \mathrm{Tr}(\tilde{\rho}_r\tilde{\rho}) \tag{6.62}$$

Then we have:

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1-\sqrt{\varepsilon} \tag{6.63}$$

In particular, if $\varepsilon = \mathsf{negl}(\lambda)$, and $\eta \neq 0$ is a constant, $\Pr[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \mathsf{negl}(\lambda)$.

Proof. We know that $\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}])}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1-\varepsilon$. Therefore, using Markov inequality we get that:

$$\Pr_{([\rho],[\tilde{\rho}])}\left[\,1-\mathrm{Tr}(\rho\tilde{\rho}) \geq \sqrt{\varepsilon}\,\right] \leq \frac{\mathbb{E}[\,1-\mathrm{Tr}(\rho\tilde{\rho})\,]}{\varepsilon} \tag{6.64}$$

$$\Pr_{([\rho],[\tilde{\rho}])}\left[\,\mathrm{Tr}(\rho\tilde{\rho}) \leq 1-\sqrt{\varepsilon}\,\right] \leq \frac{\varepsilon}{\sqrt{\varepsilon}} \tag{6.65}$$

$$\Pr_{([\rho],[\tilde{\rho}])}\left[\,\mathrm{Tr}(\rho\tilde{\rho}) \geq 1-\sqrt{\varepsilon}\,\right] \geq 1-\sqrt{\varepsilon} \tag{6.66}$$

But when $\mathrm{Tr}(\rho\tilde{\rho}) \geq 1-\sqrt{\varepsilon}$, we have $\mathrm{Round}_\Omega([\tilde{\rho}]) = \rho$.

> We will indeed show that $\forall \rho_i \in \Omega$, $\mathrm{Tr}(\rho_i\tilde{\rho}) \leq \mathrm{Tr}(\rho\tilde{\rho})$. By contradiction, we assume there exists $\rho_i \in \Omega$ such that $\rho_i \neq \rho$ and $\mathrm{Tr}(\rho_i\tilde{\rho}) > \mathrm{Tr}(\rho\tilde{\rho}) \geq 1-\sqrt{\varepsilon}$. But due to Lemma 6.5.4 we have:
>
> $$\mathrm{Tr}(\rho_i\rho) \geq 1-3(\sqrt{\varepsilon}+\sqrt{\varepsilon}) = 1-6\sqrt{\varepsilon} \tag{6.67}$$
>
> However, because both ρ_i and ρ belong to Ω, we also have $\mathrm{Tr}(\rho_i\rho) \leq 1-\eta < 1-6\sqrt{\varepsilon}$, which is absurd.

Therefore, using Eq. (6.66) we have

$$\Pr_{([\rho],[\tilde{\rho}])}[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1-\sqrt{\varepsilon} \tag{6.68}$$

which concludes the proof. □

We can now conclude our proof by remarking that describability implies signaling:

Lemma 6.4.8. *$\mathcal{S}_{UBQC1}$ cannot be $\mathsf{negl}(\lambda)$-describable with respect to converter $\mathcal{A}$.*

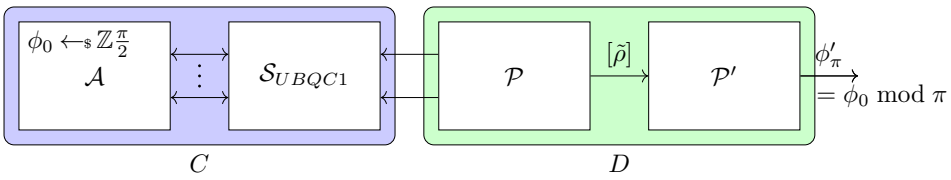


Figure 6.8: Illustration of the no-signaling argument.

Proof. If we assume that $\mathcal{S}_{UBQC1}$ is $\mathsf{negl}(\lambda)$-describable, then there exists a converter $\mathcal{P}$ (outputting $[\tilde{\rho}]$) such that:

$$\mathop{\mathbb{E}}_{([\rho],[\tilde{\rho}])\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\mathcal{P}}[\,\mathrm{Tr}(\rho\tilde{\rho})\,] \geq 1 - \mathsf{negl}(\lambda) \tag{6.69}$$

We define the set $\Omega := \{[|+_{\theta'}\rangle\langle+_{\theta'}|] \mid \theta' \in \{0, \pi/4, ..., 7\pi/4\}\}$. For simplicity, we will denote in the following $[\theta] = [|+_\theta\rangle\langle+_\theta|]$.

In the remaining of the proof, we are going to use the converters $\mathcal{A}$ and $\mathcal{P}$ together with the ideal resource $\mathcal{S}_{UBQC1}$, to construct a 2-party setting that would achieve signaling, which would end our contradiction proof. More specifically, we will define a converter D running on the right interface of $\mathcal{S}_{UBQC1}$ which will manage to recover the ϕ_0 chosen randomly by $\mathcal{A}$.
As shown in Figure 6.8, if we define C as $C := \mathcal{A}\mathcal{S}_{UBQC1}$ and D the converter described above, then the setting can be seen equivalently as: C chooses as random ϕ_0 and D needs to output $\phi_0 \bmod \pi$. This is however impossible, as no message is sent from $\mathcal{S}_{UBQC1}$ to its right interface (as seen in Figure 6.8) (and thus no message from C to D), and therefore guessing ϕ_0 is forbidden by the no-signaling principle [GRW80].

We define $\mathcal{P}'$ as the converter that, given $[\tilde{\rho}]$ from the outer interface of $\mathcal{P}$ computes $[\tilde{\phi}] = \mathrm{Round}_\Omega([\tilde{\rho}])$ and outputs $\tilde{\phi}_\pi = \tilde{\phi} \bmod \pi$ (as depicted in Figure 6.8). We will now prove that $\tilde{\phi}_\pi = \phi_0 \bmod \pi$ with overwhelming probability.

All elements in Ω are different pure states, and in finite number, so there exist a constant $\eta > 0$ respecting the first condition of Lemma 6.4.7. Moreover from Eq. (6.69) we have that $\mathcal{S}_{UBQC1}$ is ε-describable with $\varepsilon = \mathsf{negl}(\lambda)$, so we also have (for large enough n), $\eta > 6\sqrt{\varepsilon}$. Therefore, from Lemma 6.4.7, we have that:

$$\Pr_{([\rho],[\tilde{\rho}])\leftarrow\mathcal{A}\mathcal{S}_{UBQC1}\mathcal{P}}[\,\mathrm{Round}_\Omega([\tilde{\rho}]) = [\rho]\,] \geq 1 - \mathsf{negl}(\lambda) \tag{6.70}$$

But using the definition of converter $\mathcal{A}$, we have: $[\rho] = [\phi']$, where $\phi' = \phi_0 + \bar{s}\pi$, and hence $\phi' \bmod \pi = \phi_0 \bmod \pi$. Then, using the definition of $\mathcal{P}'$, the Eq. (6.70) is equivalent

to:

$$\Pr_{([\phi'],\tilde{\phi}_\pi)\leftarrow\mathcal{AS}_{UBQC1}\mathcal{PP}'}\left[\tilde{\phi}_\pi = \phi_0 \bmod \pi\right] \geq 1 - \mathsf{negl}(\lambda) \tag{6.71}$$

However, as pictured in Figure 6.8, this can be seen as a game between $C = \mathcal{AS}_{UBQC1}$ and $D = \mathcal{PP}'$, where, as explained before, C picks a $\phi_0 \in \mathbb{Z}_{\frac{\pi}{2}}$ randomly, and D needs to output $\phi_0 \bmod \pi$. From Eq. (6.71) D wins with overwhelming probability, however, we know that since there is no information transfer from C to D, the probability of winning this game with probability better than 1/2 (guessing the bit at random) would imply signaling, which is known to be impossible in quantum information. □

6.4.2 Impossibility of Composable $\mathsf{UBQC_{CC}}$ on Any Number of Qubits

We saw in Theorem 6.4.2 that it is not possible to implement a composable classical-client UBQC protocol performing a computation on a single qubit. In this section, we prove that this result generalizes to the impossibility of $\mathsf{UBQC_{CC}}$ on computations using an arbitrary number of qubits. The proof works by reducing the general case to the single-qubit case from the previous section.

Theorem 6.4.9 (No-go Composable Classical-Client UBQC)**.** *Let (P_A, P_B) be a protocol interacting only through a classical channel $\mathcal{C}$, such that $(\theta, \rho_B) \leftarrow (P_A\mathcal{C}P_B)$ with $\theta \in \mathbb{Z}_{\frac{\pi}{4}}$, and such that the trace distance between ρ_B and $|+_\theta\rangle\langle+_\theta|$ is negligible with overwhelming probability. Then, if we define (π_A^G, π_B^G) as the $\mathsf{UBQC_{CC}}$ protocol on any fixed graph G (with at least one output qubit[23]), that uses (P_A, P_B) as a sub-protocol to replace the quantum channel, (π_A^G, π_B^G) is not composable, i.e. there exists no simulator σ such that:*

$$\pi_A^G\mathcal{C}\pi_B^G \approx_\varepsilon \mathcal{S}_{UBQC} \vdash^{c=0} \tag{6.72}$$

$$\pi_A^G\mathcal{C} \approx_\varepsilon \mathcal{S}_{UBQC}\sigma \tag{6.73}$$

for some negligible $\varepsilon = \mathsf{negl}(\lambda)$, where $\mathcal{S}_{UBQC}$ is a trivial generalisation of $\mathcal{S}_{UBQC1}$ to multiple qubits (defined in [DFP+14] under the notation $\mathcal{S}^{blind}$) for which an additional leakage l^{ψ_A} is send to the server, which is (at least in our case) equal to the size of the graph state.

Proof. To prove this statement, we just need to prove that we can come back to the setting with a single qubit, where we want to perform a computation with angle ϕ, and

[23] Note, that in $\mathsf{UBQC_{CC}}$ with zero output qubits the client does not receive any results. Hence, the protocol is trivially implementable for this degenerated case.

output one angle close to ϕ as in the proof of Theorem 6.4.2. Because the graph has at least one output qubit, we will denote by ω the index of the last output qubit. So the idea is to let the distinguisher choose the client input such that for any node $i \neq \omega$ in the graph, $\phi_i = 0$, and for the output qubit, $\phi_\omega = \phi$. Moreover, on the server side, the distinguisher will behave like the honest protocol π_B^G, except that it will not entangle the qubits provided by P_A, and it will deviate on the output qubit ω by not measuring it and sending $s := 0$, the qubit being rotated again with angle $-\delta_\omega$, and outputted on the outer interface, like in the one-qubit case (see Figure 6.6). It is now easy to see by induction (over the index of the qubit, following the order chosen on G) that, in the real world, for all $i \neq \omega$, we always have $s_i = r_i$, therefore $\bar{s}_i = 0$. So for all nodes i, (including ω), $s_i^X = \oplus_{i \in D_i} \bar{s}_i = 0$ and $s_i^Z = \oplus_{i \in D_i'} \bar{s}_i = 0$. Thus we have on the last node:

$$\begin{aligned} \delta_\omega &= \theta_\omega + (-1)^{s_\omega^X} \phi_\omega + s_\omega^Z \pi + r_\omega \pi \\ &= \theta_\omega + \phi + r_\omega \pi \end{aligned}$$

which corresponds exactly to the single-qubit setting, shown to be impossible. □

6.5 Distance Measures for Quantum States

When justifying the choice of Definition 6.3.2, we gave two different interpretations, either saying that an RSP resource must be testable (i.e. $\mathrm{Tr}(\rho\sigma) \approx 1$, this is the choice used in the actual definition), or saying that a resource must be accurate and pure ($D_{TD}(\rho, \sigma) \approx 0$, $\mathrm{Tr}(\rho^2) \approx 1$ and $\mathrm{Tr}(\sigma^2) \approx 1$). In this section, we show that both interpretations are basically equivalent. First, we state this simple lemma which will be useful in the rest of the section:

Lemma 6.5.1. *For any two self-adjoint trace-class operators ρ, σ it holds that*

$$\mathrm{Tr}(\rho\sigma) = \frac{1}{2}\left[\mathrm{Tr}(\rho^2) + \mathrm{Tr}(\sigma^2)\right] - \frac{1}{2}\|\rho - \sigma\|_{HS}^2,$$

where the Hilbert-Schmidt norm is defined as

$$\|A\|_{HS} = \sqrt{\mathrm{Tr}(A^* A)}.$$

Proof. This follows directly from the relation

$$(\rho - \sigma)^2 = \rho^2 - \rho\sigma - \sigma\rho + \sigma^2$$

and the fact that ρ and σ are self-adjoint operators. □

The following lemma formalizes the following statement: If $\mathrm{Tr}(\rho\sigma)$ is close to 1, then both ρ and σ must be almost pure, and ρ and σ must be close to each other. Note that Lemma 6.5.2 holds in particular for density matrices ρ and σ, despite being stated for a more general class of operators.

Lemma 6.5.2. *Let $\varepsilon \geq 0$ and $\mathrm{Tr}(\rho\sigma) \geq 1 - \varepsilon$ for two self-adjoint, positive semi-definite operators ρ, σ with trace less than 1. Then, it holds that*

1. $\mathrm{Tr}\left(\rho^2\right) \geq 1 - 2\varepsilon$,
2. $\mathrm{Tr}\left(\sigma^2\right) \geq 1 - 2\varepsilon$, *and*
3. $\|\rho - \sigma\|_{HS} \leq \sqrt{2\varepsilon}$.

Proof. 1. With the formula from Lemma 6.5.1, we infer that

$$\mathrm{Tr}(\rho\sigma) \leq \frac{1}{2}\left[\mathrm{Tr}(\rho^2) + \mathrm{Tr}(\sigma^2)\right] \leq \frac{1}{2}\left[\mathrm{Tr}(\rho^2) + 1\right],$$

using the non-negativity of the Hilbert-Schmidt norm and the fact that $\mathrm{Tr}\left(\sigma^2\right) \leq 1$. Hence,

$$\mathrm{Tr}\left(\rho^2\right) \geq 2\,\mathrm{Tr}\left(\rho\sigma\right) - 1 \geq 1 - 2\varepsilon.$$

2. Analogously to 1.

3. Using $\mathrm{Tr}\left(\rho^2\right) \leq 1$ and $\mathrm{Tr}\left(\sigma^2\right) \leq 1$, we obtain

$$\begin{aligned} \mathrm{Tr}\left(\rho\sigma\right) &\leq 1 - \frac{1}{2}\|\rho - \sigma\|_{\mathrm{HS}}^2 \\ \Rightarrow \|\rho - \sigma\|_{\mathrm{HS}}^2 &\leq 2\left(1 - \mathrm{Tr}\left(\rho\sigma\right)\right) \leq 2\varepsilon, \end{aligned}$$

which implies the claim.

□

While the previous lemmas were first stated in term of the Hilbert-Schmidt norm, the trace distance is more meaningful since it is the natural distance between quantum states.

Lemma 6.5.3. *Let λ be a security parameter and let ρ, σ be two density matrices of finite and fixed dimension. Then, the following statements are equivalent:*

1. $\mathrm{Tr}\left(\rho^2\right) \geq 1 - \mathsf{negl}(\lambda)$, $\mathrm{Tr}\left(\sigma^2\right) \geq 1 - \mathsf{negl}(\lambda)$, *and* $D_{TD}(\rho, \sigma) \leq \mathsf{negl}(\lambda)$,

2. $\mathrm{Tr}\,(\rho\sigma) \geq 1 - \mathsf{negl}(\lambda)$,

where $D_{TD}(\rho, \sigma)$ *denotes the trace distance (Definition 2.3.2).*

Proof. One direction of the equivalence follows directly from Lemma 6.5.2. The other direction follows from the formula in Lemma 6.5.1 and the fact that in finite-dimensional spaces the trace norm is equivalent to the Hilbert-Schmidt norm. □

Note also that $\mathrm{Tr}(\rho\sigma)$ also benefits from a sort of triangle inequality.

Lemma 6.5.4. *Let* $\varepsilon_1, \varepsilon_2 \geq 0$. *Let further* $\mathrm{Tr}\,(\rho_1\rho_2) \geq 1 - \varepsilon_1$ *and* $\mathrm{Tr}\,(\rho_2\rho_3) \geq 1 - \varepsilon_2$ *for self-adjoint, positive semi-definite operators* ρ_1, ρ_2, ρ_3 *with trace less than 1. Then it holds that* $\mathrm{Tr}\,(\rho_1\rho_3) \geq 1 - 3\,(\varepsilon_1 + \varepsilon_2)$.

Proof. From Lemma 6.5.2 we know that $\mathrm{Tr}\left(\rho_1^2\right) \geq 1 - 2\varepsilon_1$, $\mathrm{Tr}\left(\rho_3^2\right) \geq 1 - 2\varepsilon_2$, and

$$\|\rho_1 - \rho_2\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_1}, \quad \|\rho_2 - \rho_3\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_2}.$$

By the triangle inequality for the Hilbert-Schmidt norm, it follows readily that

$$\|\rho_1 - \rho_3\|_{\mathrm{HS}} \leq \sqrt{2\varepsilon_1} + \sqrt{2\varepsilon_2}$$

and therefore

$$\|\rho_1 - \rho_3\|_{\mathrm{HS}}^2 \leq \left(\sqrt{2\varepsilon_1} + \sqrt{2\varepsilon_2}\right)^2 = 2\varepsilon_1 + 2\varepsilon_2 + 4\sqrt{\varepsilon_1}\sqrt{\varepsilon_2} \leq 4\,(\varepsilon_1 + \varepsilon_2)$$

where we applied the inequality of the geometric mean to obtain the last bound. Using the formula from Lemma 6.5.1, we then conclude that

$$\begin{aligned}\mathrm{Tr}\,(\rho_1\rho_3) &= \frac{1}{2}\left[\mathrm{Tr}(\rho_1^2) + \mathrm{Tr}(\rho_3^2)\right] - \frac{1}{2}\|\rho_1 - \rho_3\|_{\mathrm{HS}}^2 \\ &\geq \frac{1}{2}\left[1 - 2\varepsilon_1 + 1 - 2\varepsilon_2\right] - \frac{1}{2}4\,(\varepsilon_1 + \varepsilon_2) \geq 1 - 3\,(\varepsilon_1 + \varepsilon_2)\,.\end{aligned}$$

□

6.6 Discussions and Open Questions

In this chapter, we saw that there exist no RSP protocols secure in the generally composable Constructive Cryptography framework. We expect our method to also apply to other generally composable frameworks, like Quantum Universal Composability [Unr10] since both models are mostly equivalent for the two-party setting. There may be however multiple ways to obtain some sort of composability.

A first option may be to use additional assumptions. Note that our result rules out all "classical" assumptions like Common Reference Strings: the assumptions must fundamentally recreate a quantum channel between the simulator and the distinguisher. As already discussed in Remark 6.3.11, this is the path taken by [GV19]: they propose to use an additional assumption called "Measurement Buffer" that forces the server to externalize some operations. Unfortunately, only polynomial security has been obtained so far using this assumption. While this seems to be a quite fundamental limitation, there may be some tricky methods—potentially involving error correcting codes—to obtain superpolynomial security with the Measurement Buffer. However, I do think that Measurement Buffer is quite fundamentally bound to polynomial security: in order to recreate a quantum channel between the simulator and the server, the simulator needs to sometimes replace the state inside the Measurement Buffer with something coming from the ideal functionality (otherwise it is no better than a classical channel). But this is typically detectable by the server since it could decide to input known states into the measurement buffer and check if the measurement outcome are the expected ones, allowing him to distinguish the ideal world from the real world.

Continuing with additional assumptions, it could be interesting to see if the Quantum Random Oracle model could be used to obtain general composable security. As with the Measurement Buffer, this would also recreate a quantum link between the simulator and the server, but the issues inherent to the former may be bypassed in the Quantum Random Oracle model. Indeed, it could be much harder for the server to check if the Quantum Oracle output has been modified or not. However, the protocol should certainly be adapted to fit the Random Oracle model: in our case, the function h could for instance be replaced with a Random Oracle instead. That said, I have not tried to explore this path further.

Another possible path to explore would be to directly prove the security of larger protocols involving internally a classical RSP protocol. We already shown that UBQC-like protocols cannot be shown to be composably secure when a classical RSP protocol is used internally, but it does not rule out other schemes. Fundamentally, the reason why UBQC is not provably secure in a composable way is that the simulator should be able to guess precisely the deviations done by the server in order to be able to forward this information with enough precision to the ideal RSP resource[24]. However, the number of

[24]Our initial proof of impossibility for composable $\mathsf{UBQC}_{\mathsf{CC}}$ was actually using this point of view: we were considering the simulator and ideal resource as a black box, and we were studying the input/output probability distribution. By relying only on the correctness properties of the resource, we were able to find a set of constraints on this probability distribution. However, this set of constraint turns out to be

possible deviations is so high compared to the number of bits received by the simulator that the simulator cannot possibly guess enough information about the deviation, leading to an inevitably non-coherent output. However, if we can reduce the number of possible undetected deviations, the impossibility result may not apply anymore. In particular, if we consider the verifiable variant of UBQCcalled VBQC [FK17], any significant deviation of the server is detected: as a result, the simulator would not need to find *which* deviation has been performed, but *if* a deviation has been done. Proving the security of a classical-client VBQC protocol (with superpolynomial security and/or without a Measurement Buffer assumption) is however an open question.

Giving up on general composable security, we may also be able to obtain meaningful composable security by lowering our expectations. Notably, we may wonder whether general composability can be replaced with the sequential security offered notably by the standalone framework. However, the question of the sequential composable security of classical-client RSP protocols is also open.

incompatible with quantum mechanics as we were able to derive from them an impossible measurement using the simulator.

CHAPTER 7

Zero-Knowledge, Quantum States and Multi-Party Authorized GHZ

"I know that I know nothing."

— Socrates discovering NIZK

Due to the special no-cloning principle, quantum states appear to be very useful in cryptography. But this very same property also has drawbacks: when receiving a quantum state, it is nearly impossible for the receiver to efficiently check non-trivial properties on that state without destroying it. This allows a sender to send maliciously crafted states (potentially entangled with a larger system) without being detected.

To illustrate this, let us imagine the following simple goal. A receiver Bob would like to obtain a quantum state $|\psi\rangle$ sent by Alice and verify, without destroying that state, that this state belongs to some "*quantum language*", say the language composed of BB84 states (so $|\psi\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$). Since any direct measurement would destroy that state, a first solution could be to use a generic quantum secure multiparty computing protocol (QSMPC) [DNS12, DGJ+20] between the sender and the receiver in order to generate that state. However, these protocols are interactive and require at least 2 messages (depending on the number of users and on the complexity of the prepared state, the number of rounds can increase significantly). Therefore, the following question was left open:

Is it possible to receive via a single message a quantum state and test non-trivial properties on it non-destructively?

In this chapter, we will first see in Section 7.1 an overview of our method together with a presentation of our setup, in Section 7.3 we see how to prove complex properties on quantum states non-destructively and non-interactively, leading to the notation of *Non-Destructive and Non-Interactive Zero-Knowledge proofs on Quantum States (NIZKoQS)*. At the heart of our method are classical-client RSP protocols. We will then see in Section 7.4 how this idea can be useful to prepare "authorized GHZ states": this allows a source to share a GHZ to only a subset of the parties (this subset can be defined in many different ways as we will see) in such a way that nobody, not even the source can know who is part of the GHZ state. Notably, this can be useful to run protocols (including quantum secret sharing) between multiple parties in such a way that nobody knows who is participating in the protocol. Other applications are also imaginable like "Quantum Onion-Routing" as discussed in Section 7.1.3.

7.1 Quick Overview and Presentation of the Setup

7.1.1 NIZKoQS

Classical Zero-Knowledge. The first observation that we can do is that, given an arbitrary quantum state $|\psi\rangle$, it is impossible to extract information from it non-destructively. However, in classical cryptography there exists a well known method to check highly non-trivial statements on a classical-string without revealing anything beyond the fact that the statement is true: this is known as *Zero-Knowledge* (ZK). Even better, a prover can prove *non-interactively* to a verifier that a string s belongs to a given language $\mathcal{L}$ as soon as $\mathcal{L}$ is contained into NP[1].

The reader not familiar with ZK may find this counter-intuitive. But there exists a nice analogy to understand how ZK is possible. Let us consider the following problem: Alice would like to prove to Bob that she knows the solution to a given Sudoku[2], without revealing her solution to Bob. In order to convince Bob, Alice can follow the protocol described in Figure 7.1: after repeating this protocol enough times, Bob gets convinced that Alice knows the solution, but learns nothing about this solution. This idea can be extended to any problem in NP (a similar method can work for graph coloring which is NP-complete) and can be made non-interactive (the challenge is then generated using a

[1] NP is the set of languages for which we can verify easily (in polynomial time) using a secret witness that $s \in \mathcal{L}$. However, without this witness, it may be hard to show that $s \in \mathcal{L}$.

[2] The Sudoku is a puzzle game: given a 9×9 grid having some cells pre-filled with numbers between 1 and 9, the goal is to fill the rest of the grid in such a way that each number between 1 and 9 should appears exactly once in each line, column and in each of the 9 disjoint sub-grids of 3×3 contiguous cells.

random oracle: this is known as the Fiat-Shamir transformation [FS87]) allowing us to do *Non-Interactive Zero-Knowledge* proofs (NIZK).

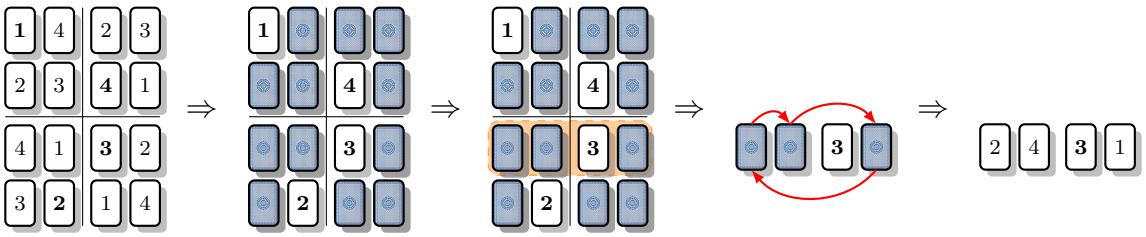


Figure 7.1: Illustration of ZK with a small sudoku (in bold). On the first image, Alice has a solution to the sudoku. Then, she hides this solution by flipping the cards (second image). Bob challenges Alice to reveal one line (in orange, third picture). Alice shuffles this line (fourth picture) and shows that it contains all numbers exactly once (last picture). Bob learns no information beside the fact that this line contains all numbers exactly once. Then, the process restarts from scratch: Bob can challenge other lines/columns/blocks until he is convinced that Alice really knows a solution to the sudoku.

NIZQoQS. The QFactory protocols we saw in Chapter 4 were initially designed to fake quantum channels with classical communication: the classical communication can be seen as a set of *classical instructions* producing some quantum states. But it turns out that classically faking a quantum channels also has a surprising side-effect as it "binds" a quantum state with a classical string. Since classical communication does not suffer from the no-cloning theorem, it is now possible to prove statements on the classical instructions—using your favorite classical NIZK protocol—to *indirectly* prove statements on the final produced quantum state. Of course, we need to trust the hardware of the receiver/verifier (here the server Bob) to be sure that the final quantum state matches the instructions. This method is therefore non-destructive and non-interactive: a single message from Alice to Bob is necessary to achieve NIZKoQS if the RSP protocol is itself non-interactive.

Remark 7.1.1. One may wonder why a single message is necessary since the QFactory protocols need two messages (a first message from Alice to Bob, and a second from Bob to Alice). However, the second message is only needed to ensure that Alice can learn the state obtained by Bob: this message is not necessary for Bob to be sure that the state he obtained has the appropriate properties. Similarly, in this chapter we will see protocols to share GHZ states that will use more that a single message: this is because these protocols do more than NIZK. The NIZK part will typically be witnessed after

the first message has been received, and the rest of the protocol will exploit the state produced during this first NIZK stage.

Remark 7.1.2. In the following we will focus on the GHZ-QFactory protocol. Note that the protocol of [Mah18a] can also be seen as a non-interactive RSP protocol, were the generated states are $\mathbf{X}^{\mathbf{a}}\mathbf{Z}^{\mathbf{b}}|\psi\rangle$ for some random one-time-pads $\mathbf{a}$ and $\mathbf{b}$, where $|\psi\rangle$ is determined by the encrypted circuit. Our NIZK approach is quite general and would also apply to [Mah18a] (or to our $\mathsf{UBQC}_{\mathsf{CC}}$ protocol if interaction is not an issue). However, for our applications (authorized GHZ preparation), GHZ-QFactory will be more efficient as it requires a single superposition instead of n as explained in Section 4.5: for this reason we will focus on GHZ-QFactory in this chapter.

Note also that some properties are not verifiable non-interactively: in particular, the proof that Alice sends cannot depend on the measurements done by Bob since Alice does not know them yet. However, we will see that there exists some non-trivial and interesting properties that can be verified on the obtained quantum state in a non-interactive manner: entanglement.

More specifically, if we consider the GHZ-QFactory protocol, the *classical* message that is sent to Bob is k, generated from $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$ where $\mathbf{d}_0 \in \{0,1\}^n$ corresponds to the support—i.e. basically the set of entangled qubits—of the GHZ state that will be produced by Bob. It is therefore possible to send a classical NIZK proof, together with k, proving that k has the "appropriate properties": Notably, we can prove (using t_k as a part of the witness) that *(i)* the message k is indeed a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable function[3] *(ii)* that $\mathsf{Auth}(\mathbf{d}_0, w) = 1$, where Auth can be any efficiently computable function, and w any secret witness depending on the wanted property on $\mathbf{d}_0(t_k)$. This last function and witness could be virtually anything, like ensuring that *There exists only two indices* $i \neq j$ *such that* $\mathbf{d}_0(i) = \mathbf{d}_0(j) = 1$ (i.e. it proves that the final state contains only two entangled qubits forming a Bell state, w is not needed here), or *Either* $\mathbf{d}_0[42] = 0$ *or I know the private key corresponding to the bitcoin wallet* *12c6DSiU4Rq3P4ZxziKxzrL5LmMBrzjrJX* (i.e. it proves that the 42th qubit is entangled to the rest of the GHZ only if the sender is Satoshi Nakamoto[4]... of course without revealing to the receiver if the sender is Satoshi Nakamoto; w being here the private key of the Bitcoin wallet of Satoshi). This kind of property will be particularly interesting in the multi-party $\mathtt{AUTH\text{-}BLIND}^{\mathtt{dist}}_{\mathtt{can}}$ protocol we will see later.

[3] In our construction, it boils down to proving that the trapdoor t_k has small enough singular values.

[4] Satoshi Nakamoto is the nickname used by the creator of Bitcoin. Nobody knows the real identity of Satoshi... and therefore Satoshi would certainly not participate in our protocol if there were a risk of revealing their identity.

That way, when receiving k and the NIZK proof, Bob can check that the proof is valid: when producing the quantum state, he will be sure that the properties on k and $\mathbf{d}_0$ are valid, and he will therefore obtain (assuming that his hardware is correct) indirectly (and therefore non-destructively) guarantees on the produced quantum state.

7.1.2 Authorized GHZ states

Setup. We can use the above ideas in a multi-party protocol: the setup that we consider is the following (note that we also derive below simpler protocols that may be of independent interest). A quantum server Bob (playing the role of a source) wants to share a GHZ state between n (weakly[5] quantum) *applicants*. However, the server wants to filter the applicants such that only a subset $\mathcal{S}$ of these applicants—the *supported*[6] applicants—share a part of the GHZ state. This subset can be determined in many different ways: for instance the supported applicants may correspond to the applicants knowing a secret password, a signature from a third party Certification Authority, the private key of a Bitcoin wallet owning more than one thousands Bitcoins... However, the applicants want to be sure that nobody (not even the server) should know whether or not they are supported (therefore, it is *not* possible to use a simple authentication step in which the source directly checks if the applicant knows the secret: we really need to bound this verification step with the production of the quantum state).

Assumptions. As in the GHZ-QFactory we require the existence of a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family. However, we also require a few more properties: More precisely, we will require the existence of a local generation procedure $(k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$ such that the public key of the δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family is obtainable via $k = (k^{(1)}, \dots, k^{(n)})$ (since $k^{(i)}$ is enough to fix $\mathbf{d}_0[i]$, we may write $\mathbf{d}_0[i] \coloneqq \mathbf{d}_0(k^{(i)})$). Moreover $t_k^{(i)}$ can be used to obtain partial information about the preimages of f_k: More precisely, given $t_k^{(i)}$ and $f_k(x)$ one can obtain the i-th bit of $h(x)$ and given $t_k^{(i)}$, b and $y = f_k(x) = f_k(x')$, we can obtain $\alpha^{(i)}$ such that $\oplus_i \alpha^{(i)} = \alpha \coloneqq \bigoplus_i b_i(x_i \oplus x'_i)$.

In practice, one can obtain a distributable δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable family from a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family (which has an additional property that the two preimages have the form $x = (0, \bar{x})$ and $x' = (1, \bar{x}')$). The idea is to sample one δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable function for

[5] The applicants need only basics quantum skills: depending on the protocol they may have nothing to do except receiving a state, or may need to apply a few gates.

[6] We call them *supported* because they are part of the support of the hidden GHZ state. We may also refer to this as being the *support status* of an applicant.

each $\mathbf{d}_0[i]$. Then, we can define our new function to be $f_{k^{(1)},\dots,k^{(n)}}(c, \bar{x}^{(1)}, \dots, \bar{x}^{(n)}) = f_{k^{(1)}}(c, \bar{x}^{(1)})|\dots|f_{k^{(n)}}(c, \bar{x}^{(n)})$. More details can be found in Section 7.4.3.

Protocol. In order to achieve the above protocol (called $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$), each applicant will be asked to sample $(k^{(i)}, t_k^{(i)}) \leftarrow \texttt{Gen}_{\texttt{Loc}}(1^\lambda, \mathbf{d}_0)$. Then $k^{(i)}$ will be sent to the server. In order to prove that they are authorized, each applicant will also include a NIZK proof confirming, as explained above, that they know a classical witness w_i such that $\mathsf{Auth}(\mathbf{d}_0[i], w_i) = 1$ (the NIZK proof also checks that $\mathbf{d}_0[i] = \mathbf{d}_0(k^{(i)})$ and that $k^{(i)} \in \mathcal{K}$). Since this protocol is a Zero-Knowledge protocol, the server will not be able to learn any information about $\mathbf{d}_0$. In return, the server will have the guarantee that it will indeed produce a hidden GHZ state whose support corresponds to the set of authorized applicants. Therefore, the server can run the quantum circuit used in the GHZ-QFactory protocol (Protocol 2), and distribute each qubit to the corresponding applicant. Non-supported applicants will just get a random non-entangled state that they can discard. Supported applicants will share a GHZ state, up to local $\mathbf{X}$ or $\mathbf{Z}$ corrections: In order to come back to a canonical GHZ, each applicant will use their local trapdoor $t_k^{(i)}$ to compute $h(x)$ so that they can apply the corresponding $\mathbf{X}$ correction. Moreover, in order to compute the α needed to apply the $\mathbf{Z}$ correction, all parties will need to run a Multi-Party Computation (MPC). The reason is that α could leak some information about the state: therefore the MPC will instead provide to each supported applicant a linear secret sharing of α, i.e. a random $\hat{\alpha}_i$ such that $\oplus_{i\in\mathcal{S}}\hat{\alpha}_i = \alpha$ ($\mathcal{S}$ being the set of supported applicants). That way, each supported applicant will be able to locally apply a $Z^{\hat{\alpha}_i}$ correction, and it will be equivalent to applying a single Z^α gate on the overall state.

Other Protocols. We also derive other simpler protocols of potential independent interest (they do not require MPC or NIZK, and are useful in our proposed Quantum Onion-like Routing protocol) in which an additional classical and honest third party, *Cupid*[7], is in charge of choosing the set of supported applicants. We assume Cupid can communicate classically with the server, as well as with all applicants using private channels[8].

Note also that these parties may not be always different entities. For example, when a user wants to send a qubit to a secret recipient, this user could be both considered as an applicant and as Cupid. Similarly, the server may want to be part of the applicants.

[7]Besides having a name starting with a 'C', Cupid, the roman god of love, is famous for sending arrows at the heart of Humans to designate the beloved among the applicants.

[8]In practice, secure authenticated channels would be enough if we ensure the length of all exchanged messages is the same.

We propose then 3 simpler protocols (BLIND, $\texttt{BLIND}^{\texttt{sup}}$ and $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$) similar to $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$, and we also show in Section 7.4.2.3 the impossibility of a desirable variant of these protocols, $\texttt{BLIND}_{\texttt{can}}$. In essence, they are all based on GHZ-QFactory, except that we leak more or less bits of information about the generated state (we are then interested in the leakage incurred by this additional information). More precisely, all these protocols derive from the BLIND protocol, in which Cupid chooses the support status of each applicant, and at the end of the protocol each supported applicant is supposed to obtain a generalized GHZ state (i.e. a GHZ state in which we applied some local X or Z gates), while non-supported applicant obtain random not entangled qubits in the computational basis (at that step no applicant know if they are supported or not). The other protocols differ slightly:

- The subscript $\cdot_{\texttt{can}}$ denotes the fact that at the end of the protocol each supported applicant ends up with a canonical GHZ state instead of a generalized GHZ state.
- The superscript $\cdot^{\texttt{sup}}$ denotes the fact that at the end of the protocol each applicant knows their own support status, chosen by Cupid.

In term of security, we typically expect that no malicious group of applicants, potentially colluding with the server, should learn the support status of honest applicants[9]. In BLIND, we can even prove that applicants cannot even learn their own support status.

7.1.3 Applications

The GHZ state (and its special case, the Bell pair) is very popular, and appears to be useful in many protocols, such as in Quantum Secret Sharing [HBB99], Quantum Teleportation [BBC+93], Entanglement Distillation [BBP+96a, BBP+96b, BDS+96], Device-Independent Quantum-Key-Distribution [MY98], Anonymous Transmission [CW05], Quantum Routing [PWD18, MMG19]... In these protocols, a (potentially untrusted) source is in charge of distributing GHZ states, which are then used differently in the protocol.

Our protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ can be used in the aforementioned protocols: instead of sending a GHZ state, the source—playing the role of the server—can run the $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ protocol with the clients—replacing the applicants—before the start of the protocol in order to generate a hidden GHZ state. Then, this resulting state can be distributed in place of the original GHZ state. The interest is to still obtain statistical

[9] In $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ and $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$, we expect at least one supported applicant to be honest when the adversary is allowed to corrupt supported applicants (the identity of this honest applicant may be unknown to the adversary), otherwise there is a trivial attack against any such protocol.

security coming from the quantum protocols, but also obtain additional computational guarantees: the source can filter the participants (like "only billionaires are allowed to participate") without learning their identity. Of course, a proper security analysis should be done for each protocol, notably if the source can collaborate with some applicants. If the source cannot collaborate with the applicants nor see the exchanged messages, then the security is more direct: any attack done in our "extended" protocol could also be done in the original protocol by just asking to the source to simulate the generation of the cryptographic assumptions.

For instance, a simple application of the $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ protocol would be to allow Bob to teleport a quantum state $|\psi\rangle$ to an unknown applicant knowing only its public key (for instance corresponding to a Bitcoin wallet). The applicant would be allowed to hide its identity to Bob, and Bob can be sure that only applicants knowing the private key of the wallet could obtain the state $|\psi\rangle$. In addition, if several applicants know the secret, then Bob is in fact secret sharing its qubit $|\psi\rangle$ among all applicants knowing this secret [HBB99].

Our protocol $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ could also be used to achieve new functionalities, such as a *Quantum Onion-like Routing* in order to route a quantum message through an untrusted quantum network (relying on a classical onion routing infrastructure), hiding the exact taken route. The idea would be to ask to each intermediate server (node) in the path of the message to blindly generate a large state in which a Bell pair is hidden (the first half of the Bell pair can be at a fixed position kept by the sender, and the other half would be randomly placed among the potential receivers) and to share this state with the neighbor nodes: this Bell pair could then be used to teleport the qubit to the next node, without revealing its identity to the previous node. One may have two objections against this protocol:

- One may think that we could achieve something similar by simply asking to the sender to directly send a state in which the Bell pair is hidden to the first server. However, this approach scale exponentially with the depth of the route: if we do not want to reveal at all to an intermediate node the identity of the next node, we also need to send the hidden Bell pair for the next node, together with dummies for the nodes that will not be taken. When doing this approach recursively, the number of qubits to send scales exponentially with the depth of the route.
- One may say that it may not be necessary to hide to each node the next node in the route (classically this is not the case). However, we can argue that a quantum network will be likely to be much smaller and centralized than a classical network (with maybe only a few quantum routers), and it would certainly be easier to find

a path in the classical network which is not controlled by the adversary than a path in the quantum network.

7.2 Introduction to Classical Zero-Knowledge and Multiparty Computing

In this section, we introduce classical Zero-Knowledge and Multiparty Computing as we used them internally in our approaches.

7.2.1 Classical Zero-Knowledge proofs and arguments for NP

In order to obtain NIZKoQS, we need to rely on a classical Zero-Knowledge (ZK) protocol for NP (to obtain NIZKoQS, we also expect the protocol to be non-interactive[10], but interactivity does not change security or correctness). Intuitively, in a ZK protocol for a language $\mathcal{L} \in \mathsf{NP}$, a prover must convince a verifier that a word x belongs to $\mathcal{L}$, in such a way that the verifier should not learn anything more about x beyond the fact that x belongs to $\mathcal{L}$. Because $\mathcal{L}$ is in NP, $\mathcal{L}$ is described by a relation $\mathcal{R}_{\mathcal{L}}$, in such a way that a word x belongs to $\mathcal{L}$ iff there exists a witness w such that $w \in \mathcal{R}_{\mathcal{L}}(x)$. Moreover, deciding if a witness w belongs to $\mathcal{R}_{\mathcal{L}}(x)$ must be doable in polynomial time.

We will now formalize the above security statements, taking the definition from [BS20]. Note that an honest verifier V outputs a single bit (1 if they accept and 0 if they reject), but a malicious verifier V^* can output an arbitrary quantum state.

Definition 7.2.1 (Computational indistinguishability [BS20])**.** *Two maps of quantum random variables $X := \{X_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ and $Y := \{Y_i\}_{\lambda\in\mathbb{N},i\in I_\lambda}$ over the same set of indices $I = \cup_{\lambda\in\mathbb{N}} I_\lambda$ are said to be* computationally indistinguishable*, denoted by $X \approx_c Y$, if for any non-uniform quantum polynomial-time distinguisher $D := \{(D_\lambda, \rho_\lambda)\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$, $i \in I_\lambda$,*

$$|\Pr[D_\lambda(X_i, \rho_\lambda) = 1] - \Pr[D_\lambda(Y_i, \rho_\lambda) = 1]| \leq \mu(\lambda) \tag{7.1}$$

Definition 7.2.2 (Post-Quantum Zero-Knowledge Classical Protocol [BS20, definitions 2.1 and 2.6])**.** *Let (P, V) be a protocol between an honest PPT prover P and an honest*

[10] Internally, we use in a blackbox manner an arbitrary classical NIZK protocol: therefore, even if we don't rely directly on the Random Oracle Model (ROM) or on the Common Reference String model (CRS), additional assumptions will be required depending on the assumptions used internally by the chosen NIZK. However, our protocol does not need any additional assumptions besides these ones and the hardness of LWE.

PPT verifier V*. Then* (P, V) *is said to be a Post-Quantum Zero-Knowledge (ZK) Classical Protocol for a language* $\mathcal{L} \in \mathsf{NP}$ *if the following properties are respected:*

1. ***Perfect Completeness****: For any* $\lambda \in \mathbb{N}$*,* $x \in \mathcal{L} \cap \{0,1\}^\lambda, w \in \mathcal{R}_\mathcal{L}(x)$*,*

$$\Pr\left[\,\mathsf{OUT}_\mathsf{V}(\mathsf{P}(w,x) \leftrightsquigarrow \mathsf{V}(x)) = 1\,\right] = 1 \tag{7.2}$$

2. ***Soundness****: The protocol satisfies one of the following.*
   - ***Computational Soundness****: For any non-uniform QPT malicious prover* $\mathsf{P}^* = \{(\mathsf{P}^*_\lambda, \rho_\lambda)\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $\mu(\cdot)$ *such that for any security parameter* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,*

$$\Pr\left[\,\mathsf{OUT}_\mathsf{V}(\mathsf{P}^*_\lambda(\rho_\lambda, x) \leftrightsquigarrow \mathsf{V}(x)) = 1\,\right] \leq \mu(\lambda) \tag{7.3}$$

     A protocol with computational soundness is called an argument.
   - ***Statistical Soundness****: There exists a negligible function* $\mu(\cdot)$ *such that for any unbounded prover* P^**, any security parameter* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,*

$$\Pr\left[\,\mathsf{OUT}_\mathsf{V}(\mathsf{P}^*(x) \leftrightsquigarrow \mathsf{V}(x)) = 1\,\right] \leq \mu(\lambda) \tag{7.4}$$

     A protocol with statistical soundness is called a proof.

3. ***Quantum Zero Knowledge****: There exists a QPT simulator* Sim *such that for any QPT verifier* $\mathsf{V}^* = \{(\mathsf{V}^*_\lambda, \rho_\lambda)\}_{\lambda\in\mathbb{N}}$*,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}(\mathsf{P}(w,x) \leftrightsquigarrow \mathsf{V}^*_\lambda(\rho_\lambda, x))\}_{\lambda,x,w} \approx_c \{\mathsf{Sim}(x, \mathsf{V}^*_\lambda, \rho_\lambda)\}_{\lambda,x,w} \tag{7.5}$$

   where $\lambda \in \mathbb{N}$*,* $x \in \mathcal{L} \cup \{0,1\}^\lambda$*,* $w \in \mathcal{R}_\mathcal{L}(x)$*, and* V^* *is given to* Sim *by sending the circuit description of* V^**.*

A Non-Interactive ZK protocol will be denoted NIZK.

In our last protocol, in order to get stronger guarantees, we may also want to ensure that the prover "knows" the secret (note that this property is not used extensively besides the fact that it provides more guarantees: therefore the reader may skip this part). Therefore, to get stronger guarantees, we will require the ZK protocol to also be a Proof of Knowledge protocol. Intuitively, we would like to check that any malicious prover P^* that can convince a verifier with a non-negligible probability has the witness w "encoded in its source code or memory". We formalize this notion by saying that there exist a QPT

circuit K, the extractor, which can recover w with non-negligible probability from a full description of P^* and its input.

This non-negligible probability is usually enough, since being able to obtain a witness with non-negligible probability is usually sufficient to break the security: for example it could be used to forge a signature and break the unforgeability property as explained in [Unr12].

Definition 7.2.3 (Post-Quantum Zero-Knowledge Proof of Knowledge [Unr12])**.** *We say that a Post-Quantum Zero-Knowledge protocol (P, V) for a relation $\mathcal{R}_{\mathcal{L}}$ is a* Proof of Knowledge *protocol, if it is* quantum extractable *with knowledge error $\kappa = \mathsf{negl}(\lambda)$, i.e. if there exists a constant $d > 0$, a polynomially-bounded function $p > 0$, and a QPT K such that for any interactive QPT malicious prover P^*, any polynomial l, any security parameter $\lambda \in \mathbb{N}$, any state ρ, and any $x \in \{0,1\}^\lambda$, we have:*

$$\Pr\left[\mathsf{OUT}_V(P^*(\rho, x) \leftrightsquigarrow V(x)) = 1\right] \geq \kappa(\lambda)$$
$$\implies \Pr\left[w \in \mathcal{R}_{\mathcal{L}}(x) \mid w \leftarrow K(P^*, \rho, x)\right] \geq \frac{1}{p(\lambda)}\left(\Pr\left[\mathsf{OUT}_V(P^*(\rho) \leftrightsquigarrow V(x)) = 1\right] - \kappa(\lambda)\right)^d$$

Several Post-Quantum ZK protocols have been proposed in the literature [Wat09, Unr12, BS20] and have been shown to obey properties similar to both Definitions 7.2.2 and 7.2.3. Moreover, [LZ19, DFM+19] explain how to obtain quantum-secure NIZK (which are also Proof of Knowledge) using the Fiat-Shamir transformation and the hardness of the LWE problem in a Quantum Random Oracle model. In the following, we are agnostic of the used NIZK protocol and we assume the existence of a NIZK protocol obeying Definitions 7.2.2 and 7.2.3.

7.2.2 Classical Multi-Party Computations

In the $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ protocol, we also need to use a classical[11] Multi-Party Computation (MPC) protocol Π. A MPC protocol works as follows: given n (public and deterministic) functions $(f_1, \ldots, f_n)$, at the end of the protocol involving n parties $\mathsf{P}_1, \ldots, \mathsf{P}_n$, we expect party P_i to get $f_i(x_1, \ldots, x_n)$, where x_j is the (secret) input of the party P_j. Moreover, we expect that no party can learn anything more than what they can already learn from $f_i(x_1, \ldots, x_n)$. For simplicity, we define $f(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$.

[11] But of course post-quantum secure.

We can formalize the above security statements using the usual (quantum) real/ideal world paradigm (this is similar to Constructive Cryptography except that here we have only sequential security: this is know as the *standalone* model).

Informally, the protocol Π will be said to be secure if it is impossible to distinguish two "worlds". On the one hand, we have a *real world* in which an adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ can corrupt a subset $\mathcal{M} \subsetneq [n]$ of parties and interact in an arbitrary way with the other honest parties. On the other hand, we have an *ideal world*, in which a simulator Sim interacts with a functionality, this functionality behaving as a trivially-secure trusted third-party. If these two worlds are indistinguishable, it means that the protocol is "secure" because any secret obtained from the real world would also be obtainable from the ideal world... which is impossible because the ideal world is trivially secure.

More precisely, we define, following [ABG+21], the real and ideal world as follows, where $\vec{x} \coloneqq (x_1, \dots, x_n)$ is the inputs of the parties:

Definition 7.2.4 ($\mathsf{REAL}_{\Pi,\mathcal{A}}(\lambda, \vec{x}, \rho_\lambda)$)**.** *$\mathcal{A}_\lambda$ is given ρ_λ, and gives a subset $\mathcal{M} \subsetneq [n]$ of corrupted (malicious) parties. Then $\mathcal{A}_\lambda$ receives the inputs x_i of all corrupted parties P_i ($i \in \mathcal{M}$), sends and receive all the messages on the behalf of these corrupted parties, and communicates in an arbitrary quantum polynomial time way with the honest parties that follow the protocol Π. At the end of the interaction, $\mathcal{A}_\lambda$ outputs an arbitrary state ρ, and we define as $\vec{y}$ the output of the honest parties P_j, $j \notin \mathcal{M}$. Finally, we define $\mathsf{REAL}_{\Pi,\mathcal{A}}(\lambda, \vec{x}, \rho_\lambda)$ as the random variable corresponding to $(\rho, \vec{y})$.*

Definition 7.2.5 ($\mathsf{IDEAL}_{f,\mathsf{Sim}}(\lambda, \vec{x}, \rho_\lambda)$)**.** Sim *(playing the role of the adversary) receives ρ_λ, outputs a set $\mathcal{M} \subsetneq [n]$ of corrupted parties, interacts with a trusted party (called the ideal functionality) defined below, and outputs at the end a state ρ. The ideal functionality also outputs at the end a message $\vec{y}$ corresponding to the output of the trusted party. We then define $\mathsf{IDEAL}_{f,\mathsf{Sim}}(\lambda, \vec{x}, \rho_\lambda)$ as the random variable corresponding to $(\rho, \vec{y})$. Now we define the ideal functionality:*

- *The ideal functionality receives the set $\mathcal{M} \subsetneq [n]$ a subset of corrupted parties, and for each party P_i, it receives an input x'_i: if P_i is honest ($i \notin \mathcal{M}$), we have $x'_i = x_i$, otherwise x'_i can be arbitrary.*
- *Then, it computes $(y_1, \dots, y_n) \coloneqq f(x'_1, \dots, x'_n)$, and sends $\{(i, y_i)_{i\in\mathcal{M}}\}$ to the simulator.*
- *The simulator can choose to abort by sending a message $\bot$ to the ideal functionality. Otherwise it sends a "continue" message $\top$. If the message received by the ideal functionality is $\bot$, then it outputs $\bot$ to each honest party, which is formalized by outputting $\vec{y} \coloneqq \{(i, \bot)\}_{i\notin\mathcal{M}}$. Otherwise, it outputs $\vec{y} \coloneqq \{(i, y_i)\}_{i\notin\mathcal{M}}$.*

Definition 7.2.6 (Secure MPC [ABG+21])**.** *Let f be a deterministic function with n inputs and n outputs, and Π be an n-party protocol. Protocol Π securely computes f if for every non-uniform quantum polynomial-time adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ corrupting a set of at most $n-1$ players, there exists a non-uniform quantum polynomial-time ideal-world adversary Sim such that for any combination of inputs $\vec{x} \in (\{0,1\}^*)^n$ and any non-uniform quantum advice $\rho = \{\rho_\lambda\}_{\lambda\in\mathbb{N}}$,*

$$\{\mathsf{REAL}_{\Pi,\mathcal{A}}(\lambda,\vec{x},\rho_\lambda)\}_{\lambda\in\mathbb{N}} \approx_c \{\mathsf{IDEAL}_{f,\mathsf{Sim}}(\lambda,\vec{x},\rho_\lambda)\}_{\lambda\in\mathbb{N}} \tag{7.6}$$

7.3 Non-Interactive Zero-Knowledge Proofs on Quantum States

In this section we first define our new concept of Non-Interactive and Non-Destructive Zero-Knowledge proofs on Quantum States (NIZKoQS), and define a protocol achieving NIZKoQS. The more involved protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ defined in the next section will also exploits NIZKoQS (but this protocol will have more than one message as it is also consuming the state produced by the NIZKoQS), while the other simpler protocols will only rely on the correctness (or completeness) of the following NIZKoQS protocol. Before giving the formal definition, let us motivate and describe informally the definition.

7.3.1 Intuitive motivation

The formal definition will be defined in the next section, but we first motivate informally our definition here.

Quantum language. In classical NIZK, a language $\mathcal{L}$ is a set of strings, so similarly we will define a quantum language $\mathcal{L}_\mathcal{Q}$ as a set of quantum states. For instance, we could consider:

- the quantum language made of BB84 states $\mathcal{L}_\mathcal{Q}^{\text{BB84}} = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$,
- the quantum language $\mathcal{L}_\mathcal{Q}^{ex} = \{|\mathbf{d}\rangle \pm |\mathbf{d}\oplus\mathbf{d}_0\rangle \mid (\mathbf{d},\mathbf{d}_0) \in (\{0,1\}^n)^2, \mathbf{d}_0[1] = 1, w_H(\mathbf{d}_0) = 2\}$ (w_H denotes the Hamming weight), corresponding to all hidden GHZ states whose first qubit is supported and where the support has size 2 (i.e. only two qubits are entangled forming a Bell state, where one of these qubits is at the first position),
- but we can also consider quantum languages referring to classical secrets, for instance if p_k is a public key (say of a bitcoin wallet), and if $\texttt{Ver}_{p_k}(s_k) = 1$ iff s_k is the

private key of p_k, we can define $\mathcal{L}_{\mathcal{Q}}^{p_k} = \{|\mathbf{d}\rangle \pm |\mathbf{d} \oplus \mathbf{d}_0\rangle \mid (\mathbf{d}, \mathbf{d}_0) \in (\{0,1\}^n)^2, \mathbf{d}_0[1] = 0 \vee (\mathbf{d}_0[1] = 1 \wedge \exists s_k, \mathtt{Ver}_{p_k}(s_k) = 1)\}$ that informally allows the prover to "send" a hidden GHZ state where the first qubit is supported *only* if the prover knows the private key of p_k.

Classically, both the prover and the verifier typically have a copy of the word x, and since information can be copied classically, the verification process cannot "destroy" x. Quantumly, this is not true anymore, therefore, instead of saying that all parties agree on ρ before the protocol, what matters is that *at the end* of the protocol, the verifier should end up with a $\rho \in \mathcal{L}_{\mathcal{Q}}$.

Relation, witness and quantum ZK. Classically, to check if a word x belongs to a language $\mathcal{L}$, we usually define a relation $\mathcal{R}$ between a *witness* w and the word, saying that $x \in L \Leftrightarrow \exists w, w\mathcal{R}x$. The prover typically knows the witness w and the ZK property ensures that the verifier has no way to learn the witness w, formalizing the fact that the verifier learns nothing beyond the fact that the statement is true. Quantumly, we mimic this definition by defining a relation between classical *witnesses* (or *classes*[12]) ω and quantum states $\mathcal{R}$, saying that a quantum state ρ belongs to a quantum language $\mathcal{L}_{\mathcal{Q}}$ if and only if there exists ω such that $\omega\mathcal{R}\rho$. Similarly, we want to ensure that the verifier has no way to learn ω.

However, even if our definition does not say anything more about witnesses, we need to choose them appropriately to obtain a meaningful and secure protocol. Moreover, at that stage it may not even be clear what could be used as a witness. For instance, in the quantum language $\mathcal{L}_{\mathcal{Q}}^{\mathrm{BB84}}$ defined above, what would be the witness of $|0\rangle$? Because the ZK property ensures that no information leaks about the witness, while we typically want to ensure that no information is leaked about the received state, one could naively say that the witness is the classical description of the state. Unfortunately if each witness ω_ρ is in a 1-to-1 correspondence to its corresponding state $\rho \in \mathcal{L}_{\mathcal{Q}}$, then it would be impossible to obtain the ZK property: for instance, given a random state in $\mathcal{L}_{\mathcal{Q}}^{\mathrm{BB84}} = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, it is possible to rule out one of the 4 states: for instance by measuring the state in the computational basis, if we measure b we know that the state can't be $|1-b\rangle$ and therefore we know that the witness can't be $\omega_{|1-b\rangle}$, contradicting the ZK property.

[12] Unlike in classical ZK, the witness ω cannot be used to verify that a quantum state belongs to the quantum language, for this reason the term *class* may be more appropriate. This also justifies the usage of a different notation ω instead of w.

To overcome this issue, a single witness ω must characterize a *class* of states. For instance, for the language $\mathcal{L}_{\mathcal{Q}}^{\mathrm{BB84}}$ we will define two witnesses 0 and 1 characterizing the basis of the state and we therefore define the relation $0\mathcal{R}\left|0\right\rangle$, $0\mathcal{R}\left|1\right\rangle$, $1\mathcal{R}\left|+\right\rangle$ and $1\mathcal{R}\left|-\right\rangle$. These classes will be used to characterize two of the three wanted properties:

- **Completeness** (or correctness): An honest prover should be able to choose ω and generate on the side of the verifier a state in $\mathcal{L}_\omega \coloneqq \{\rho \mid \omega\mathcal{R}\rho\}$.
- **Zero-Knowledge**: A malicious verifier should be unable to learn the witness ω with significant advantage over a random guess. Because of the completeness property, the verifier should therefore be unable to learn the class $\mathcal{L}_\omega$ chosen by the prover that was supposed to contain the target state.
- **Soundness**: Finally we also expect that if the prover is malicious, then an honest verifier will obtain a state in $\mathcal{L}$ whenever it accepts. (Note that this property does not depend on $\mathcal{R}$)

From the above properties, it clearly appears that when the relation $\mathcal{R}$ is thinner (i.e. when $|\mathcal{L}_\omega|$'s are smaller and the number of witnesses increases), we get a stronger result: indeed, an honest prover can choose more precisely the sent state and a malicious verifier is more confused as there are more classes to which a state could belong to. In particular, it is always possible to define a trivial NIZKoQS protocol for any language $\mathcal{L}_\mathcal{Q}$ if there is a single witness ω_0 such that $\forall\rho \in \mathcal{L}_\mathcal{Q}, \omega_0\mathcal{R}\rho$: the prover would not do anything and the verifier would simply generating an arbitrary state in $\mathcal{L}_\mathcal{Q}$. However, the guarantees are quite poor in that case as the verifier can fully describe the state... For this reason, we will focus on non trivial relations, and we will always specify the relation associated to a quantum language.

Note that in some cases, it may be cumbersome to write separately the language and the relation, especially when the witness is an arbitrary label and when only the equivalence class formed by the relation matters. In that case, we may abuse notations and write directly $\mathcal{L} = \{\mathcal{L}_\omega\}_\omega$, like $\mathcal{L}_{\mathcal{Q}}^{\mathrm{BB84}} = \{\{\left|0\right\rangle, \left|1\right\rangle\}, \{\left|+\right\rangle, \left|-\right\rangle\}\}$. This more succinct notation makes it clearer that the prover can choose in advance the basis (computation or Hadamard) of the state obtained by the verifier, that an honest verifier would always obtain a BB84 state, and that a malicious verifier would be unable to learn the basis of the output state. For the other above examples of languages $\mathcal{L}_{\mathcal{Q}}^{ex}$ and $\mathcal{L}_{\mathcal{Q}}^{p_k}$, the witness that we will consider is the support $\mathbf{d}_0$ of the GHZ state (therefore no malicious verifier will be able to learn any information about $\mathbf{d}_0$ beyond the fact that it respects the constraints that are specified in the language).

7.3.2 Formal definition of NIZKoQS

We can now provide the formal definition of (NI)ZKoQS.

Definition 7.3.1 (Zero-Knowledge Proof on Quantum State (ZKoQS))**.** *Let* (P, V) *be a protocol between an honest* QPT *prover*[13] P *and an honest* QPT *verifier* V *(that also outputs a final quantum state). Let* $E = \cup_{n\in\mathbb{N}}\mathscr{L}_\circ(\mathcal{H}_n)$ *be the set of finite dimensional quantum states (where* $\mathcal{H}_n$ *is the Hilbert space of dimension* n*),* $\mathcal{R} \subseteq \{0,1\}^* \times E$ *be a relation between bit strings (called* witnesses *or* classes*) and quantum states, and* $\mathcal{L}_\mathcal{Q} = \{\rho \in E \mid \exists\omega, \omega\mathcal{R}\rho\}$ *be a* quantum language *defined by* $\mathcal{R}$*. Then* $(\mathsf{P}_\lambda, \mathsf{V}_\lambda)$ *is said to be a* Zero-Knowledge proof on Quantum State *(ZKoQS) for* $\mathcal{L}_\mathcal{Q}$ *if the following properties are respected:*

1. ***Completeness****: There exists a negligible function* $\mu(\cdot)$ *such that for any* $\lambda \in \mathbb{N}$ *and* ω *such that* $\exists\rho' \in \mathcal{L}, \omega\mathcal{R}\rho'$*,*

$$\Pr\left[\, a = 1 \text{ and } \omega\mathcal{R}\rho \mid (a,\rho) \leftarrow \mathsf{OUT}_\mathsf{V}\langle \mathsf{P}_\lambda(\omega), \mathsf{V}_\lambda\rangle \,\right] = 1 - \mu(\lambda) \tag{7.7}$$

2. ***Soundness****: For any non-uniform* QPT *malicious prover* $\mathsf{P}^* = \{(\mathsf{P}^*_\lambda, \sigma_\lambda)\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $\mu(\cdot)$ *such that for any security parameter* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\, a = 1 \text{ and } \rho \notin \mathcal{L} \mid (a,\rho) \leftarrow \mathsf{OUT}_\mathsf{V}\langle \mathsf{P}^*_\lambda(\sigma_\lambda), \mathsf{V}_\lambda\rangle \,\right] \leq \mu(\lambda) \tag{7.8}$$

   When P^* *is unbounded, it is called a* proof *otherwise an* argument*.*

3. ***Quantum Zero Knowledge****: There exists a* QPT *simulator* Sim_λ *such that for any* QPT *verifier* $\mathsf{V}^* = \{(\mathsf{V}^*_\lambda, \sigma_\lambda)\}_{\lambda\in\mathbb{N}}$*,*

$$\{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}_\lambda(\omega), \mathsf{V}^*_\lambda(\sigma_\lambda)\rangle\}_{\lambda,\omega} \approx_c \{\mathsf{Sim}_\lambda(\mathsf{V}^*_\lambda, \sigma_\lambda)\}_{\lambda,\omega} \tag{7.9}$$

   where $\lambda \in \mathbb{N}$*,* $\omega \in \{\omega \mid \exists\rho, \omega\mathcal{R}\rho\}$*, and* V^* *is given to* Sim_λ *by sending the circuit description of* V^**.*

A Non-Interactive ZKoQS protocol—in which a single message is sent, from the prover to the verifier—will be denoted NIZKoQS.

Now, we define a protocol where we can prove any property on the set of entangled qubits of a hidden GHZ state in a NIZKoQS fashion.

[13] In our case the prover is actually PPT.

Remark 7.3.2. Note that we also require here the existence of $\texttt{CheckTrapdoor}(\mathbf{d}_0, t_k, k)$ that will check if $k \in \mathcal{K}$ and if $\mathbf{d}_0$ corresponds to the constant involved in the XOR property of k. The reason is that we cannot anymore be sure that Alice (the prover) is honest: therefore we need to check that the k sent by Alice was honestly prepared. This is particularly important when $\mathcal{K}_\lambda \subseteq \{0,1\}^*$ and when there exists no efficient algorithm to decide if a bit string $k^{(i)}$ is indeed an element of $\mathcal{K}_\lambda$. For example, with our construction, it is easy to produce a key k' that is indistinguishable from a key $k \in \mathcal{K}$, and such that the function $f_{k'}$ is injective instead of δ-2-to-1. Our construction based on LWE does guarantee that there exists a function `CheckTrapdoor`, which internally checks if the singular values of the trapdoor $\mathbf{R}$ are small enough and if the norm of $(\mathbf{s}_0, \mathbf{e}_0)$ is small enough (see Lemmas 5.3.2 and 5.3.5 for more details).

Protocol 8 BLIND-ZK

Assumptions: There exists a $\mathsf{negl}(\lambda)$-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions (Definition 4.2.1), together with an efficient function $\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0, t_k, k)$ outputting **true** if $k \in \mathcal{K}$ and if $\mathbf{d}_0 = \mathbf{d}_0(t_k)$.
Parties: A classical sender/prover (Alice) and a quantum receiver/verifier (Bob).
Common inputs: The size n of the hidden GHZ and an efficiently computable function $\mathsf{Auth}\colon \{0,1\}^n \times \mathcal{W} \to \{0,1\}$ where $\mathcal{W}$ is a set witnesses.
Alice's input: The support $\mathbf{d}_0 \in \{0,1\}^n$ of the hidden GHZ state and a witness $w \in \mathcal{W}$ such that $\mathsf{Auth}(\mathbf{d}_0, w) = 1$.
Bob's output: Bob can reject or accept and output a quantum state if he thinks that there exist $\mathbf{d}_0$ and w such that $\mathsf{Auth}(\mathbf{d}_0, w) = 1$ and such that the quantum state is a hidden GHZ state of support $\mathbf{d}_0$.
Protocol:

1. Alice generates $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0)$, a NIZK proof π proving that $\texttt{CheckTrapdoor}(\mathbf{d}_0, t_k, k) \wedge \mathsf{Auth}(\mathbf{d}_0, w) = 1$ (the witness being $(\mathbf{d}_0, t_k, w)$ and the word k) and sends (k, π) to Bob.
2. Bob checks that π is correct (if not it rejects), performs the quantum circuit described in GHZ-QFactory (described in Remark 4.2.2) to obtain a hidden GHZ state, and outputs that state.

Theorem 7.3.3 (NIZKoQS)**.** *Let $n \in \mathbb{N}$ be the size of the produced hidden GHZ state and $\delta = \mathsf{negl}(\lambda)$. The protocol BLIND-ZK (where Alice plays the role of the prover P and Bob the verifier V) is a NIZKoQS for the quantum language defined by all hidden GHZ states ρ on n qubits whose support $\mathbf{d}_0$ is such that there exists w such that $\mathsf{Auth}(\mathbf{d}_0, w)$ (defining the relation $\mathbf{d}_0 \mathcal{R} \rho$).*

Proof. The protocol is non interactive since a single message (k, π) is sent (the rest of the proof will also work for interactive protocols). The completeness is direct given the fact that the protocol is correct (this is a direct consequence of the correctness of GHZ-QFactory), that the NIZK is complete, and that δ is negligible. The soundness property relies on the soundness property of the classical NIZK protocol, and again on the correctness of the circuit performed by the server: the probability of accepting a k which is not in $\mathcal{K}_\lambda$ or such that there exists no w such that $\mathsf{Auth}(\mathbf{d}_0, w)$ is negligible; the correctness of the protocol is enough to conclude that the hidden GHZ has the expected properties.

For the Zero-Knowledge property, we define the following simulator, where Sim_{ZK} is the simulator of the classical (NI)ZK protocol, and $k \rightsquigarrow \mathsf{V}^*_\lambda$ is the machine obtained by running V^*_λ, and sending k as first message:

$\mathsf{Sim}_\lambda(\mathsf{V}^*_\lambda, \sigma_\lambda)$

1 : $\mathbf{d}'_0 \xleftarrow{\$} \{0,1\}^n$
2 : $(k', t_{k'}) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}'_0)$
3 : **return** $\mathsf{Sim}_{ZK}(k', k' \rightsquigarrow \mathsf{V}^*_\lambda, \sigma_\lambda)$

To prove that the output of $\mathsf{Sim}_\lambda(\mathsf{V}^*_\lambda, \sigma_\lambda)$ is indistinguishable from the real world, we define an hybrid distribution:

$\mathtt{Game1}(\mathbf{d}_0, \mathsf{V}^*_\lambda, \sigma_\lambda)$

1 : $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$
2 : **return** $\mathsf{Sim}_{ZK}(k, k \rightsquigarrow \mathsf{V}^*_\lambda, \sigma_\lambda)$

First, one can see that $\{\mathtt{Game1}(\mathbf{d}_0, \mathsf{V}^*_\lambda, \sigma_\lambda)\}_{\lambda,\mathbf{d}_0} \approx_c \{\mathsf{Sim}_\lambda(\mathsf{V}^*_\lambda, \sigma_\lambda)\}_{\lambda,\mathbf{d}_0}$. Indeed, if a non-uniform distinguisher D can distinguish between these two distributions, then we can use D to break the game $\mathtt{IND\text{-}DO}^{\mathcal{A}}_{\mathtt{Gen}}(\lambda)$ by simply sending for any λ a random $\mathbf{d}_0$ and the $\mathbf{d}_0$ which maximizes the distinguishing probability (anyway, D is already non-uniform). Then, $\{\mathtt{Game1}(\mathbf{d}_0, \mathsf{V}^*_\lambda, \sigma_\lambda)\}_{\lambda,\mathbf{d}_0} \approx_c \{\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}_\lambda(\mathbf{d}_0), \mathsf{V}^*_\lambda(\sigma_\lambda)\rangle\}_{\lambda,\mathbf{d}_0}$ since $\mathtt{Game1}$ is exactly the same as the RHS, except that we replaced the actual ZK protocol with its simulator, which is an indistinguishable process by definition of ZK. □

7.4 Multi-Party Generation of Authorized Hidden GHZ States

7.4.1 Cryptographic requirements

All the protocols are based on the existence of a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions, already defined in Definition 4.2.1. This property is enough to prove the security of BLIND and $\mathtt{BLIND}^{\mathtt{sup}}$ against an arbitrary corruption of parties, and can be used to prove the security of $\mathtt{BLIND}_{\mathtt{can}}$ and $\mathtt{BLIND}^{\mathtt{sup}}_{\mathtt{can}}$ when the adversary corrupts only the server and the non-supported applicants. However, if we want to prove the security of these last two protocols in a stronger attack model, namely when the adversary can also corrupt some supported applicants, we also require our function to have a stronger property. Intuitively, the PartInfo function will list the messages to send to all applicants: if it contains a ✗ for applicant i, it means that applicant i is not part of the support of the GHZ, if it is a 0 or a 1, it means that the applicant gets a GHZ canonical state—up to a local X correction if it is a 1—and, if it is a $\perp$, it means that the protocol aborts "locally".

Remark 7.4.1. This local abort is interesting since it triggers when y has only one preimage, and this means that the server is malicious with overwhelming probability[14]. Note that one may want to send this abort bit to all applicants, however it is not yet known if leaking this bit to other corrupted applicants could reduce the security of the protocol (for example, it is not clear if a malicious server could force the protocol to abort when one specific honest applicant is not part of the $\mathsf{GHZ}^{\mathsf{G}}$). To avoid that issue, we introduce this notion of local abort, that tells locally to participants if the server was behaving honestly. Note that it is important to make sure that this abort bit do not leak to the adversary later, otherwise the security is not guaranteed anymore.

Definition 7.4.2. *A δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of function $\{f_k\}$ is said to be δ-$\mathsf{GHZ}^{\mathsf{can}}$ capable if this additional property is respected:*

- ***indistinguishability with partial knowledge**: We want to show that, by leaking some information about the "key" of the GHZ state owned by malicious applicants, we do not reveal additional information about the support status of applicants. More precisely, there exists a PPT algorithm $\mathtt{PartInfo}$: $\mathcal{T}_\lambda \times \mathcal{Y}_\lambda \to \{0, 1, ✗, \perp\}^n$ with the following properties:*

[14] This is the case when $\delta = \mathsf{negl}(\lambda)$, which is possible to obtain using LWE with superpolynomial noise ratio.

- ***correctness***: $\forall k \in \mathcal{K}_\lambda, y \in \mathcal{Y}_\lambda$, *and* $v \leftarrow \texttt{PartInfo}(t_k, y)$:
  - *y has exactly two preimages iff there is no partial abort (see discussion above):* $|f_k^{-1}(y)| = 2$ *iff* $\perp \notin v$
  - *for all i, if $v[i] \in \{0,1\}$ then $\mathbf{d}_0[i] = 1$, and if $v[i] =$ ✗, then $\mathbf{d}_0[i] = 0$ (required to make sure ✗ is sent only to non-supported applicants and that $0/1$ is sent only to supported applicants).*
  - *if $|f_k^{-1}(y)| = 2$, $\exists u \in \{h(x), h(x')\}$, such that $\forall i$, if $\mathbf{d}_0[i] = 1$ then $v[i] = u[i]$ (required to make sure that all corrections are correct).*
- ***security***: *The game on the left is impossible to win with non negligible advantage for any QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ (note that $\mathcal{M}$ is intuitively the set of malicious corrupted applicants, and the condition line 2 is added because otherwise there is a trivial uninteresting way to distinguish).*

$\texttt{IND-PARTIAL}^{\mathcal{A}}_{\texttt{Gen},\texttt{PartInfo}}(\lambda)$

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : **if** $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$: **return false fi**
3 : $c \xleftarrow{\$} \{0,1\}$
4 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
5 : $y \leftarrow \mathcal{A}_2(k)$
6 : $v \leftarrow \texttt{PartInfo}(t_k, y)$
7 : $\tilde{c} \leftarrow \mathcal{A}_3(\{(i, v[i])\}_{i \in \mathcal{M}})$
8 : **return** $\tilde{c} = c$

For our protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$, we also need to make sure that this family of functions can be computed in a distributed manner among users:

Definition 7.4.3. *A δ-$\mathsf{GHZ}^{\mathsf{can}}$ capable family of function $\{f_k\}$ is said to be distributable if the above procedures can be computed after gathering partial results from the parties. More precisely:*

- *There exists $\texttt{Gen}_{Loc}$, a "local" generation procedure such that sampling $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0)$ can be done by first sampling for all i: $(k^{(i)}, t_k^{(i)}) \leftarrow \texttt{Gen}_{Loc}(1^\lambda, \mathbf{d}_0[i])$ and defining $k := (k^{(1)}, \ldots, k^{(n)})$ and $t_k := (t_k^{(1)}, \ldots, t_k^{(n)})$. We will denote as $\mathcal{K}_{\lambda,Loc}$ the set of such $k^{(i)}$, and we assume that $\mathcal{K} = \mathcal{K}_{\lambda,Loc}^n$.*
- *Similarly, there exists $\texttt{PartInfo}_{Loc}$, a "local" version of $\texttt{PartInfo}$ such that sampling $v \leftarrow \texttt{PartInfo}(t_k, y)$ can be done by sampling for all i: $v[i] \leftarrow \texttt{PartInfo}_{Loc}(t_k^{(i)}, y)$.*

- *Finally, there exists a method* $\texttt{PartAlpha}_{Loc}$ *such that for any bit string* b *and for any* y *such that* $f_k^{-1}(y) = \{x, x'\}$ *with* $x \neq x'$ *we have* $\langle b, x \oplus x' \rangle = \oplus_i \texttt{PartAlpha}_{Loc}(i, t_k^{(i)}, y, b)$.

Moreover, as discussed in Remark 7.3.2, we cannot assume anymore that people running these functions will be honest. Therefore, if we want to make sure that a non-supported malicious applicant cannot alter the state obtained by supported applicants (for example by providing a function which is not δ*-2-to-1), we also require the existence of a circuit* $\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)})$ *that returns* **true** *iff* $k^{(i)} \in \mathcal{K}_{\lambda,Loc}$ *and if* $k^{(i)}$ *is the public key corresponding to the trapdoor* $t_k^{(i)}$*, embedding the bit* $\mathbf{d}_0[i]$*. This circuit can in particular be combined with a ZK protocol to prove in a Zero-Knowledge way that* $k^{(i)} \in \mathcal{K}_{\lambda,Loc}$.

We also provide in Section 7.4.3 a generic construction that turns a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions into a δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable distributable family of functions, with $\delta' = 1 - (1-\delta)^n \leq \delta n$. In particular, if δ is a negligible function of λ as in Theorem 5.3.7 and $n = O(\lambda)$, δ' is negligible.

7.4.2 The different protocols

In this section, we define the protocols BLIND (Protocol 9), $\texttt{BLIND}^{\texttt{sup}}$ (Protocol 10), $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ (Protocol 11) and finally our main protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ (Protocol 12). We also prove in Section 7.4.2.3 the impossibility of $\texttt{BLIND}_{\texttt{can}}$.

7.4.2.1 The protocol BLIND

We define now the protocols BLIND (Protocol 9), which is the basic building block of all the other protocols. This protocol is basically like GHZ-QFactory except that a trusted classical party, Cupid, is in charge of choosing the support of the hidden GHZ state, and that the output state is distributed among the applicants (one qubit per applicant), who just need to store the received qubit. Besides the protocol itself, what interests us here (and more importantly, in the coming protocols) is how security degrades when information is leaked about the hidden GHZ state.

Lemma 7.4.4 (Correctness of BLIND and $\texttt{BLIND}^{\texttt{sup}}$)**.** *At the end of an honest run of protocol BLIND, when* y *has exactly two distinct preimages* x, x' *(which occurs with probability* $1 - \delta$ *according to Definition 4.2.1, which is overwhelming when we use the construction defined in Theorem 5.3.7), the state shared between all applicants is a hidden GHZ state* $|d\rangle + (-1)^\alpha |d'\rangle$ *state, with:*

$$\mathbf{d} = h(x) \qquad \mathbf{d}' = h(x') \qquad \alpha = \bigoplus_i b_i(x_i \oplus x'_i) = \langle b, x \oplus x' \rangle \tag{7.10}$$

Protocol 9 BLIND

Assumptions: There exists a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family of functions with $\delta = \mathsf{negl}(\lambda)$.
Inputs: Cupid gets as input $\mathbf{d}_0 \in \{0,1\}^n$, a bit string describing the final supported applicants: applicant a_i will be supported iff $\mathbf{d}_0[i] = 1$. $\lambda \in \mathbb{N}$ is a public, fixed, security parameter.
Protocol:

1. Cupid generates $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0)$, and send k to the server.
2. Bob performs the circuit done in the GHZ-QFactory protocol (circuit in Figure 4.1). Then, he sends (y, b) to Cupid, and for all i, Bob send the ith output qubit to applicant a_i.
3. Each applicant just receives and stores the qubit sent by the server.

In particular, since by definition of f_k we have $\mathbf{d}_0 = h(x) \oplus h(x') = \mathbf{d} \oplus \mathbf{d}'$, the support of the hidden GHZ is $\mathbf{d}_0$.

Proof. This is a direct consequence of the correctness of GHZ-QFactory (Lemma 4.3.3). □

We show now that at the end of a fully malicious interaction during the protocol BLIND, where all applicants and the server can be fully malicious and can all collude together, the set of supported applicants is completely hidden:

Lemma 7.4.5 (Security of BLIND)**.** *If we define a game following the spirit of* IND-CPA *security, no QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can win the game IND-BLIND with probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$.*

$\mathtt{IND\text{-}BLIND}^{\mathcal{A}}_{\mathtt{Gen}}(\lambda)$

1 : $(\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $(k, t_k) \leftarrow \mathtt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
4 : $(y, b, \tilde{c}) \leftarrow \mathcal{A}_2(k)$
5 : // No more interaction
6 : **return** $\tilde{c} = c$

Proof. This is a direct consequence of the soundness of GHZ-QFactory shown in Lemma 4.3.4. □

7.4.2.2 The protocol $\texttt{BLIND}^{\texttt{sup}}$

We describe now the protocol $\texttt{BLIND}^{\texttt{sup}}$ (Protocol 10). In this protocol, all the applicants will obtain a qubit part of a hidden GHZ state, and they will learn their own support status. However, they will not know the "key" of the hidden GHZ state.

Protocol 10 $\texttt{BLIND}^{\texttt{sup}}$

Assumptions: Same as $\texttt{BLIND}$ (δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family with $\delta = \mathsf{negl}(\lambda)$).
Inputs: Same as $\texttt{BLIND}$: Cupid gets $\mathbf{d}_0$ and λ.
Protocol:

1. Run the protocol $\texttt{BLIND}$, so that Cupid gets (b, y) and each applicant a_i the i-th qubit
2. For all i, Cupid sends $\mathbf{d}_0[i]$ to applicant a_i, so that each applicant knows whether they are supported or not.

Now, in order to prove the security of the $\texttt{BLIND}^{\texttt{sup}}$ protocol, we first need to define what we mean by security. Since in this protocol Cupid reveal to all applicants their respective support status, we cannot use the previous definition of security.

We show now that if we allow in the protocol $\texttt{BLIND}^{\texttt{sup}}$ the fully malicious server Bob to corrupt an arbitrary subset of applicants, then the support status of the remaining honest applicants is completely hidden:

Lemma 7.4.6 (Security of $\texttt{BLIND}^{\texttt{sup}}$)**.** *No QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ can win the game $\textit{IND-BLIND}^{\textit{sup}}$ with probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$. In the following, $\mathcal{M}$ is the set of malicious applicants corrupted by Bob, and the condition $\forall i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] = \mathbf{d}_0^{(1)}[i]$ is required to avoid a trivial uninteresting distinguishing strategy.*

Proof. To prove the security of this scheme, we will assume by contradiction that there exists an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ that can win the game $\texttt{IND-BLIND}^{\texttt{sup}}$ with probability $p_{\mathcal{A}} \coloneqq \frac{1}{2} + \frac{1}{\mathsf{poly}(\lambda)}$, and we will construct an adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ that can win the game $\texttt{IND-BLIND}$ with a non negligible advantage (which is impossible by assumption). So we define $\mathcal{A}'_1(\lambda)$ as follows: $\mathcal{A}'_1$ runs in a blackbox way $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1$, returns $(\mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)})$ and keeps $(\mathcal{M}, \mathbf{d}_0^{(0)})$ in its internal state. We then define:

$$\mathcal{A}'_2(k, \mathsf{state}_1 \coloneqq (\mathcal{M}, \mathbf{d}_0^{(0)})) \coloneqq (y, b) \leftarrow \mathcal{A}_2(k); \tilde{c} \leftarrow \mathcal{A}_3(\{(i, \mathbf{d}_0^{(0)}[i])\}_{i \in \mathcal{M}}); \textbf{return } \tilde{c} \quad (7.11)$$

It is then easy to see that $\mathcal{A}'$ wins the game $\texttt{IND-DO}$ with probability greater than $p_{\mathcal{A}}$. Indeed, when $\mathcal{A}_1$ outputs a $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)})$ that does not respect the condition $\forall i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] = \mathbf{d}_0^{(1)}[i]$, then $\mathcal{A}$ always lose (while $\mathcal{A}'$ may win the game). Moreover, when

$\texttt{IND-BLIND}^{\texttt{sup}}\mathcal{A}_{\{f_k\}}(\lambda)$

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : **if** $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$:
3 : **return false fi**
4 : $c \xleftarrow{\$} \{0,1\}$
5 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
6 : $(y, b) \leftarrow \mathcal{A}_2(k)$
7 : // The adversary has only access
8 : // to the messages sent by Cupid
9 : // to corrupted applicants:
10 : $\tilde{c} \leftarrow \mathcal{A}_3(\{(i, \mathbf{d}_0^{(c)}[i])\}_{i\in\mathcal{M}})$
11 : **return** $\tilde{c} = c$

the condition is respected, since $\{(i, \mathbf{d}_0^{(0)}[i])\}_{i\in\mathcal{M}} = \{(i, \mathbf{d}_0^{(1)}[i])\}_{i\in\mathcal{M}} = \{(i, \mathbf{d}_0^{(c)}[i])\}_{i\in\mathcal{M}}$, we can replace the input of $\mathcal{A}'_3$ with $\{(i, \mathbf{d}_0^{(c)}[i])\}_{i\in\mathcal{M}}$: the game is now exactly equivalent to $\texttt{IND-BLIND}^{\texttt{sup}}$, so in that case $\mathcal{A}'$ win with the exact same probability as $\mathcal{A}$. Therefore, $\mathcal{A}'$ wins the game IND-DO with probability greater than $p_{\mathcal{A}} = \frac{1}{2} + \frac{1}{\mathsf{poly}(n)}$: contradiction. □

7.4.2.3 The impossible protocol $\texttt{BLIND}_{\texttt{can}}$

One may be interested by a protocol $\texttt{BLIND}_{\texttt{can}}$, that would make sure that all supported applicants share a canonical GHZ, but such that at the same time none of them know if they are part of the GHZ or not. We state here that such security guarantee is impossible, and why it is therefore meaningful to consider instead $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ protocols.

We show now that there exists no protocol $\texttt{BLIND}_{\texttt{can}}$ such that, at the end of an honest interaction, all supported applicants share a canonical GHZ, and such that none of them know their own support status:

Lemma 7.4.7 (Impossibility of a secure $\texttt{BLIND}_{\texttt{can}}$ protocol)**.** *There exists always an adversary $\mathcal{A}$ that can win the game ImpossibleGame.*

Proof. For simplicity, we only sketch the proof. When it comes to proving the security of the protocol, we realize that at least one of the supported applicants needs to be honest, otherwise it is trivial to distinguish any correct protocol: The attacker can always send $\mathbf{d}_0^{(0)} = \begin{pmatrix}1 & 1 & 0\dots & 0\end{pmatrix}$ and $\mathbf{d}_0^{(1)} = \begin{pmatrix}0 & 0 & 0\dots & 0\end{pmatrix}$, and at the end of any (correct) protocol run honestly, the attacker will get either a Bell pair on the first two qubits or two qubits not entangled. It is therefore easy to distinguish, so the condition in

ImpossibleGame

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$

// Avoid trivial attack: check if at least one honest applicant is in the GHZ

2 : **if** $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(0)}[i] = 0)$ or $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(1)}[i] = 0)$ **then return false fi**

3 : // Run $\texttt{BLIND}_{\texttt{can}}$ with adversary.$\tilde{c} \leftarrow \mathcal{A}_3()$

4 : **return** $\tilde{c} = c$

ImpossibleGame is indeed required. But it is not enough. Even if we assume that one applicant is honest (say the first one), it is still impossible to prove the security of the protocol.

Indeed, let us consider an adversary that sends:

$$\mathbf{d}_0^{(0)} = \begin{pmatrix} 1 & \dots 1 \end{pmatrix} \text{ and } \mathbf{d}_0^{(0)} = \begin{pmatrix} 1 & \dots & 1 & 0 \dots & 0 \end{pmatrix} \tag{7.12}$$

(the first half of the qubits is 1, the second half is 0). Then, a first remark is that at the end of an honest protocol, all the qubits that are not entangled must be all equal, i.e. if $c = 1$, the state obtained is $(|0 \dots 0\rangle + |1 \dots 1\rangle) \otimes |0 \dots 0\rangle$ or $(|0 \dots 0\rangle + |1 \dots 1\rangle) \otimes |1 \dots 1\rangle$. Indeed, if some qubits in the second half are different, then a measurement in the computational basis will reveal some different outcomes with high probability (while when $c = 0$ all measurements are equal since the state is a canonical GHZ state by the correctness property). But even in that case, it is still easy to distinguish: when we do the measurement, in the first case, we either get $1 \dots 1$ or $0 \dots 0$. In the second case, however, the first part may be different compared to the second part, i.e. we can measure either a $0 \dots 0$ or a $1 \dots 10 \dots 0$ with probability $\frac{1}{2}$. This last measurement is enough to distinguish, we can just ask to $\mathcal{A}$ to measure the state in the computational basis: if all measurements are equal, $\mathcal{A}$ picks $\tilde{c}$ uniformly at random, otherwise $\mathcal{A}$ outputs $\tilde{c} = 1$. $\mathcal{A}$ will succeed with non negligible advantage. □

Therefore it is not possible to hide to an adversary its support status, so the best we can get is to prove that no adversary can learn the support status of the honest applicants, which is the goal of the protocol $\texttt{BLIND}_{\texttt{can}}^{\texttt{sup}}$.

7.4.2.4 The protocol $\texttt{BLIND}_{\texttt{can}}^{\texttt{sup}}$

We present now the protocol $\texttt{BLIND}_{\texttt{can}}^{\texttt{sup}}$ (Protocol 11): at the end of the protocol, the supported applicants share a canonical GHZ state, and each applicant knows their own support status.

Protocol 11 $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$

Assumptions: There exists a δ-$\mathsf{GHZ}^{\mathsf{can}}$ capable family of functions with $\delta = \mathsf{negl}(\lambda)$.
Inputs: Same as BLIND: Cupid gets $\mathbf{d}_0$ and λ.
Protocol:

1. Run the protocol BLIND, so that Cupid gets (b, y) and each applicant a_i the ith qubit
2. Cupid computes $v \leftarrow \texttt{PartInfo}(t_k, y)$, and if $f_k^{-1}(y) = \{x, x'\}$ with $x \neq x'$, compute $\alpha := \langle b, x \oplus x' \rangle$ (otherwise, sample α randomly). Computes the supported set $\mathcal{S} = \{i \mid \mathbf{d}_0[i] = 1\}$. Sample uniformly at random $\hat{\alpha} \leftarrow \{0,1\}^n$ such that $\alpha = \oplus_{i \in \mathcal{S}} \hat{\alpha}_i$. For all i, send $(\hat{\alpha}_i, v[i])$ to applicant a_i.
3. All applicants: When receiving the message $(\hat{\alpha}_i, v[i])$:
   - If $v[i] = \boldsymbol{\times}$, then it means that the applicant is not part of the support of the final GHZ. The end.
   - If $v[i] = \perp$, it is a local abort. It's likely that the server was malicious. Do not reveal this information to the server. The end.
   - If $v[i] \in \{0,1\}$, it means that this applicant is part of the final GHZ state. Apply $Z^{\hat{\alpha}_i} X^{v[i]}$ on the qubit sent by the server.

Lemma 7.4.8 (Correctness of $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$)**.** *If all parties are honestly running the* $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ *protocol, then at the end of the protocol, with probability* $1 - \delta$ *(so with overwhelming probability when* δ *is negligible), all supported applicants share a canonical* GHZ*, and all applicants know whether or not they are supported.*

Proof. With probability $1 - \delta$, the y obtained by Cupid has exactly two preimages. In that case, due to the correctness property of PartInfo given Definition 7.4.2 (part 1) we get $\forall i,\ v[i] \neq \perp$, so $v[i] \in \{0, 1, \boldsymbol{\times}\}$. Then, using Lemma 7.4.4, we know that the state shared by all participants after the BLIND part is $|h(x)\rangle + (-1)^{\alpha} |h(x')\rangle$, with $h(x) \oplus h(x') = \mathbf{d}_0$. We can combine this using part 2 of the correctness property given in Definition 7.4.2, (that states that $v[i] \in \{0,1\}$ iff $\mathbf{d}_0[i] = 1$): because the set of supported participants is $\mathcal{S} := \{i \mid \mathbf{d}_0[i] = 1\}$, we have for all $i \notin \mathcal{S}$: $h(x)[i] = h(x')[i]$. Thus the register of each applicant $i \notin \mathcal{S}$ is in a tensor product with all the other qubits, so we can factor them out, and consider only the state shared by applicants $i \in \mathcal{S}$ (in that case $h(x')[i] = 1 \oplus h(x)[i]$):

$$\bigotimes_{i \in \mathcal{S}} |h(x)[i]\rangle + (-1)^{\alpha} \bigotimes_{i \in \mathcal{S}} |h(x')[i]\rangle = \bigotimes_{i \in \mathcal{S}} |h(x)[i]\rangle + (-1)^{\alpha} \bigotimes_{i \in \mathcal{S}} |1 \oplus h(x)[i]\rangle \tag{7.13}$$

After the corrections, the state becomes:

$$\bigotimes_{i \in \mathcal{S}} X^{v[i]} |h(x)[i]\rangle + (-1)^{\alpha \oplus \bigoplus_{i \in \mathcal{S}} \hat{\alpha}_i} \bigotimes_{i \in \mathcal{S}} X^{v[i]} |1 \oplus h(x)[i]\rangle \tag{7.14}$$

And due to the fact that $\alpha = \bigoplus_{i\in\mathcal{S}} \hat{\alpha}_i$, we can get rid of the phase. Moreover, we can now use the part 3 of Definition 7.4.2 which states that there exists $u \in \{h(x), h(x')\}$ such that if $\mathbf{d}_0[i] = 1$, then $v[i] = u[i]$. So if $\forall i \in \mathcal{S}$, we have $u[i] = v[i] = h(x)[i]$, then after the correction we get the state $|0\dots0\rangle + |1\dots1\rangle$, which is a canonical GHZ, and if $u[i] = v[i] = h(x')[i] = 1 - h(x)[i]$, then we get $|1\dots1\rangle + |0\dots0\rangle$, which is the same canonical GHZ state. □

Similarly, we can show the security of $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$. We prove that, if we allow in the protocol $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ the fully malicious server to corrupt a subset of applicants in such a way that either at least one supported applicant is not corrupted or no supported applicant is corrupted[15], then the support status of honest applicants is completely hidden:

Lemma 7.4.9 (Security of $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$)**.** *No QPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ can win the game $\textit{IND-BLIND}^{\textit{sup}}_{\textit{can}}$ with probability better than $\frac{1}{2} + \mathsf{negl}(\lambda)$. In the following, $\mathcal{M}$ is the set of malicious applicants corrupted by Bob, and the condition $\forall i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] = \mathbf{d}_0^{(1)}[i]$ is required to avoid a trivial uninteresting distinguishing strategy.*

$\texttt{IND-BLIND}^{\texttt{sup}}_{\texttt{can}}{}^{\mathcal{A}}_{\{f_k\}}(\lambda)$

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : **if** $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$ **then return false fi**
3 : **if** $(\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)} = 1)$ and $((\forall i \notin \mathcal{M}, \mathbf{d}_0^{(0)}[i] = 0)$ or $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(1)}[i] = 0))$
4 : **then return false fi**
5 : $c \xleftarrow{\$} \{0,1\}$
6 : $(k, t_k) \leftarrow \texttt{Gen}(1^\lambda, \mathbf{d}_0^{(c)})$
7 : $(y, b) \leftarrow \mathcal{A}_2(k)$
8 : $v \leftarrow \texttt{PartInfo}(t_k, y)$
9 : **if** $\perp \notin v$ **then** $\alpha := \langle b, x \oplus x' \rangle$ **else** $\alpha \xleftarrow{\$} \{0,1\}$ **fi**
10 : $\hat{\alpha} \leftarrow \{\hat{\alpha} \mid \hat{\alpha} \in \{0,1\}^n, \bigoplus_{i\in\mathcal{S}} \hat{\alpha}_i = \alpha \text{ or } \mathcal{S} = \emptyset\}$
11 : // The adversary has only access to the messages sent by Cupid to corrupted applicants:
12 : $\tilde{c} \leftarrow \mathcal{A}_3(\{(i, \hat{\alpha}_i, v[i])\}_{i\in\mathcal{M}})$
13 : **return** $\tilde{c} = c$

[15] Otherwise there is a trivial, fundamental, attack to any protocol which consists in setting $\mathbf{d}_0^{(0)} = (01)$, $\mathbf{d}_0^{(1)} = (11)$, $\mathcal{M} = \{2\}$ and then testing if the quantum state obtained by the party 2 is a $|+\rangle$ or not. However, this attack is not possible anymore if the adversary is not in possession of one part of the GHZ (for example if we replace $\mathbf{d}_0^{(0)} = (00)$ and $\mathbf{d}_0^{(1)} = (10)$ in the above example), that is the reason why we can provide a stronger security guarantee when no supported applicant is supported.

Proof. The first step in the proof is to note that, due to the condition line 3, we have either:

- $\forall i \in \mathcal{M}$, $\mathbf{d}_0^{(0)} = 0$, i.e. $\mathcal{M} \cap \mathcal{S} = \emptyset$. But since we send $\hat{\alpha}_i$ to the adversary only if $i \in \mathcal{M}$, and since the line 10 does not put any restriction on the sampling $\hat{\alpha}_i$ when $i \notin \mathcal{S}$, the line 9 and 10 can be replaced with a single line $\hat{\alpha} \leftarrow \{0,1\}^n$.
- or there exists $j \notin \mathcal{M}$ such that $\mathbf{d}_0^{(c)}[j] = 1$, i.e. such that $j \in \mathcal{S}$. Therefore, $\hat{\alpha}$ can be sampled by choosing for all $i \neq j$, $\hat{\alpha}_i$ randomly, and finally by setting $\hat{\alpha}_j = \alpha \oplus \bigoplus_{i \in \mathcal{S} \setminus \{j\}} \hat{\alpha}_j$ (this is statistically indistinguishable). But since $\hat{\alpha}_j$ is never sent to $\mathcal{A}$ because $j \notin \mathcal{M}$, we can also remove lines 9 and 10 and replace them with $\hat{\alpha} \leftarrow \{0,1\}^n$.

This gives us a new game Game1.

$\mathtt{Game1}^{\mathcal{A}}$

1 : // ...First 7 lines like $\mathtt{IND\text{-}BLIND}^{\mathtt{sup}}_{\mathtt{can}}$
2 : $\hat{\alpha} \leftarrow \{0,1\}^n$
3 : $\tilde{c} \leftarrow \mathcal{A}_3(\{(i, \hat{\alpha}_i, v[i])\}_{i \in \mathcal{M}})$
4 : **return** $\tilde{c} = c$

Since the two games are exactly equivalent, we have $\Pr\left[\mathtt{IND\text{-}BLIND}^{\mathtt{sup}\,\mathcal{A}}_{\mathtt{can}}\right] = \Pr\left[\mathtt{Game1}^{\mathcal{A}}\right]$. Then, define a new game Game2 by removing the condition line 3 and 4.

$\mathtt{Game2}^{\mathcal{A}}$

1 : // ...Remove line 3 and 4 of Game1
2 : ~~**if** $(\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)} = 1)$ and $((\forall i \notin \mathcal{M}, \mathbf{d}_0^{(0)}[i] = 0)$ or $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(1)}[i] = 0))$~~
3 : ~~**then return false fi**~~
4 : // ...Rest is like Game1

We can remark that this condition cannot help the adversary to win since entering inside this condition always returns "false", therefore $\Pr\left[\mathtt{Game1}^{\mathcal{A}}\right] \leq \Pr\left[\mathtt{Game2}^{\mathcal{A}}\right]$. But now, we remark that Game2 is very similar to the game defined Definition 7.4.2, except that we provide an additional random string $\hat{\alpha}$ to $\mathcal{A}$. But since this string is random, it is easy to see that we can turn any adversary $\mathcal{A}$ winning Game2 with probability p into an adversary $\mathcal{A}'$ winning the game with the same probability p by defining $\mathcal{A}'(\{(i, v[i])\}_{i \in \mathcal{M}})$ as an adversary sampling a uniformly random bit string $\hat{\alpha}$ and calling $\mathcal{A}(\{(i, \hat{\alpha}_i, v[i])\}_{i \in \mathcal{M}})$ (and reciprocally, any adversary that can win without access to $\hat{\alpha}$ can win Game2 with access to $\hat{\alpha}$ by simply forgetting this value). So we get:

$$\max_{\text{QPT } \mathcal{A}} \Pr\left[\mathtt{Game2}^{\mathcal{A}}\right] = \max_{\text{QPT } \mathcal{A}} \Pr\left[{}^{\mathcal{A}}\right] \tag{7.15}$$

But by assumption, for any QPT $\mathcal{A}$, $\Pr\left[{}^{\mathcal{A}}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$. So by combining all the inequations we showed on games, we also get:

$$\Pr\left[\texttt{IND-BLIND}^{\texttt{sup}}_{\texttt{can}}{}^{\mathcal{A}}_{\{f_k\}}(\lambda)\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{7.16}$$

which ends the proof. □

7.4.2.5 The protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$

We can now define our main protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$. Similarly to the $\texttt{BLIND}^{\texttt{sup}}_{\texttt{can}}$ protocol, each supported applicant is supposed to end up with a canonical GHZ state, and the support status of each applicant should be unknown to the other applicants and to the server. However, in this protocol the trusted party Cupid is not needed anymore: each applicant is supposed to choose themselves their own support status, and they will be assured that no malicious party (including the server) can learn it.

Moreover, the server can have some guarantees on the support status of the applicants: for example, the server can ensure that *if* some applicants are supported, then they all know a classical secret (but the server has no way to know whether or not a given applicant is supported). This secret can be any witness of a NP relation: it could be a password, a private key linked with some known public key, a signature from a third party Certification Authority, the proof of any famous theorem... We formalize it by defining n deterministic functions $\mathsf{Auth}_i \colon \{0,1\} \times \{0,1\}^* \to \{0,1\}$ responsible of the "authorization" of the applicants: the server will allow applicant i to be part of the protocol iff they can prove in a NIZK way that they know w such that $\mathsf{Auth}_i(\mathbf{d}_0[i], w) = 1$. For instance, we can use the Auth_i function to ensure that an applicant is part of the GHZ iff they know a password whose hash by h is x by defining $\mathsf{Auth}_i(\mathbf{d}_0[i], \tilde{s}_i)) \coloneqq (\mathbf{d}_0[i] = 0 \vee h(\tilde{s}_i) = x)$. Again, we emphasize that Auth_i does *not* reveal the value of $\mathbf{d}_0[i]$: it just reveals that *if* the user is supported, then they know the password. This verification only requires a single message from the client(s) and is therefore achieving NIZKoQS, as formalized in Section 7.3.

We will now prove the correctness and security of the $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ protocol. Note that an honest server can obtain guarantees on the distributed state even in the presence of malicious or noisy applicants. Assuming here an honest server is not absurd, notably when the server wants to use this GHZ, for example to share a quantum state or if a verification is done afterwards. This centralization is also useful in the presence of many noisy clients (a single hardware needs to be noiseless, while in a decentralized MPC computation the protocol is likely to always abort if a single client is noisy).

Protocol 12 AUTH-BLIND$^{\text{dist}}_{\text{can}}$

Inputs: Each applicant i gets λ. They also get $\mathbf{d}_0[i] \in \{0,1\}$ and $w_i \in \{0,1\}^*$ such that $\mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$ ($\mathbf{d}_0[i] = 1$ iff applicant i wants to be supported). The authentication functions $\{\mathsf{Auth}_i\}_{i\in[n]}$ are also public.

Assumptions: There exists a δ-$\mathsf{GHZ}^{\mathsf{can}}$ capable distributable family of functions with $\delta = \mathsf{negl}(\lambda)$. Cupid is not required anymore, and instead we require the existence of a classical (but quantum-secure) Multi-Party Computation protocol.

Protocol:

1. Each applicant a_i: Run $(k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$, send $k^{(i)}$ to the server, and continue the protocol.
2. Server: Run (as a verifier) a Zero-Knowledge protocol with each applicant (the prover) to check that $k^{(i)}$ is well prepared, and that the applicant can authenticate the quantum state. More precisely, each applicant i proves to the server that they know $(\mathbf{d}_0[i], t_k^{(i)}, w^{(i)})$ such that $\mathtt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)}) \wedge \mathsf{Auth}_i(\mathbf{d}_0[i], w^{(i)}) = 1$. If the protocol fails with at least one applicant, abort after sending $\perp$ to all applicants (the server can also output if needed the identity of the applicant who were malicious in case other actions should be performed with respect to them, e.g. in further runs). Otherwise, the protocol continues.
3. Server: Compute $k \coloneqq (k^{(1)}, \ldots, k^{(n)})$, run the quantum circuit already described in protocol BLIND, and for all i, send the ith qubit of the second register to applicant a_i together with (y, b).
4. For each applicant i: Compute $v[i] \leftarrow \mathtt{PartInfo}_{\mathtt{Loc}}(t_k^{(i)}, y)$, and compute via a MPC protocol the function `CombineAlpha` which returns a secret share of α between supported applicants. More precisely, it returns to each applicant i a random bit $\hat{\alpha}_i$ such that $\oplus_{i\in\mathcal{S}}\hat{\alpha}_i = \alpha = \bigoplus_i \mathtt{PartAlpha}_{\mathtt{Loc}}(i, t_k^{(i)}, y, b)$, where $\mathcal{S}$ is the set of supported applicants (details in Figure 7.2). The reason we use a MPC protocol is that $\mathtt{PartAlpha}_{\mathtt{Loc}}(i, t_k^{(i)}, y, b)$ can leak[a] information about the bit $\mathbf{d}_0[i]$, so this bit should not be revealed directly.
5. For each applicant i: If the outcome $\hat{\alpha}_i$ of the MPC is $\perp$, abort. Otherwise, similarly to the last step of the BLIND$^{\text{sup}}_{\text{can}}$protocol: if $v[i] \in \{0,1\}$, apply the correction $X^{v[i]}Z^{(\hat{\alpha}_i)}$ on the qubit, else discard the qubit.

[a] The attack would be as follows: the malicious server Bob can run the GHZ-QFactory protocol with $k^{(i)}$ which gives him a BB84 state in the basis $\mathbf{d}_0[i]$: if $\mathbf{d}_0[i] = 0$ then it gets either $|0\rangle$ or $|1\rangle$, but if $\mathbf{d}_0[i] = 1$ then it gets the state $|+\rangle$ if $\alpha_p^{(i)}\alpha_i = 0$ and a $|-\rangle$ otherwise. So the trick is to measure the state in the Hadamard basis: if the measurement is different from $\alpha_p^{(i)}$ then we know that $\mathbf{d}_0[i] = 0$, and otherwise the server will randomly guess the value of $\mathbf{d}_0[i]$. It is easy to see that if $\mathbf{d}_0[i]$ is chosen uniformly at random, then the server has a non-negligible advantage in guessing $\mathbf{d}_0[i]$.

Algorithm 4 $\texttt{CombineAlpha}((k, y, b), (t_k^{(1)}, \mathbf{d}_0[1], k^{(1)}, y^{(1)}, b^{(1)}), \ldots, (t_k^{(n)}, \mathbf{d}_0[n], k^{(1)}, y^{(n)}, b^{(n)}))$

// Check if the input are honestly prepared
if $k \neq (k^{(1)}, \ldots, k^{(n)})$ or
  $\exists i, y^{(i)} \neq y$ or $b^{(i)} \neq b$ or $\neg\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)})$
 then return $\perp^{n+1}$ **fi**
// Compute the correction α, the set of supported applicants, sample a first version of $\hat{\alpha}$
$\alpha = \bigoplus_i \texttt{PartAlpha}_{\texttt{Loc}}(i, t_k^{(i)}, y, b)$; $\mathcal{S} = \{i \mid \mathbf{d}_0[i] = 1\}$; $\forall i, \hat{\alpha}_i = \bigoplus_l r^{(l)}[i]$
if $\mathcal{S} \neq \emptyset$ **then** // If at least one person is supported, ensure $\oplus_{i\in\mathcal{S}}\hat{\alpha}_i = \alpha$
  $j = \max_{i\in\mathcal{S}} i$// Pick an arbitrary $j \in \mathcal{S}$ to change
  $\hat{\alpha}_j = \alpha \oplus \bigoplus_{i\in\mathcal{S}\setminus\{j\}} \hat{\alpha}_i$
fi // Return $\hat{\alpha}_i$ to applicant i and $\top$ to the server to indicate no problem occurred.
return $(\top, \hat{\alpha}_1, \ldots, \hat{\alpha}_n)$

Figure 7.2: The function to compute in the $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ protocol using MPC. The first input is the input of the server, and the other inputs are from the applicants (the $y^{(i)}$ and $b^{(i)}$ are supposed to be equal to y and b and are just used to ensure that the server provided coherent inputs in the MPC, and $r^{(i)} \in \{0,1\}^n$ is a string supposed to be sampled uniformly at random).

Informally, in the presence of malicious applicants, an honest server is guaranteed that with overwhelming probability the protocol will either abort, or only the applicants i knowing $w^{(i)}$ such that $\mathsf{Auth}_i(1, w^{(i)}) = 1$ will share a GHZ state, up to some unavoidable local deviation performed by supported malicious applicants on their own parts of the GHZ:

Lemma 7.4.10 (Correctness of $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ in the presence of malicious applicants)**.** *Formally, if the server is honest, and if we allow an attacker to corrupt an arbitrary subset $\mathcal{M}$ of applicants, then with overwhelming probabilities, the protocol either aborts, or the k received by the server belongs to $\mathcal{K}$, and for each applicant i, there exists w_i such that $\mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$. If the ZK protocol is also a Proof of Knowledge protocol, then the applicant "knows" w_i, in the sense that if the adversary can pass the test with non-negligible probability, there exists an extractor that can extract w_i given the applicant's circuit with non-negligible probability (this is a direct application of Definition 7.2.3)*[16].

[16]In particular, if it is impossible to forge with non-negligible probability a w_i such that $\mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$ (for instance because this w_i is a signature coming from an unforgeable signa-

Moreover, if the protocol did not abort before, at the end of the protocol, with probability $1-\delta-\mathsf{negl}(\lambda)$ *(i.e. overwhelming if* δ *is negligible), the protocol will either abort, or a state will be obtained by applicants. In this later case, if we denote by* $\rho_{A,\mathcal{M}}$ *the joint state of the honest applicants (register* A*) and of the adversary (register* $\mathcal{M}$*) obtained at the end of the protocol, then* $\rho_{A,\mathcal{M}}$ *can be written as a Completely Positive Trace Preserving (CPTP) map*[17] *applied on a* GHZ *state shared among all parties* i *such that* $\mathbf{d}_0[i]=1$*, in such a way that the CPTP map leaves untouched the qubits of the* GHZ *state owned by honest applicants* i*. In particular, if all supported parties are honest, they all share a* GHZ *state.*

Proof. The first action of the server (which is assumed to be honest here) is to run a ZK protocol to check that $\forall i, k^{(i)} \in \mathcal{K}_{\lambda,\mathtt{Loc}}$ and $\mathsf{Auth}_i(\mathbf{d}_0[i], w_i)=1$. Therefore, we can use the soundness property of the ZK protocol to claim that with overwhelming probability $\forall i, k^{(i)} \in \mathcal{K}_{\lambda,\mathtt{Loc}}$ and there exist w_i such that $\mathsf{Auth}_i(\mathbf{d}_0[i], w_i)=1$ (since the provers are the applicants, they are bounded so we can rely on both computational or statistical soundness). The fact that w_i is actually "known" to the applicant comes directly from the fact that the ZK protocol is a Proof of Knowledge and is extractable. So with overwhelming probability, $k \coloneqq (k^{(1)},\dots,k^{(n)})$ belongs to $\mathcal{K} \coloneqq \mathcal{K}^n_{\lambda,\mathtt{Loc}}$. Therefore, since the server is honest, with probability $1-\delta$ it will measure a y such that $|f_k^{-1}(y)|=2$, and due to the properties of the family f_k, the two preimages x and x' are such that $h(x)\oplus h(x')=\mathbf{d}_0$. So the state sent by the server is $|h(x)\rangle+(-1)^a|h(x')\rangle$ with $a=\langle b, x\oplus x'\rangle$. Then, the MPC protocol will be performed. If the MPC aborts, then we are already in the setting of the theorem. If the MPC does not aborts, then, due to the fact that in quantum mechanics, operations performed by two non-communicating parties commute, without any loss of generality we can assume that the honest applicants will apply the correction before the deviation of the malicious party. Moreover, since the honest corrections are unitary, we can also assume without any loss of generality that the first step of the malicious party is to apply the honest correction on the state received from the server and then deviate (eventually by starting to undo the correction). Note that we do not even ask this correction to be efficiently computable by the adversary (see Footnote 17) since we just claim that such deviation exists. Due to the definition of

ture scheme), then it means that with overwhelming probability the applicant is indeed in possession of w_i.

[17] Note that for simplicity, we just require the existence of this CPTP map, and therefore it can depend on any quantity, including k, $\mathbf{d}_0$... Therefore we do not require this map to be efficiently computable since it will not be useful for us. However, a similar "efficient" version should be derivable if we make sure our ZK protocol is extractable, i.e. that the trapdoor of each $k^{(i)}$ can be extracted by a simulator. This is however out of the scope of this thesis.

`PartInfo`, after applying the X correction, the parties i for which $\mathbf{d}_0[i] = 1$ share a state $|0\ldots0\rangle + (-1)^\alpha |1\ldots1\rangle$. Now, we have two cases:

1. If there exists at least one supported applicant which is malicious, then the proof is done: no matter what are the values of $\hat{\alpha}_i$ which will be used by the honest applicant to correct the state, we can always include in the CPTP map a first step that applies $Z^{\alpha \oplus \bigoplus_{i\in\mathcal{S}, i\notin\mathcal{M}} \hat{\alpha}_i}$ on the qubit of the malicious applicant to map the step back to a GHZ state. Again, this is possible since we just require the existence of the CPTP map. Then, any CPTP deviation can be applied on the state owned by the malicious adversary, including undoing the previous Z and X corrections.

2. If there exists no malicious supported applicant, and if the probability of having no abort and no malicious supported applicant is non-negligible[18], then with overwhelming probability we must have $\bigoplus_{i\in\mathcal{S}} \hat{\alpha}_i = \alpha$. Indeed, if it is not the case, then it is possible to distinguish the real world from the ideal world of the MPC computation. Therefore, after the Z correction the honest applicants having $\mathbf{d}_0[i] = 1$ will share a canonical GHZ, which ends the proof.

□

Lemma 7.4.11 (Blindness of $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$ in the presence of malicious applicants)**.** *If the server corrupts a set of applicants, in such a way that at least one supported applicant is not corrupted, or that no supported applicant is corrupted, then the support status of the honest applicants is hidden in the $\textit{AUTH-BLIND}^{dist}_{can}$ protocol, beyond the fact that server knows whether or not they can pass the authorization step. More formally, no adversary can win the game $\textit{IND-AUTH-BLIND}^{dist}_{can}$.*

Proof. The above game is more formally defined in `Game1`.

We will prove the above theorem by using a hybrid argument. First, we can easily see that if the adversary corrupts all applicants ($\mathcal{M} = [n]$), then it cannot win the game with probability better than $\frac{1}{2}$. Indeed, the line 2 forces $\mathbf{d}_0^{(0)} = \mathbf{d}_0^{(1)}$ (and both maps w_i are empty), therefore the view of the adversary is exactly the same for $c = 0$ and $c = 1$. So we can define a new hybrid game `Game2` in which we return **false** if $\mathcal{M} = [n]$:

Then, we can turn any adversary $\mathcal{A}$ winning `Game1` with probability p into another adversary $\mathcal{A}'$ winning `Game2` with probability p. To do so, $\mathcal{A}'$ runs first $\mathcal{A}_1$: if $\mathcal{A}_1$ returns $\mathcal{M} \neq [n]$, then $\mathcal{A}'$ continues normally with $\mathcal{A}$, otherwise if $\mathcal{M} = [n]$ then $\mathcal{A}'$ removes

[18] If on the other hand this quantity is negligible, then this second case occurs with negligible probability so it is absorbed in the $\mathsf{negl}(\lambda)$ of the theorem.

$\texttt{IND-AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}{}^{\mathcal{A}}_{\{f_k\}}(\lambda)$

$(\mathcal{M}, \mathbf{d}_0^{(0)}, \{(i, w_i^{(0)})\}_{i\in[n]\setminus\mathcal{M}}), \mathbf{d}_0^{(1)}, \{(i, w_i^{(1)})\}_{i\in[n]\setminus\mathcal{M}}) \leftarrow \mathcal{A}_1(1^\lambda)$

if $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$ **then return false fi**

if $(\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)} = 1)$ and $((\forall i \notin \mathcal{M}, \mathbf{d}_0^{(0)}[i] = 0)$ or $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(1)}[i] = 0))$

then return false fi

// Check that the adversary did not gave wrong witnesses w_i:

if $\exists i \in [n], \exists c \in \{0,1\}, \mathsf{Auth}_i(\mathbf{d}_0^{(c)}[i], w_i^{(c)}) \neq 1$ **then return false fi**

$c \xleftarrow{\$} \{0,1\}; \mathbf{d}_0 \coloneqq \mathbf{d}_0^{(c)}; w_i \coloneqq w_i^{(c)}$

Run with $\mathcal{A}_2$ the protocol $\texttt{AUTH-BLIND}^{\texttt{dist}}_{\texttt{can}}$.

$\tilde{c} \leftarrow \mathcal{A}_3$

return $\tilde{c} = c$

$\texttt{Game1}^{\mathcal{A}}(\lambda)$

$(\mathcal{M}, \mathbf{d}_0^{(0)}, \{(i, w_i^{(0)})\}_{i\in[n]\setminus\mathcal{M}}), \mathbf{d}_0^{(1)}, \{(i, w_i^{(1)})\}_{i\in[n]\setminus\mathcal{M}}) \leftarrow \mathcal{A}_1(1^\lambda)$

if $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$ **then return false fi**

if $(\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)} = 1)$ and $((\forall i \notin \mathcal{M}, \mathbf{d}_0^{(0)}[i] = 0)$ or $(\forall i \notin \mathcal{M}, \mathbf{d}_0^{(1)}[i] = 0))$

then return false fi

// Check that the adversary did not gave wrong witnesses w_i:

if $\exists i \in [n], \exists c \in \{0,1\}, \mathsf{Auth}_i(\mathbf{d}_0^{(c)}[i], w_i^{(c)}) \neq 1$ **then return false fi**

$c \xleftarrow{\$} \{0,1\}; \mathbf{d}_0 \coloneqq \mathbf{d}_0^{(c)}; w_i \coloneqq w_i^{(c)}$

$\forall i \notin \mathcal{M}, (k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$

$\mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$

for $i \notin \mathcal{M}$ **do**

Prove in ZK with $\mathcal{A}_{2,i}$ that $\mathtt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)}) \wedge \mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$

endfor

if $\mathcal{A}_2$ aborts **then** wait for $\tilde{c}$ from $\mathcal{A}_2$. **return** $\tilde{c} = c$ **fi**

$(y, b) \leftarrow \mathcal{A}_3$

Compute in a MPC way the `CombineAlpha` function, where $\mathcal{A}_4$ controls adversaries in $\mathcal{M}$.

// All others operations are not sent to the adversary, and operations

// applied on the quantum state do not change anything due to non-signaling.

$\tilde{c} \leftarrow \mathcal{A}_5$

return $\tilde{c} = c$

$\mathtt{Game2}^{\mathcal{A}}(\lambda)$

// Just update the line 2 of Game1 as follows:

2 : **if** $\mathcal{M} = [n]$ or $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$ **then return false fi**

// Rest is like Game1...

one element of $\mathcal{M}$ (of course, $[n]$ is assumed to be non empty...), and $\mathcal{A}'$ aborts when $\mathcal{A}_2$ is supposed to run, and output a random $\tilde{c}$. Therefore, we have:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathtt{Game1}^{\mathcal{A}}\right](\lambda) = \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\mathtt{Game2}^{\mathcal{A}'}\right](\lambda) \tag{7.17}$$

The second hybrid game that we define is the same as the game Game2, except that we replace lines 14 to 15 with one line $(\tilde{c}, \vec{y}) \leftarrow \mathsf{REAL}_{\Pi,\mathcal{A}'}(\lambda, \vec{x}, \rho_3)$, where ρ_3 is the final internal state of the adversary $\mathcal{A}_3$, $\vec{x}$ contains the honest inputs of the MPC computation for the non-corrupted adversaries and dummy inputs for the corrupted adversaries (we will ignore them anyway), and $\mathcal{A}'$ is the adversary that outputs the corrupted set $\mathcal{M}$, that runs $\mathcal{A}_4(\rho_3)$ followed by $\tilde{c} \leftarrow \mathcal{A}_5(\rho_4)$, where ρ_4 is the final internal state of $\mathcal{A}_4$, and that finally returns $\tilde{c}$. This defines a new game Game3:

$\mathtt{Game3}^{\mathcal{A}}$

14 : ~~Compute in a MPC way the CombineAlpha function, where $\mathcal{A}_4$ controls adversaries in $\mathcal{M}$.~~

15 : ~~$\tilde{c} \leftarrow \mathcal{A}_5$~~ $(\tilde{c}, \vec{y}) \leftarrow \mathsf{REAL}_{\Pi,\mathcal{A}'}(\lambda, \vec{x}, \rho_3)$

Since this is perfectly equivalent from the point of view of the adversary (due to the definition or REAL), the probability of winning these two games are exactly the same: $\Pr\left[\mathtt{Game2}^{\mathcal{A}}\right] = \Pr\left[\mathtt{Game3}^{\mathcal{A}}\right]$. Now, because the MPC protocol is secure, there exists a simulator Sim fitting Definition 7.2.6. Therefore, we can now define a new game Game4, in which we replace the real world with the ideal world:

$\mathtt{Game4}^{\mathcal{A}}$

15 : ~~$(\tilde{c}, \vec{y}) \leftarrow \mathsf{REAL}_{\Pi,\mathcal{A}'}(\lambda, \vec{x}, \rho_3)$~~ $(\tilde{c}, \vec{y}) \leftarrow \mathsf{IDEAL}_{\mathtt{CombineAlpha},\mathsf{Sim}}(\lambda, \vec{x}, \rho_3)$

Then, we have

$$\Pr\left[\mathsf{REAL}_{\Pi,\mathcal{A}'}(\lambda, \vec{x}, \rho_3)[0] = c\right] \leq \Pr\left[\mathsf{IDEAL}_{\mathtt{CombineAlpha},\mathsf{Sim}}(\lambda, \vec{x}, \rho_3)[0] = c\right] + \mathsf{negl}(\lambda) \tag{7.18}$$

(otherwise we could distinguish between the real and ideal worlds), and therefore $\Pr\left[\mathtt{Game3}^{\mathcal{A}}\right] \leq \Pr\left[\mathtt{Game4}^{\mathcal{A}}\right] + \mathsf{negl}(\lambda)$. Now, we can define CombineRandom, which

is an adaptation of CombineAlpha that does not depend anymore on the secret values of the honest parties: We can now define a new game Game5 in which we substitute the

Algorithm 5 CombineRandom$((k, y, b), (t_k^{(1)}, \mathbf{d}_0[1], r^{(1)}, k^{(1)}, y^{(1)}, b^{(1)}), \ldots$
$\ldots, (t_k^{(n)}, \mathbf{d}_0[n], r^{(n)}, k^{(1)}, y^{(n)}, b^{(n)}))$

1 : // Check if the input are honestly prepared
2 : **if** $k \neq (k^{(1)}, \ldots, k^{(n)})$ or $\exists i, y^{(i)} \neq y$
3 : or $b^{(i)} \neq b$ or $\exists i \in \mathcal{M}, \neg\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)})$ **then**
4 : **return** $\perp^{n+1}$ **fi**
5 : $\forall i, \hat{\alpha}_i = \bigoplus_l r^{(l)}[i]$
6 : **return** $(\top, \hat{\alpha}_1, \ldots, \hat{\alpha}_n)$

CombineAlpha function with the CombineRandom function:

$\texttt{Game5}^{\mathcal{A}}$

1 : $(\tilde{c}, \vec{y}) \leftarrow \mathsf{IDEAL}_{\sout{\texttt{CombineAlpha}}\texttt{CombineRandom},\mathsf{Sim}}(\lambda, \vec{x}, \rho_3)$

Then, we have $\Pr\left[\texttt{Game4}^{\mathcal{A}}\right] = \Pr\left[\texttt{Game5}^{\mathcal{A}}\right]$. Indeed, by construction, the inputs of the honest parties always pass the CheckTrapdoor test, so removing this test for the honest parties cannot help the adversary to distinguish the two games. Moreover, since at least one applicant is honest, the string $\bigoplus_l r^{(l)}$ is indistinguishable from a random string. Therefore, we can use the same trick used already in the proof of Lemma 7.4.9: the condition line 3 gives us two cases.

- If all malicious applicants are not supported, then, since the output of honest applicants are never given back the adversary, we do not need to update $\hat{\alpha}_j$
- Similarly, if at least one honest applicant is supported, then instead of updating $\hat{\alpha}_j$, we can update the $\hat{\alpha}_j$ or this applicant... But since the output of honest applicants are never given to the adversary, we do not even need to update it.

Therefore, $\Pr\left[\texttt{Game4}^{\mathcal{A}}\right] = \Pr\left[\texttt{Game5}^{\mathcal{A}}\right]$. In the next hybrid, we are going to remove completely the MPC computation, and the previous line that can now be merged in a single one:

$\texttt{Game6}^{\mathcal{A}}$

12 : ~~**if** $\mathcal{A}_2$ aborts **then** wait for $\tilde{c}$ from $\mathcal{A}_2$. **return** $\tilde{c} = c$ **fi**~~
13 : ~~$(y, b) \leftarrow \mathcal{A}_3$~~
12 : ~~$(\tilde{c}, \vec{y}) \leftarrow \mathsf{IDEAL}_{\texttt{CombineRandom},\mathsf{Sim}}(\lambda, \vec{x}, \rho_3)$~~ $\tilde{c} \leftarrow \mathcal{A}_3$

The reason is that now, since the `CombineRandom` function does not depend on any secret own by honest parties, the input of honest parties can be replaced with wrong trapdoors $t_k^{(i)}$ and $\mathbf{d}_0[i]$. Therefore, since the adversary knows already $k^{(i)}$, it can simulate locally the ideal world. More precisely, from an adversary $\mathcal{A}$ winning the game $\texttt{Game5}^{\mathcal{A}}$ with probability p, we can create another adversary $\mathcal{A}'$ winning the game $\texttt{Game6}^{\mathcal{A}}$ with the same probability p: $\mathcal{A}'$ will run $\mathcal{A}_1$ and $\mathcal{A}_2$ against the challenger, keeping locally the $k^{(i)}$. If $\mathcal{A}_2$ aborts and sends $\tilde{c}$, then it returns $\tilde{c}$ directly. Otherwise $\mathcal{A}'$ runs as a blackbox $\mathcal{A}_3$ to obtain (y, b), and then it locally runs Sim to compute $\mathsf{IDEAL}_{\texttt{CombineRandom},\mathsf{Sim}}(\lambda, \vec{x}, \rho_3)$, by feeding the input of honest parties input $\vec{x}$ with the $k^{(i)}$ that it got before, $y^{(i)} := y$, $b^{(i)} := b$, $r^{(i)} \xleftarrow{\$} \{0,1\}^n$ and since the values of $t_k^{(i)}$ do not matter anymore, it can put any value here. Finally $\mathcal{A}'$ outputs the $\tilde{c}$ obtained from the simulation IDEAL. Therefore, since we do not change what is done, but who's doing what, we get:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\texttt{Game5}^{\mathcal{A}}\right](\lambda) = \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\texttt{Game6}^{\mathcal{A}'}\right](\lambda) \tag{7.19}$$

So now, the game `Game6` is exactly like `Game1` except that the lines 12–15 are replaced with a single line $\tilde{c} \leftarrow \mathcal{A}_3$. We will now do a similar strategy to remove the ZK protocol (line 9). The first step is to formalize this line (we will also merge it with the next line). We define $\mathcal{R}_{\mathcal{L}}$ as the relation $(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)}) \in \mathcal{R}_{\mathcal{L}}(k^{(i)})$ iff $\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)}) \wedge \mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$, and P the honest ZK prover associated with $\mathcal{R}_{\mathcal{L}}$. If we define ρ_1 as the internal state at the end of $\mathcal{A}_1$, and $\mathsf{V}^*(\{k^{(i)}\}_{i\notin\mathcal{M}}, \rho_1) := (\rho_2 \leftarrow \mathcal{A}_2(\rho_1); \tilde{c} \leftarrow \mathcal{A}_3(\rho_2))$ then we can merge the line 12 of `Game6` with the line 9 as follows:

$\texttt{Game7}^{\mathcal{A}}$

8 : $\rho \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$ // We just explicit the internal state after $\mathcal{A}_2$
9 : **for** $i \notin \mathcal{M}$ **do**
10 : ~~Prove in ZK with $\mathcal{A}_{2,i}$ that $\texttt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)}) \wedge \mathsf{Auth}_i(\mathbf{d}_0[i], w_i) = 1$~~
$\rho \leftarrow \mathsf{OUT}_{\mathcal{A}_{2,i}}\langle \mathsf{P}(\mathbf{d}_0[i], t_k^{(i)}, w_i), \mathcal{A}_{2,i}(\rho)\rangle(k^{(i)})$
11 : **endfor**

Since both games are exactly identical (up to the notation), we get $\Pr\left[\texttt{Game6}^{\mathcal{A}}\right] = \Pr\left[\texttt{Game7}^{\mathcal{A}}\right]$. Now, due to the fact that the MPC protocol respects the property *Quantum Zero Knowledge* defined in Definition 7.2.2, there exist for all $i \notin \mathcal{M}$ a simulator Sim_i fitting Definition 7.2.2. To be completely formal, one should define a series of games in which we replace in the loop only one OUT at a time by the simulated version, and we can then claim that the probability of having $\tilde{c} = c$ in each hybrid game is negligibly close to the probability of having $\tilde{c} = c$ in the first game, otherwise we could distinguish between the real world and the ideal world. This gives us at the end a new game `Game8`:

$\mathsf{Game8}^{\mathcal{A}}$

10 : ~~$\rho \leftarrow \mathsf{OUT}_{\mathcal{A}_{2,i}}\langle \mathsf{P}(\mathbf{d}_0[i], t_k^{(i)}, w_i), \mathcal{A}_{2,i}(\rho)\rangle(k^{(i)})$~~ $\rho \leftarrow \mathsf{Sim}_i(k^{(i)}, \mathsf{V}^*, \rho)$

And using the above argument, $\Pr\left[\mathsf{Game7}^{\mathcal{A}}\right] \leq \Pr\left[\mathsf{Game8}^{\mathcal{A}}\right] + \mathsf{negl}(\lambda)$. Now, the simulators Sim_i can be fully simulated by the adversary since there is no more secret information. So, exactly like we did for the MPC computation, we can move the loop into the adversary:

$\mathsf{Game9}^{\mathcal{A}}$

8 : ~~$\rho \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$~~ $\tilde{c} \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$

9 : ~~**for** $i \notin \mathcal{M}$ **do**~~

10 : ~~$\rho \leftarrow \mathsf{Sim}(k^{(i)}, \mathsf{V}^*, \rho)$~~

11 : ~~**endfor**~~

12 : ~~$\tilde{c} \leftarrow \mathcal{A}_3$~~

and we get:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathsf{Game8}^{\mathcal{A}}\right](\lambda) = \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\mathsf{Game9}^{\mathcal{A}'}\right](\lambda) \tag{7.20}$$

So now, Game9 is like Game1 except that all lines starting from line 8 are replaced with a single line $\tilde{c} \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$. We see now that the conditions line 5 and line 3 can only decrease the probability of winning the game. Therefore we can remove them (as well as w_i's which are not used anymore). This gives us this new game (after removing empty lines):

$\mathsf{Game10}^{\mathcal{A}}(\lambda)$

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$

2 : **if** $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$ **then return false fi**

3 : $c \xleftarrow{\$} \{0,1\}; \mathbf{d}_0 := \mathbf{d}_0^{(c)}$

4 : $\forall i \notin \mathcal{M}, (k^{(i)}, t_k^{(i)}) \leftarrow \mathsf{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$

5 : $\tilde{c} \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i\notin\mathcal{M}})$

6 : **return** $\tilde{c} = c$

Since we only increase the probability of success, we have:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathsf{Game9}^{\mathcal{A}}\right](\lambda) \leq \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\mathsf{Game10}^{\mathcal{A}'}\right](\lambda) \tag{7.21}$$

$\mathtt{Game11}^{\mathcal{A}}$

4 : ~~$\forall i \notin \mathcal{M}$~~$\forall i \in [n], (k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$

5 : $\tilde{c} \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i \notin \mathcal{M} \in [n]})$

Similarly, we can decide to give more advices to the adversary, by running $\mathtt{Gen}_{\mathtt{Loc}}$ on all $i \in [n]$ instead of only on the $i \notin \mathcal{M}$:

Since we give more advices to the adversary, its probability of winning the game can only increase (it can always decide to drop this additional information). Therefore

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathtt{Game10}^{\mathcal{A}}\right](\lambda) \leq \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\mathtt{Game11}^{\mathcal{A}'}\right](\lambda) \tag{7.22}$$

However, since $\mathtt{Gen}$ is defined as the concatenation of $\mathtt{Gen}_{\mathtt{Loc}}$, then this game is actually exactly the $\mathtt{IND\text{-}DO}^{\mathcal{A}}_{\mathtt{Gen}}(\lambda)$ game defined Section 4.2, and by assumption, the probability of winning this game is smaller than $\frac{1}{2} + \mathsf{negl}(\lambda)$. Therefore, we also have:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathtt{Game1}^{\mathcal{A}}\right](\lambda) \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{7.23}$$

which ends the proof. □

7.4.3 Generic construction to create distributable δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable primitives from δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable primitives

We prove in this section that we can create a distributable δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable family of functions from a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family having a small additional assumption (which our construction in Chapter 5 has).

Theorem 7.4.12. *Let $\delta \in [0,1]$, and $\{f_k\}_{k\in\mathcal{K}}$ be a δ-$\mathsf{GHZ}^{\mathsf{H}}$ capable family[19] of functions, such that $f_k(x)$ can be written as $f_k((c,\bar{x}))$ with $c \in \{0,1\}$ a bit labelling the preimage[20], i.e. such that when a given y has exactly two preimages, one preimage has the form $(0,\bar{x})$ and the other $(1,\bar{x}')$. Then there exists a family $\{f'_k\}_{k\in\mathcal{K}'}$ which is a distributable δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable family with $\delta' = 1-(1-\delta)^n \leq \delta n$. In particular, if δ is negligible (and n polynomial) then δ' is negligible. Moreover, if the family $\{f_k\}$ admits a circuit $\mathtt{H.CheckTrapdoor}_\lambda(\mathbf{d}_0, t_k, k)$ that returns 1 iff t_k is the trapdoor of k, $k \in \mathcal{K}$ and $\mathbf{d}_0 = \mathbf{d}_0(t_k)$, then there exists a function $\mathtt{CheckTrapdoor}$ for $\{f'_k\}$ having the properties from Definition 7.4.3.*

[19] In fact we only require this function to work when $\mathbf{d}_0$ is a single bit.

[20] This is quite similar to the concept of claw-free functions used in [Mah18a].

The family $\{f'_k\}_{k\in\mathcal{K}'}$ can be obtained by generating n independent functions in $\{f_k\}$ (one for each $\mathbf{d}_0[i]$). More precisely, we define in Figure 7.3 the precise construction (where `H.Gen`, `H.Enc`, `H.Invert` and `H.Eval` are coming from the family $\{f_k\}$).

$\mathtt{Gen}_{\mathtt{Loc}}(1^\lambda, \mathbf{d}_0[i])$

1 : $(k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{H.Gen}(1^\lambda, \mathbf{d}_0[i])$
2 : **return** $(k^{(i)}, t_k^{(i)})$

$\mathtt{Eval}_{\mathcal{P}}((k^{(1)}, \ldots, k^{(n)}), (c, \bar{x}^{(1)}, \ldots, \bar{x}^{(n)}))$

1 : **return** $(\mathtt{H.Eval}(k^{(1)}, (c, \bar{x}^{(1)})),$
2 : $\quad \ldots, \mathtt{H.Eval}(k^{(n)}, (c, \bar{x}^{(n)})))$

$\mathtt{PartInfo}_{\mathtt{Loc}}(t_k^{(i)}, y)$

1 : **if** $\mathbf{d}_0(t_k^{(i)}) = 0$ **then return** ✗ **fi**
2 : $\{(0, \bar{x}), (1, \bar{x}')\} \leftarrow \mathtt{H.Invert}(t_k^{(i),y})$
3 : **if** $\bar{x} = \perp$ or $\bar{x}' = \perp$ **then return** $\perp$
4 : **return** $\mathtt{H}.h((0, \bar{x}))$

$h((x^{(1)}, \ldots, x^{(n)}))$

1 : **return** $h(x^1)|\ldots|h(x^n)$

$\mathtt{PartAlpha}_{\mathtt{Loc}}(i, t_k^{(i)}, y, b)$

1 : $(y^{(1)}, \ldots, y^{(n)}) \coloneqq y$
2 : $(b_c, b^{(1),\ldots,b^{(n)}}) \coloneqq b$
3 : $\{(0, \bar{x}), (1, \bar{x}')\} \leftarrow \mathtt{H.Invert}(t_k^{(i),y})$
4 : **if** $\bar{x} = \perp$ or $\bar{x}' = \perp$ **then return** $\perp$ **fi**
5 : **if** $i = 1$ **then return** $b_c \oplus \langle b^{(i)}, \bar{x} \oplus \bar{x}' \rangle$
6 : **else return** $\langle b^{(i)}, \bar{x} \oplus \bar{x}' \rangle$ **fi**

$\mathtt{CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)})$

1 : **return** $\mathtt{H.CheckTrapdoor}_\lambda(\mathbf{d}_0[i], t_k^{(i)}, k^{(i)})$

Figure 7.3: Construction of a distributable δ'-$\mathsf{GHZ}^{\mathsf{can}}$ capable family.

Proof. Most of the properties are simple to check. We just precise the δ'-2-to-1 proof and blindness. First, we show that the function f'_k are δ'-2-to-1 with $\delta = 1 - (1-\delta)^n$. Let $\#_2(f)$ be the number of images having exactly 2 preimages by f (we will call this kind of preimages "twin"), and $|\mathcal{K}|$ the number of elements in $\mathcal{K}$. Then by definition, for all k, $1 - \delta \leq \#_2(f_k)/|\mathcal{X}|$. Let $k' = (k^{(1)}, \ldots, k^{(n)}) \in \mathcal{K}^n = \mathcal{K}'$, we want to show that $1 - \delta' \coloneqq (1-\delta)^n \leq \#_2(f'_{k'})/(|\mathcal{X}'|)$. First, we compute $\#_2(f_{k'})$. Because of the assumption of the shape $(0, \bar{x})$ and $(1, \bar{x}')$ of all the couples of preimages, we can define for any $k^{(i)}$ the sets

$$A_0^{(i)} \coloneqq \{\bar{x} \mid f_{k^{(i)}}^{-1}(f_{k^{(i)}}(x)) = \{(0, \bar{x}), (1, \bar{x}')\}\} \tag{7.24}$$

and

$$A_1^{(i)} \coloneqq \{\bar{x}' \mid f_{k^{(i)}}^{-1}(f_{k^{(i)}}(x)) = \{(0, \bar{x}), (1, \bar{x}')\}\} \tag{7.25}$$

Moreover, due to this same condition, we have $|A_0^{(i)}| = |A_1^{(i)}| = \frac{1}{2}\#_2(f_{k^{(i)}})$. Now, we compute a lower bound on the number of twin preimages of $f_{k'}$. Let $(\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}) \in$

$A_0^{(1)} \times \cdots \times A_0^{(n)}$: then for all i there exists a unique $\bar{x}^{(1)\prime} \in A_1^{(i)}$ such that $f_{k^{(i)}}((0, \bar{x}^{(i)})) = f_{k^{(i)}}((1, \bar{x}^{(i)})$. So

$$f_{k'}(0, \bar{x}^{(1)}, \ldots, \bar{x}^{(n)}) = f_{k^{(1)}}(0, \bar{x}^{(1)}) | \ldots | f_{k^{(n)}}(0, \bar{x}^{(n)}) \tag{7.26}$$
$$= f_{k^{(1)}}(1, \bar{x}^{(1)\prime}) | \ldots | f_{k^{(n)}}(1, \bar{x}^{(n)\prime}) \tag{7.27}$$
$$= f_{k'}(1, \bar{x}^{(1)\prime}, \ldots, \bar{x}^{(n)\prime}) \tag{7.28}$$

So we found at least one different preimage, and due to the uniqueness of the above $\bar{x}^{(1)\prime}$ it is the only second preimage. Therefore:

$$\frac{\#_2(f_{k'})}{|\mathcal{X}'|} = \frac{2 \times |A_0^{(1)}| \times \cdots \times |A_0^{(n)}|}{2\left(\frac{|\mathcal{X}|}{2}\right)^n} \tag{7.29}$$
$$= \frac{\#_2(f_{k^{(1)}}) \times \cdots \times \#_2(f_{k^{(n)}})}{(|\mathcal{X}|)^n} \tag{7.30}$$
$$\geq (1-\delta)^n \tag{7.31}$$

Which concludes the proof.

To prove the inequality, we use the Bernoulli's inequality: since $\delta \in [0, 1]$ and n is a non-negative integer, we get: $(1-\delta)^n \geq 1 - \delta n$, so

$$\delta' = 1 - (1-\delta)^n \leq 1 - (1 - \delta n) = \delta n \tag{7.32}$$

Since the keys of $\{f'_k\}$ are keys of $\{f'_k\}$ (except that $\mathbf{d}_0$ is a single bit), the properties of `CheckTrapdoor` come directly from the properties of `H.CheckTrapdoor`. All the other correctness properties are true by construction.

The security is quite intuitive: since all trapdoors are independently sampled, if one can learn information about the d_0 sampled by another party, then it can break the $\texttt{IND-DO}^{\mathcal{A}}_{\texttt{Gen}}(\lambda)$ game of f_k. More formally, because of the properties on $\texttt{Gen}_{\texttt{Loc}}$, the game $^{\mathcal{A}}_{\texttt{Gen},\texttt{PartInfo}}(\lambda)$ can equivalently be rewritten as follows:

Then, we can define the following game in which the sampling of malicious $t_k^{(i)}$, the computing of $\texttt{PartInfo}_{\texttt{Loc}}$ and the initial condition are removed:

Then, because $\mathcal{A}_2$ in Game2 can do itself the sampling and computing done in Game1, and because the removing the condition line 2 can only increase the probability of winning the game, we have:

$$\max_{\text{QPT}\mathcal{A}} \Pr\left[\texttt{Game1}^{\mathcal{A}}\right](\lambda) \leq \max_{\text{QPT}\mathcal{A}'} \Pr\left[\texttt{Game2}^{\mathcal{A}'}\right](\lambda) \tag{7.33}$$

Then, we define a series a hybrid games in which we gradually replace the $\texttt{H.Gen}(1^\lambda, d_0^{(c)}[i])$'s with $\texttt{H.Gen}(1^\lambda, 0)$, which $\mathcal{Z}$ starting from $\emptyset$ until $\mathcal{Z} = \mathcal{M}$.

Game1

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : **if** $\exists i \in \mathcal{M}, \mathbf{d}_0^{(0)}[i] \neq \mathbf{d}_0^{(1)}[i]$: **return false fi**
3 : $c \xleftarrow{\$} \{0,1\}$
4 : $\forall i, (k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{H.Gen}(1^\lambda, \mathbf{d}_0^{(c)}[i])$
5 : $y \leftarrow \mathcal{A}_2(k^{(1)}, \ldots, k^{(n)})$
6 : $\forall i, v[i] \leftarrow \mathtt{PartInfo}_{\mathtt{Loc}}(i, t_k^{(i)}, y)$
7 : $\tilde{c} \leftarrow \mathcal{A}_3(\{(i, v[i])\}_{i \in \mathcal{M}})$
8 : **return** $\tilde{c} = c$

Game2

1 : $(\mathcal{M}, \mathbf{d}_0^{(0)}, \mathbf{d}_0^{(1)}) \leftarrow \mathcal{A}_1(1^\lambda)$
2 : $c \xleftarrow{\$} \{0,1\}$
3 : $\forall i \in \mathcal{M}, (k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{H.Gen}(1^\lambda, \mathbf{d}_0^{(c)}[i])$
4 : $\tilde{c} \leftarrow \mathcal{A}_2(\{k^{(i)}\}_{i \in \mathcal{M}})$
5 : **return** $\tilde{c} = c$

Game3$_\mathcal{Z}$

1 : ~~$\forall i \in \mathcal{M}, (k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{H.Gen}(1^\lambda, \mathbf{d}_0^{(c)}[i])$~~
$\forall i \in \mathcal{M}$, **if** $i \in \mathcal{Z}$ **then** $x \coloneqq 0$ **else** $x \coloneqq \mathbf{d}_0^{(c)}[i]$ **fi** ; $(k^{(i)}, t_k^{(i)}) \leftarrow \mathtt{H.Gen}(1^\lambda, x)$

This is possible because the game IND-D0 of $\mathtt{H.Gen}$ can be seen as a IND-CPA secure encryption (where $\mathtt{Gen}$ is the concatenation of the key generation and encryption of $\mathbf{d}_0$), itself equivalent to semantic security: i.e. the ciphertext does not give any advice on the clear text. Therefore we can replace the clear text with 0 for example, which gives for all j:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[\mathtt{Game3}_{\mathcal{Z}}^{\mathcal{A}}\right](\lambda) \leq \max_{\mathsf{QPT}\mathcal{A}'} \Pr\left[\mathtt{Game3}_{\mathcal{Z}\cup\{j\}}^{\mathcal{A}'}\right](\lambda) \tag{7.34}$$

At the end of the hybrid series, when $\mathcal{Z} = \mathcal{M}$, we get a final game where no information about c are given to the adversary: it is therefore impossible to win this game $\mathtt{Game3}_\mathcal{M}$ with probability better than $\frac{1}{2}$. Therefore we get:

$$\max_{\mathsf{QPT}\mathcal{A}} \Pr\left[{}^{\mathcal{A}}_{\mathtt{Gen},\mathtt{PartInfo}}\right](\lambda) \leq \frac{1}{2} + \mathsf{negl}(\lambda) \tag{7.35}$$

Which ends the proof.

Finally, the security of the game implies the security of IND-DO since when $\mathcal{M} = \emptyset$, both games are equivalent. □

7.5 Discussions and Open Questions

In this chapter, we saw that classical-client RSP protocols can have surprising connections with Zero-Knowledge: they allow us to perform Non-Destructive and Non-Interactive Zero-Knowledge proofs on Quantum States (NIKZoQS), which seems to be impossible to do with more standard quantum protocols. Moreover, we can extend our GHZ-QFactory protocol to a multi-party setting by distributing a GHZ state among multiple parties: revealing information about the shared state does not significantly decrease the security. We can also force the applicants to prove that if they share a part of the GHZ state, then they know secrets. This is done without revealing to anybody whether they are part of the GHZ state of not.

This may have multiple applications, as already discussed in Section 7.1.3. One could notably study the obtainable guarantees, on a per-protocol basis, notably if the source can collaborate with some applicants (as already discussed, if the source cannot collaborate with the applicants the security proof should be direct). One could also formalize our sketched protocol and proof concerning quantum "onion routing" and anonymous quantum secret sharing (for instance in this later protocol, one must make sure that the output of the measurement done during the share does not leak to people outside of the GHZ state: therefore, an additional MPC step may be required, or in the presence of a single party sharing the quantum state, one may prefer to use its public key to encrypt the value of the measurement).

It would also be interesting to see if NIZKoQS and quantum languages could also have other more fundamental implications, potentially linked with complexity theory. Notably, being able to characterize completely the set of quantum languages for which there exist NIZKoQS proofs could be of great interest: using [Mah18a] we can certainly prove that a given state is of the form $\mathbf{X}^{\mathbf{a}}\mathbf{Z}^{\mathbf{b}}|\phi\rangle$ for any ϕ obtainable using an efficient quantum circuit (to which we can also add ZK proofs on the quantum circuit itself), but so far we do not know if it is possible or not to prove statement on the one-time padded state directly. Note that since the entanglement between the qubits is independent of the one-time pad, it means that we should be able to prove any property on the entanglement of any given state.

Chapter 8

Conclusion

"Is there no end to his disguises of benevolence?"

— Margaret Atwood, *The Handmaid's Tale*

Our journey is coming to an end. Across this thesis, we saw that post-quantum cryptography can be used to surprisingly enhance quantum cryptography, up to the point of revealing previously unimaginable applications. It is now time to quickly summarize our findings and describe some ongoing projects I am currently working on.

The first fundamental discovery of our work is the creation of a series of protocols called QFactory, realizing for the first time a core atomic primitive: classical-client Remote State Preparation (RSP). It allows a purely classical client to prepare on a remote quantum server a quantum state, in such a way that the classical description of that state is only known to the client. We show how to produce various sets of states, including a set of states which is universal and can be used to produce any state. Moreover, we also show how to produce large multi-qubits GHZ states more efficiently.

This modular primitive can be used to replace quantum channels by classical channels, which is particularly interesting as quantum channels are extremely hard to deploy widely and are not always compatible with the promising technologies used to build quantum computers. Notably, we showed how our QFactory protocol can be used modularly inside the UBQC [BFK09] protocol to allow a classical client to delegate a quantum computation to an untrusted remote quantum server without revealing the input, the output and the algorithm in use.

In order to obtain such a functionality, we rely on the hardness of the Learning With Errors (LWE) problem. We explicitly provide two constructions, depending on whether we want to rely on the hardness of LWE with superpolynomial noise ratio, or on the more standard LWE with polynomial noise ratio[1]. To that end, we provide a detailed analysis in order to find a set of parameters fulfilling all the requirements of our constructions.

We also showed that classical-client RSP protocols cannot be proven secure in a general composable framework: as a result, all our security proofs are stated in the weaker game-based security model. Our result is very general and also apply to approximate "noisy" RSP resources. Moreover, we proved that if we replace the quantum channel of the UBQC protocol with any classical-client RSP protocol, then the resulting protocol cannot be generally composably secure.

While QFactory can be used in order to turn quantum clients into classical clients in existing protocols, we also showed that it can be useful to obtain new—a priori unrelated—protocols that were unimaginable before. Notably, we showed that a party can send a quantum state to a recipient while proving non-interactively and non-destructively highly non-trivial properties on the received quantum state (we call this functionality NIZKoQS). In particular, we can prove any property on the set of entangled qubits of a transmitted quantum state, potentially linked with classical secret data.

Finally, we showed how our QFactory protocol can be extended to a multi-party setting. Combined with NIZKoQS, it proves useful to allow an untrusted source to distribute a GHZ state in such a way that the participants can be filtered in an arbitrary way, so that the identity of the filtered participants remains hidden to all parties, including the source. In particular, this can be used to filter and anonymize the list of participants in any protocol involving shared GHZ states. We also mention potential use cases, notably to obtain anonymous filtered quantum secret sharing or quantum onion routing protocols.

Regarding ongoing projects, we already presented in the concluding section of each chapter some remaining open questions. More recently, I've also been working on various problems, for instance to use QFactory to avoid the impossibility result on classical Position Based Verification (answered meanwhile in [LLQ21]) or on unrelated methods to obtain One-Time Memories (this project is still too fresh to be presented in this thesis).

[1] In that latter case, we need to rely on an additional conjecture for the security proof to hold.

The Concluding Word

Nature is fair and in favor of equity.
Sadly, fairness has a price called "efficiency".

Bibliography

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 7779, October 2019. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5 (cited on page 3).

[Aar05] Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2063):3473–3482, November 2005. DOI: 10.1098/rspa.2005.1546 (cited on page 24).

[ABE08] Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive Proofs For Quantum Computations. November 18, 2008. arXiv: 0810.5375 [quant-ph] (cited on page 4).

[ABG+21] Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-Quantum Multi-Party Computation. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, Lecture Notes in Computer Science, pages 435–464, Cham. Springer International Publishing, 2021. ISBN: 978-3-030-77870-5. DOI: 10.1007/978-3-030-77870-5_16 (cited on pages 186, 187).

[ACC+21] Bar Alon, Hao Chung, Kai-Min Chung, Mi-Ying Huang, Yi Lee, and Yu-Ching Shen. Round Efficient Secure Multiparty Quantum Computation with Identifiable Abort. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, Lecture Notes in Computer Science, pages 436–466, Cham. Springer International Publishing, 2021. ISBN: 978-3-030-84242-0. DOI: 10.1007/978-3-030-84242-0_16 (cited on page 6).

[ACG+19] Scott Aaronson, Alexandru Cojocaru, Alexandru Gheorghiu, and Elham Kashefi. Complexity-Theoretic Limitations on Blind Delegated Quantum Computation. Version 1.0:13 pages, 2019. In collaboration with Michael Wagner. DOI: 10.4230/LIPICS.ICALP.2019.6 (cited on page 7).

[ACG+20] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive Classical Verification of Quantum Computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 153–180, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-64381-2. DOI: 10.1007/978-3-030-64381-2_6 (cited on page 9).

[ACP+09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, Lecture Notes in Computer Science, pages 595–618, Berlin, Heidelberg. Springer, 2009. ISBN: 978-3-642-03356-8. DOI: 10.1007/978-3-642-03356-8_35 (cited on pages 113, 117).

[Ajt99] Miklós Ajtai. Generating Hard Instances of the Short Basis Problem. In Jiří Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 1–9, Berlin, Heidelberg. Springer, 1999. ISBN: 978-3-540-48523-0. DOI: 10.1007/3-540-48523-6_1 (cited on page 117).

[AP11] Joël Alwen and Chris Peikert. Generating Shorter Bases for Hard Random Lattices. *Theory of Computing Systems*, 48(3):535–553, April 1, 2011. ISSN: 1433-0490. DOI: 10.1007/s00224-010-9278-3 (cited on page 117).

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3), January 1, 2015. ISSN: 1862-2976, 1862-2984. DOI: 10.1515/jmc-2015-0016 (cited on page 113).

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, December 1, 1993. ISSN: 1432-1807. DOI: 10.1007/BF01445125 (cited on page 114).

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, India, 1984 (cited on pages xvi, 71).

[BBC+93] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters*, 70(13):1895–1899, March 29, 1993. DOI: 10.1103/PhysRevLett.70.1895 (cited on pages 9, 181).

[BBP+96a] Charles H. Bennett, Herbert J. Bernstein, Sandu Popescu, and Benjamin Schumacher. Concentrating partial entanglement by local operations. *Physical Review A*, 53(4):2046–2052, April 1, 1996. DOI: 10.1103/PhysRevA.53.2046 (cited on pages 9, 181).

[BBP+96b] Charles H. Bennett, Gilles Brassard, Sandu Popescu, Benjamin Schumacher, John A. Smolin, and William K. Wootters. Purification of Noisy Entanglement and Faithful Teleportation via Noisy Channels. *Physical Review Letters*, 76(5):722–725, January 29, 1996. DOI: 10.1103/PhysRevLett.76.722 (cited on pages 9, 181).

[BCC+20] Christian Badertscher, Alexandru Cojocaru, Léo Colisson, Elham Kashefi, Dominik Leichtle, Atul Mantri, and Petros Wallden. Security Limitations of Classical-Client Delegated Quantum Computing. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, Lecture Notes in Computer Science, pages 667–696, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-64834-3. DOI: 10.1007/978-3-030-64834-3_23 (cited on pages 8, 10).

[BCK+20] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On The Round Complexity of Two-Party Quantum Computation. December 17, 2020. arXiv: 2011.11212 [quant-ph] (cited on page 9).

[BCM+18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pages 320–331, October 2018. DOI: 10.1109/FOCS.2018.00038 (cited on pages 4, 103).

[BDS+96] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54(5):3824–3851, November 1, 1996. DOI: 10.1103/PhysRevA.54.3824 (cited on pages 9, 181).

[Ben80] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics*, 22(5):563–591, May 1980. ISSN: 0022-4715, 1572-9613. DOI: 10.1007/BF01011339 (cited on page 3).

[BFK09] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal Blind Quantum Computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. 2009 50th Annual IEEE Symposium on Foundations of Computer Science, pages 517–526, October 2009. DOI: 10.1109/FOCS.2009.36 (cited on pages 4, 40, 45, 46, 82, 219).

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 103–112, New York, NY, USA. Association for Computing Machinery, January 1, 1988. ISBN: 978-0-89791-264-8. DOI: 10.1145/62212.62222 (cited on page 8).

[BFS+13] Harry Buhrman, Serge Fehr, Christian Schaffner, and Florian Speelman. The garden-hose model. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 145–158, New York, NY, USA. Association for Computing Machinery, January 9, 2013. ISBN: 978-1-4503-1859-4. DOI: 10.1145/2422436.2422455 (cited on page 6).

[BG20] A. Broadbent and A. B. Grilo. QMA-hardness of Consistency of Local Density Matrices with Applications to Quantum Zero-Knowledge. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), pages 196–205, November 2020. DOI: 10.1109/FOCS46700.2020.00027 (cited on page 9).

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, Lecture Notes in Computer Science, pages 533–556, Berlin, Heidelberg. Springer, 2014. ISBN: 978-3-642-55220-5. DOI: 10.1007/978-3-642-55220-5_30 (cited on pages 7, 115).

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum Homomorphic Encryption for Circuits of Low T-gate Complexity. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, Lecture Notes in Computer Science, pages 609–629, Berlin, Heidelberg. Springer, 2015. ISBN: 978-3-662-48000-7. DOI: 10.1007/978-3-662-48000-7_30 (cited on pages 4, 6).

[BJS+16] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-Knowledge Proof Systems for QMA. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pages 31–40, October 2016. DOI: 10.1109/FOCS.2016.13 (cited on page 9).

[BK15] Joonwoo Bae and Leong-Chuan Kwek. Quantum state discrimination and its applications. *Journal of Physics A: Mathematical and Theoretical*, 48(8):083001, January 2015. ISSN: 1751-8121. DOI: 10.1088/1751-8113/48/8/083001 (cited on page 36).

[BKV+20] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. Simpler Proofs of Quantumness. May 10, 2020. arXiv: 2005.04826 [quant-ph] (cited on page 4).

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 575–584, New York, NY, USA. Association for Computing Machinery, June 1, 2013. ISBN: 978-1-4503-2029-0. DOI: 10.1145/2488608.2488680 (cited on page 100).

[BM88] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, April 1988. ISSN: 00220000. DOI: 10.1016/0022-0000(88)90028-1 (cited on pages 2, 8).

[BPW03] Michael Backes, Birgit Pfitzmann, and Michael Waidner. A composable cryptographic library with nested operations. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, pages 220–230, New York, NY, USA. Association for Computing Machinery, October 27, 2003. ISBN: 978-1-58113-738-5. DOI: 10.1145/948109.948140 (cited on pages 7, 140).

[Bra13] Zvika Brakerski. When Homomorphism Becomes a Liability. In Amit Sahai, editor, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 143–161, Berlin, Heidelberg. Springer, 2013. ISBN: 978-3-642-36594-2. DOI: 10.1007/978-3-642-36594-2_9 (cited on page 133).

[Bra18] Zvika Brakerski. Quantum FHE (Almost) As Secure As Classical. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, Lecture Notes in Computer Science, pages 67–95, Cham. Springer International Publishing, 2018. ISBN: 978-3-319-96878-0. DOI: 10.1007/978-3-319-96878-0_3 (cited on pages 4, 102, 103).

[Bro15] Anne Broadbent. Delegating private quantum computations. *Canadian Journal of Physics*, 93(9):941–946, September 1, 2015. ISSN: 0008-4204. DOI: 10.1139/cjp-2015-0030 (cited on page 4).

[BS20] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 269–279, New York, NY, USA. Association for Computing Machinery, June 22, 2020. ISBN: 978-1-4503-6979-4. DOI: 10.1145/3357713.3384324 (cited on pages 8, 183, 185).

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. *SIAM Journal on Computing*, 43(2):831–871, January 2014. ISSN: 0097-5397. DOI: 10.1137/120868669 (cited on page 6).

[Can01] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. Proceedings 42nd IEEE Symposium on Foundations of Computer Science, pages 136–145, October 2001. DOI: 10.1109/SFCS.2001.959888 (cited on pages 7, 140).

[CCK+18] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. Delegated Pseudo-Secret Random Qubit Generator. February 23, 2018. arXiv: 1802.08759v1 [cs.CR] (cited on pages 5, 10, 62).

[CCK+19] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. QFactory: Classically-Instructed Remote Secret Qubits Preparation. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019*, Lecture Notes in Computer Science, pages 615–645, Cham. Springer International Publishing, 2019. ISBN: 978-3-030-34578-5. DOI: 10.1007/978-3-030-34578-5_22 (cited on pages 5, 10, 62, 71).

[CCK+21] Alexandru Cojocaru, Léo Colisson, Elham Kashefi, and Petros Wallden. On the Possibility of Classical Client Blind Quantum Computing. *Cryptography*, 5(1):3, 1, March 2021. DOI: 10.3390/cryptography5010003 (cited on pages 10, 62).

[CD08] Bob Coecke and Ross Duncan. Interacting Quantum Observables. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 298–310, Berlin, Heidelberg. Springer, 2008. ISBN: 978-3-540-70583-3. DOI: 10.1007/978-3-540-70583-3_25 (cited on pages 14, 30).

[CDP09] Giulio Chiribella, Giacomo Mauro D'Ariano, and Paolo Perinotti. Theoretical framework for quantum networks. *Physical Review A*, 80(2):022339, August 31, 2009. DOI: 10.1103/PhysRevA.80.022339 (cited on pages 53, 54).

[Cen] Center for Cryptologic History, National Security Agency. The Venona Story (cited on page 2).

[CF01] Ran Canetti and Marc Fischlin. Universally Composable Commitments. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, Lecture Notes in Computer Science, pages 19–40, Berlin, Heidelberg. Springer, 2001. ISBN: 978-3-540-44647-7. DOI: 10.1007/3-540-44647-8_2 (cited on page 55).

[CGK21] Léo Colisson, Frédéric Grosshans, and Elham Kashefi. Non-Destructive Zero-Knowledge Proofs on Quantum States, and Multi-Party Generation of Authorized Hidden GHZ States. April 10, 2021. arXiv: 2104.04742 [quant-ph] (cited on pages 5, 10, 62).

[CGM+09] Nishanth Chandran, Vipul Goyal, Ryan Moriarty, and Rafail Ostrovsky. Position Based Cryptography. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, Lecture Notes in Computer Science, pages 391–407, Berlin, Heidelberg. Springer, 2009. ISBN: 978-3-642-03356-8. DOI: 10.1007/978-3-642-03356-8_23 (cited on pages 6, 101).

[CGS02] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 643–652, New York, NY, USA. Association for Computing Machinery, May 19, 2002. ISBN: 978-1-58113-495-7. DOI: 10.1145/509907.510000 (cited on page 6).

[CHK+10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, Lecture Notes in Computer Science, pages 523–552, Berlin, Heidelberg. Springer, 2010. ISBN: 978-3-642-13190-5. DOI: 10.1007/978-3-642-13190-5_27 (cited on page 6).

[Chr21] Schaffner Christian. Position Based Cryptography. Personal homepage of Christian Schaffner. April 12, 2021. URL: https://homepages.cwi.nl/~schaffne/positionbasedqcrypto.php (visited on 04/12/2021) (cited on page 6).

[CK17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, 2017. ISBN: 978-1-107-10422-8. DOI: 10.1017/9781316219317 (cited on pages 14, 30).

[Col17] Léo Colisson. Classically Driven Delegated Blind Quantum Computing. Internship Report, August 21, 2017 (cited on pages 5, 10, 62).

[CVZ20] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive Zero-Knowledge Arguments for QMA, with Preprocessing. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, Lecture Notes in Computer Science, pages 799–828, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-56877-1. DOI: 10.1007/978-3-030-56877-1_28 (cited on page 9).

[CW05] Matthias Christandl and Stephanie Wehner. Quantum Anonymous Transmissions. In Bimal Roy, editor, *Advances in Cryptology - ASIACRYPT 2005*, Lecture Notes in Computer Science, pages 217–235, Berlin, Heidelberg. Springer, 2005. ISBN: 978-3-540-32267-2. DOI: 10.1007/11593447_12 (cited on pages 9, 181).

[DFM+19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, Lecture Notes in Computer Science, pages 356–383, Cham. Springer International Publishing, 2019. ISBN: 978-3-030-26951-7. DOI: 10.1007/978-3-030-26951-7_13 (cited on pages 8, 185).

[DFP+14] Vedran Dunjko, Joseph F. Fitzsimons, Christopher Portmann, and Renato Renner. Composable Security of Delegated Quantum Computation. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, Lecture Notes in Computer Science, pages 406–425, Berlin, Heidelberg. Springer, 2014. ISBN: 978-3-662-45608-8. DOI: 10.1007/978-3-662-45608-8_22 (cited on pages 4, 48, 139, 140, 160, 167).

[DGJ+20] Yfke Dulek, Alex B. Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure Multi-party Quantum Computation with a Dishonest Majority. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, Lecture Notes in Computer Science, pages 729–758, Cham. Springer International Publishing, 2020. ISBN: 978-3-030-45727-3. DOI: 10.1007/978-3-030-45727-3_25 (cited on pages 6, 9, 175).

[DH76] Whitfield Diffie and Martin E. Hellman. Multiuser cryptographic techniques. In *Proceedings of the June 7-10, 1976, National Computer Conference and Exposition*, AFIPS '76, pages 109–112, New York, NY, USA. Association for Computing Machinery, June 7, 1976. ISBN: 978-1-4503-7917-5. DOI: 10.1145/1499799.1499815 (cited on page 2).

[DK06] Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Physical Review A*, 74(5):052310, November 8, 2006. DOI: 10.1103/PhysRevA.74.052310 (cited on page 40).

[DK16] Vedran Dunjko and Elham Kashefi. Blind quantum computing with two almost identical states. April 6, 2016. arXiv: 1604.01586 [quant-ph] (cited on page 140).

[DL70] E. B. Davies and J. T. Lewis. An operational approach to quantum probability. *Communications in Mathematical Physics*, 17(3):239–260, September 1, 1970. ISSN: 1432-0916. DOI: 10.1007/BF01647093 (cited on page 29).

[DN06] Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm. *Quantum Information & Computation*, 6(1):81–95, January 2006. ISSN: 1533-7146 (cited on pages 19, 30).

[DNS12] Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively Secure Two-Party Evaluation of Any Quantum Operation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, Lecture Notes in Computer Science, pages 794–811, Berlin, Heidelberg. Springer, 2012. ISBN: 978-3-642-32009-5. DOI: 10.1007/978-3-642-32009-5_46 (cited on pages 6, 9, 175).

[DSS16] Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum Homomorphic Encryption for Polynomial-Sized Circuits. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, Lecture Notes in Computer Science, pages 3–32, Berlin, Heidelberg. Springer, 2016. ISBN: 978-3-662-53015-3. DOI: 10.1007/978-3-662-53015-3_1 (cited on pages 4, 6).

[Ein05] A. Einstein. Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt. *Annalen der Physik*, 322(6):132–148, 1905. ISSN: 1521-3889. DOI: 10.1002/andp.19053220607 (cited on page 3).

[FBS+14] K. a. G. Fisher, A. Broadbent, L. K. Shalm, Z. Yan, J. Lavoie, R. Prevedel, T. Jennewein, and K. J. Resch. Quantum computing on encrypted data. *Nature Communications*, 5(1):3074, 1, January 21, 2014. ISSN: 2041-1723. DOI: 10.1038/ncomms4074 (cited on page 4).

[Fey82] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, June 1982. ISSN: 0020-7748, 1572-9575. DOI: 10.1007/BF02650179 (cited on page 3).

[Fit17] Joseph F. Fitzsimons. Private quantum computation: an introduction to blind quantum computing and related protocols. *npj Quantum Information*, 3(1):1–11, 1, June 15, 2017. ISSN: 2056-6387. DOI: 10.1038/s41534-017-0025-3 (cited on page 4).

[FK17] Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Physical Review A*, 96(1):012303, July 5, 2017. DOI: 10.1103/PhysRevA.96.012303 (cited on pages xvi, 4, 74, 101, 103, 139, 172).

[FS87] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, Lecture Notes in Computer Science, pages 186–194, Berlin, Heidelberg. Springer, 1987. ISBN: 978-3-540-47721-1. DOI: 10.1007/3-540-47721-7_12 (cited on pages 8, 9, 177).

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA. Association for Computing Machinery, May 31, 2009. ISBN: 978-1-60558-506-2. DOI: 10.1145/1536414.1536440 (cited on page 6).

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski, editor, *Advances in Cryptology — CRYPTO '97*, Lecture Notes in Computer Science, pages 112–131, Berlin, Heidelberg. Springer, 1997. ISBN: 978-3-540-69528-8. DOI: 10.1007/BFb0052231 (cited on page 117).

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, August 10, 1986. ISSN: 0004-5411. DOI: 10.1145/6490.6503 (cited on page 2).

[GHZ89] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going Beyond Bell's Theorem. In Menas Kafatos, editor, *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, Fundamental Theories of Physics, pages 69–72. Springer Netherlands, Dordrecht, 1989. ISBN: 978-94-017-0849-4. DOI: 10.1007/978-94-017-0849-4_10 (cited on page 62).

[GKK19] Alexandru Gheorghiu, Theodoros Kapourniotis, and Elham Kashefi. Verification of Quantum Computation: An Overview of Existing Approaches. *Theory of Computing Systems*, 63(4):715–808, May 1, 2019. ISSN: 1433-0490. DOI: 10.1007/s00224-018-9872-3 (cited on page 4).

[GMR85] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 291–304, New York, NY, USA. Association for Computing Machinery, December 1, 1985. ISBN: 978-0-89791-151-1. DOI: 10.1145/22145.22178 (cited on pages 2, 8).

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, NY, USA. Association for Computing Machinery, January 1, 1987. ISBN: 978-0-89791-221-1. DOI: 10.1145/28395.28420 (cited on page 2).

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, July 1991. ISSN: 0004-5411, 1557-735X. DOI: 10.1145/116825.116852 (cited on pages 2, 8).

[GNW98] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR-Lemma. 6650, November 21, 1998. ISSN: 978-3-642-22669-4. DOI: 10.1007/978-3-642-22670-0_23 (cited on pages 89, 90).

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1, 1994. ISSN: 1432-1378. DOI: 10.1007/BF00195207 (cited on page 8).

[Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1: Basic Tools*, volume 1. Cambridge University Press, Cambridge, 2001. ISBN: 978-0-521-03536-1. DOI: 10.1017/CBO9780511546891 (cited on page 55).

[Gol04] Oded Goldreich. *Foundations of Cryptography: Volume 2: Basic Applications*, volume 2. Cambridge University Press, Cambridge, 2004. ISBN: 978-0-521-11991-7. DOI: 10.1017/CBO9780511721656 (cited on page 56).

[Goo] Google. Google Quantum Computing Service | Cirq. Google Quantum AI. URL: https://quantumai.google/cirq/google/concepts (visited on 01/24/2022) (cited on page 3).

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA. Association for Computing Machinery, May 17, 2008. ISBN: 978-1-60558-047-0. DOI: 10.1145/1374376.1374407 (cited on pages 6, 113).

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA. Association for Computing Machinery, July 1, 1996. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866 (cited on page 3).

[GRW80] G. C. Ghirardi, A. Rimini, and T. Weber. A general argument against superluminal transmission through the quantum mechanical measurement process. *Lettere al Nuovo Cimento (1971-1985)*, 27(10):293–298, March 1, 1980. ISSN: 1827-613X. DOI: 10.1007/BF02817189 (cited on page 166).

[GSY19] Alex B. Grilo, William Slofstra, and Henry Yuen. Perfect Zero Knowledge for Quantum Multiprover Interactive Proofs. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), pages 611–635, November 2019. DOI: 10.1109/FOCS.2019.00044 (cited on page 9).

[GV19] A. Gheorghiu and T. Vidick. Computationally-Secure and Composable Remote State Preparation. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), pages 1024–1033, November 2019. DOI: 10.1109/FOCS.2019.00066 (cited on pages 4, 6, 7, 62, 100–104, 139, 156, 157, 171).

[GW07] Gus Gutoski and John Watrous. Toward a general theory of quantum games. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 565–574, New York, NY, USA. Association for Computing Machinery, June 11, 2007. ISBN: 978-1-59593-631-8. DOI: 10.1145/1250790.1250873 (cited on page 53).

[Hal17] Shai Halevi. Homomorphic Encryption. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, Information Security and Cryptography, pages 219–276. Springer International Publishing, Cham, 2017. ISBN: 978-3-319-57048-8. DOI: 10.1007/978-3-319-57048-8_5 (cited on page 133).

[HBB99] Mark Hillery, Vladimír Bužek, and André Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829–1834, March 1, 1999. DOI: 10.1103/PhysRevA.59.1829 (cited on pages 9, 181, 182).

[Hel69] Carl W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, June 1, 1969. ISSN: 1572-9613. DOI: 10.1007/BF01007479 (cited on page 36).

[HG21] Shuichi Hirahara and François Le Gall. Test of Quantumness with Small-Depth Quantum Circuits. August 9, 2021. DOI: 10.4230/LIPIcs.MFCS.2021.59. arXiv: 2105.05500 [quant-ph] (cited on page 4).

[HIL+88] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-Random Generation from One-Way Functions. In *Proc. 20th Stoc*, pages 12–24, 1988 (cited on page 2).

[Hol73] A. S Holevo. Statistical decision theory for quantum systems. *Journal of Multivariate Analysis*, 3(4):337–394, December 1, 1973. ISSN: 0047-259X. DOI: 10.1016/0047-259X(73)90028-6 (cited on page 36).

[Hon] Honeywell. Quantum computer. URL: https://www.honeywell.com/us/en/company/quantum/quantum-computer (visited on 09/13/2021) (cited on page 3).

[IBM] IBM. IBM Quantum. IBM Quantum. URL: https://quantum-computing.ibm.com/ (visited on 09/13/2021) (cited on page 3).

[Imp95a] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*. Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference, pages 134–147, June 1995. DOI: 10.1109/SCT.1995.514853 (cited on page 1).

[Imp95b] R. Impagliazzo. Hard-core distributions for somewhat hard problems. *Proceedings of IEEE 36th Annual Foundations of Computer Science*, 1995. DOI: 10.1109/SFCS.1995.492584 (cited on page 90).

[Ion] IonQ. Get started with trapped ion quantum computing. IonQ. URL: https://ionq.com/get-started/ (visited on 09/13/2021) (cited on page 3).

[IY88] Russell Impagliazzo and Moti Yung. Direct Minimum-Knowledge Computations (Extended Abstract). In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, Lecture Notes in Computer Science, pages 40–51, Berlin, Heidelberg. Springer, 1988. ISBN: 978-3-540-48184-3. DOI: 10.1007/3-540-48184-2_4 (cited on page 9).

[JM17] D. Jost and U. Maurer. Context-Restricted Indifferentiability: Generalizing UCE and Implications on the Soundness of Hash-Function Constructions. *IACR Cryptol. ePrint Arch.*, 2017 (cited on pages 158, 159).

[JM98] J. A. Jones and M. Mosca. Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer. *The Journal of Chemical Physics*, 109(5):1648–1653, August 1, 1998. ISSN: 0021-9606. DOI: 10.1063/1.476739 (cited on page 3).

[JPV19] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A Generic Normal Form for ZX-Diagrams and Application to the Rational Angle Completeness. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), pages 1–10, June 2019. DOI: 10.1109/LICS.2019.8785754 (cited on page 99).

[Kit97] A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. ISSN: 0036-0279, 1468-4829. DOI: 10.1070/RM1997v052n06ABEH002155 (cited on pages 19, 30).

[KKM+21] Theodoros Kapourniotis, Elham Kashefi, Luka Music, and Harold Ollivier. Delegating Multi-Party Quantum Computations vs. Dishonest Majority in Two Quantum Rounds. September 9, 2021. arXiv: 2102.12949 [quant-ph] (cited on pages 6, 9, 101).

[KMS11] Adrian Kent, William J. Munro, and Timothy P. Spiller. Quantum tagging: Authenticating location via quantum information and relativistic signaling constraints. *Physical Review A*, 84(1):012326, July 21, 2011. DOI: 10.1103/PhysRevA.84.012326 (cited on page 6).

[KMW17] Elham Kashefi, Luka Music, and Petros Wallden. The Quantum Cut-and-Choose Technique and Quantum Two-Party Computation. March 10, 2017. arXiv: 1703.03754 [quant-ph] (cited on page 6).

[KP17] Elham Kashefi and Anna Pappa. Multiparty Delegated Quantum Computing. *Cryptography*, 1(2):12, 2, September 2017. DOI: 10.3390/cryptography1020012 (cited on page 6).

[KRK13] Alastair Kay, Ravishankar Ramanathan, and Dagomir Kaszlikowshi. Optimal asymmetric quantum cloning for quantum information and computation. *Quantum Information & Computation*, 13(9-10):880–900, September 2013. ISSN: 1533-7146 (cited on page 156).

[KŠdW07] Hartmut Klauck, Robert Špalek, and Ronald de Wolf. Quantum and Classical Strong Direct Product Theorems and Optimal Time-Space Tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, January 1, 2007. ISSN: 0097-5397. DOI: 10.1137/05063235X (cited on page 90).

[KW15] Elham Kashefi and Petros Wallden. Optimised resource construction for verifiable quantum computation. *Journal of Physics A: Mathematical and Theoretical*, 50, October 26, 2015. DOI: 10.1088/1751-8121/aa5dac (cited on page 74).

[KW17] Elham Kashefi and Petros Wallden. Garbled Quantum Computation. *Cryptography*, 1(1):6, 1, June 2017. DOI: 10.3390/cryptography1010006 (cited on page 6).

[Lev87] Leonid A. Levin. One way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, December 1, 1987. ISSN: 1439-6912. DOI: 10.1007/BF02579323 (cited on page 90).

[Lia15] Min Liang. Quantum fully homomorphic encryption scheme based on universal quantum circuit. *Quantum Information Processing*, 14(8):2749–2759, August 1, 2015. ISSN: 1573-1332. DOI: 10.1007/s11128-015-1034-9 (cited on page 4).

[Lin17] Yehuda Lindell. How to Simulate It – A Tutorial on the Simulation Proof Technique. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, Information Security and Cryptography, pages 277–346. Springer International Publishing, Cham, 2017. ISBN: 978-3-319-57048-8. DOI: 10.1007/978-3-319-57048-8_6 (cited on page 55).

[LLQ21] Jiahui Liu, Qipeng Liu, and Luowen Qian. Beating Classical Impossibility of Position Verification. September 15, 2021. arXiv: 2109.07517 [quant-ph] (cited on pages 6, 220).

[LM08] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically Efficient Lattice-Based Digital Signatures. In Ran Canetti, editor, *Theory of Cryptography*, Lecture Notes in Computer Science, pages 37–54, Berlin, Heidelberg. Springer, 2008. ISBN: 978-3-540-78524-8. DOI: 10.1007/978-3-540-78524-8_3 (cited on page 6).

[LR85] Michael Luby and Charles Rackoff. How to Construct Pseudo-Random Permutations from Pseudo-Random Functions (Abstract). In SIAM Journal on Computing, volume 17, page 447, August 18, 1985. ISBN: 978-3-540-16463-0. DOI: 10.1007/3-540-39799-X_34 (cited on page 2).

[LRW20] Victoria Lipinska, Jérémy Ribeiro, and Stephanie Wehner. Secure multiparty quantum computation with few qubits. *Physical Review A*, 102(2):022405, August 7, 2020. DOI: 10.1103/PhysRevA.102.022405 (cited on page 6).

[LSB+19] Andreas Lochbihler, S. Reza Sefidgar, David Basin, and Ueli Maurer. Formalizing Constructive Cryptography using CryptHOL. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*. 2019 IEEE 32nd Computer Security Foundations Symposium (CSF), pages 152–15214, June 2019. DOI: 10.1109/CSF.2019.00018 (cited on page 147).

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting Post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, Lecture Notes in Computer Science, pages 326–355, Cham. Springer International Publishing, 2019. ISBN: 978-3-030-26951-7. DOI: 10.1007/978-3-030-26951-7_12 (cited on pages 8, 185).

[Mah18a] Urmila Mahadev. Classical Homomorphic Encryption for Quantum Circuits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pages 332–338, Paris. IEEE, October 2018. ISBN: 978-1-5386-4230-6. DOI: 10.1109/FOCS.2018.00039 (cited on pages 4, 5, 62, 102, 103, 178, 213, 217).

[Mah18b] Urmila Mahadev. Classical Verification of Quantum Computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pages 259–267, October 2018. DOI: 10.1109/FOCS.2018.00033 (cited on pages 4, 102–104).

[Man80] IUrii Ivanovich Manin. *Vychislimoe i nevychislimoe*. "Sov. radio, ", Moskva, 1980 (cited on page 3).

[Mau12] Ueli Maurer. Constructive Cryptography – A New Paradigm for Security Definitions and Proofs. In Sebastian Mödersheim and Catuscia Palamidessi, editors, *Theory of Security and Applications*, Lecture Notes in Computer Science, pages 33–56, Berlin, Heidelberg. Springer, 2012. ISBN: 978-3-642-27375-9. DOI: 10.1007/978-3-642-27375-9_3 (cited on pages 140, 143, 145–147).

[MDC+21] Tony Metger, Yfke Dulek, Andrea Wei Coladangelo, and Rotem Arnon-Friedman. Device-independent quantum key distribution from computational assumptions. *New Journal of Physics*, 2021. ISSN: 1367-2630. DOI: 10.1088/1367-2630/ac304b (cited on page 4).

[MDM+17] Atul Mantri, Tommaso F. Demarie, Nicolas C. Menicucci, and Joseph F. Fitzsimons. Flow Ambiguity: A Path Towards Classically Driven Blind Quantum Computation. *Physical Review X*, 7(3):031004, July 11, 2017. DOI: 10.1103/PhysRevX.7.031004 (cited on page 5).

[Mer89] N. David Mermin. What's Wrong with this Pillow? *Physics Today*, 42(4):9–11, April 1989. ISSN: 0031-9228. DOI: 10.1063/1.2810963 (cited on page 13).

[MF18] Tomoyuki Morimae and Joseph F. Fitzsimons. Post hoc verification with a single prover. *Physical Review Letters*, 120(4):040501, January 22, 2018. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.040501. arXiv: 1603.06046 (cited on page 103).

[MK19] Tomoyuki Morimae and Takeshi Koshiba. Impossibility of perfectly-secure one-round delegated quantum computing for classical client. March 24, 2019. arXiv: 1407.1636 [quant-ph] (cited on page 7).

[MMG19] Clément Meignant, Damian Markham, and Frédéric Grosshans. Distributing graph states over arbitrary quantum networks. *Physical Review A*, 100(5):052333, November 27, 2019. DOI: 10.1103/PhysRevA.100.052333 (cited on pages 10, 181).

[MMK+95] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. Demonstration of a Fundamental Quantum Logic Gate. *Physical Review Letters*, 75(25):4714–4717, December 18, 1995. DOI: 10.1103/PhysRevLett.75.4714 (cited on page 3).

[MMP+18] Christian Matt, Ueli Maurer, Christopher Portmann, Renato Renner, and Björn Tackmann. Toward an algebraic theory of systems. *Theoretical Computer Science*, 747:1–25, November 7, 2018. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2018.06.001 (cited on page 147).

[Mon16] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):1–8, 1, January 12, 2016. ISSN: 2056-6387. DOI: 10.1038/npjqi.2015.23 (cited on page 3).

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: simpler, Tighter, Faster, Smaller. In David Pointcheval and Thomas Johansson, editors. Redacted by David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, and Gerhard Weikum, *Advances in Cryptology – EUROCRYPT 2012*. Volume 7237, Lecture Notes in Computer Science, pages 700–718. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN: 978-3-642-29010-7 978-3-642-29011-4. DOI: 10.1007/978-3-642-29011-4_41 (cited on pages 108, 109, 111, 112, 114, 116–121).

[MR09] Daniele Micciancio and Oded Regev. Lattice-based Cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer, Berlin, Heidelberg, 2009. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_5 (cited on page 117).

[MR11] U. Maurer and R. Renner. Abstract Cryptography. In *ICS*, 2011 (cited on pages 7, 8, 140, 146, 147, 154).

[MR82] Jagdish Mehra and Helmut Rechenberg. *The Historical Development of Quantum Theory*. Springer-Verlag, New York, 1982. 6 pages. ISBN: 978-0-387-90642-3 (cited on page 3).

[MV21] Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *Quantum*, 5:544, September 16, 2021. DOI: 10.22331/q-2021-09-16-544 (cited on page 4).

[MY21] Tomoyuki Morimae and Takashi Yamakawa. Classically Verifiable (Dual-Mode) NIZK for QMA with Preprocessing. February 17, 2021. arXiv: 2102.09149 [quant-ph] (cited on page 9).

[MY98] D. Mayers and A. Yao. Quantum cryptography with imperfect apparatus. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280), pages 503–509, November 1998. DOI: 10.1109/SFCS.1998.743501 (cited on pages 9, 181).

[NC10] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition. en. https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE, December 2010. DOI: 10.1017/CBO9780511976667 (cited on pages 14, 36, 38).

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, pages 33–43, New York, NY, USA. Association for Computing Machinery, February 1, 1989. ISBN: 978-0-89791-307-2. DOI: 10.1145/73007.73011 (cited on page 2).

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 333–342, New York, NY, USA. Association for Computing Machinery, May 31, 2009. ISBN: 978-1-60558-506-2. DOI: 10.1145/1536414.1536461 (cited on page 6).

[Pei10] Chris Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In Tal Rabin, editor. Redacted by David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, and Gerhard Weikum, *Advances in Cryptology – CRYPTO 2010*. Volume 6223, Lecture Notes in Computer Science, pages 80–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN: 978-3-642-14622-0 978-3-642-14623-7. DOI: 10.1007/978-3-642-14623-7_5 (cited on page 115).

[Pei16] Chris Peikert. A Decade of Lattice Cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, March 1, 2016. ISSN: 1551-305X. DOI: 10.1561/0400000074 (cited on pages 6, 114, 115, 117).

[Pla01] Max Planck. Ueber das Gesetz der Energieverteilung im Normalspectrum. *Annalen der Physik*, 309(3):553–563, 1901. ISSN: 1521-3889. DOI: 10.1002/andp.19013090310 (cited on page 3).

[PMM+17] Christopher Portmann, Christian Matt, Ueli Maurer, Renato Renner, and Björn Tackmann. Causal Boxes: quantum Information-Processing Systems Closed Under Composition. *IEEE Transactions on Information Theory*, 63(5):3277–3305, May 2017. ISSN: 1557-9654. DOI: 10.1109/TIT.2017.2676805 (cited on pages 54, 147).

[PR21] Christopher Portmann and Renato Renner. Security in Quantum Cryptography. August 30, 2021. arXiv: 2102.00021 [quant-ph] (cited on page 147).

[PRS17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 461–473, New York, NY, USA. Association for Computing Machinery, June 19, 2017. ISBN: 978-1-4503-4528-6. DOI: 10.1145/3055399.3055489 (cited on pages 114, 128).

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 187–196, New York, NY, USA. Association for Computing Machinery, May 17, 2008. ISBN: 978-1-60558-047-0. DOI: 10.1145/1374376.1374406 (cited on page 119).

[PWD18] A. Pirker, J. Wallnöfer, and W. Dür. Modular architectures for quantum networks. *New Journal of Physics*, 20(5):053054, May 2018. ISSN: 1367-2630. DOI: 10.1088/1367-2630/aac2aa (cited on pages 9, 181).

[RB01] Robert Raussendorf and Hans J. Briegel. A One-Way Quantum Computer. *Physical Review Letters*, 86(22):5188–5191, May 28, 2001. DOI: 10.1103/PhysRevLett.86.5188 (cited on page 40).

[RB02] Robert Raussendorf and Hans J. Briegel. Computational model underlying the one-way quantum computer. *Quantum Information & Computation*, 2(6):443–486, October 1, 2002. ISSN: 1533-7146 (cited on page 40).

[RBB03] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, August 25, 2003. DOI: 10.1103/PhysRevA.68.022312 (cited on page 40).

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA. Association for Computing Machinery, May 22, 2005. ISBN: 978-1-58113-960-0. DOI: 10.1145/1060590.1060603 (cited on pages 2, 6, 112–114).

[Rig] Rigetti. Rigetti Computing. URL: https://www.rigetti.com/get-quantum (visited on 09/13/2021) (cited on page 3).

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1, 1978. ISSN: 0001-0782. DOI: 10.1145/359340.359342 (cited on pages 2, 3).

[Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, January 1, 1987. ISSN: 0304-3975. DOI: 10.1016/0304-3975(87)90064-8 (cited on pages 7, 115).

[SE02] Michael Smith and Ralph Erskine. *Action This Day: Bletchley Park : From the Breaking of Enigma to the Birth of the Modern Computer*. Bantam, London, 2002. ISBN: 978-0-593-04982-2 (cited on page 2).

[She12] Alexander A. Sherstov. Strong Direct Product Theorems for Quantum Communication and Query Complexity. *SIAM Journal on Computing*, 41(5):1122–1165, January 1, 2012. ISSN: 0097-5397. DOI: 10.1137/110842661 (cited on page 90).

[Shm20] Omri Shmueli. Multi-theorem (Malicious) Designated-Verifier NIZK for QMA. July 25, 2020. arXiv: 2007.12923 [quant-ph] (cited on page 9).

[Sho04] Victor Shoup. Sequences of Games: a Tool for Taming Complexity in Security Proofs. *IACR Cryptology ePrint Archive*, 2004:332, January 1, 2004 (cited on page 55).

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 35th Annual Symposium on Foundations of Computer Science, pages 124–134, Santa Fe, NM, USA. IEEE Comput. Soc. Press, 1994. ISBN: 978-0-8186-6580-6. DOI: 10.1109/SFCS.1994.365700 (cited on page 3).

[Unr10] Dominique Unruh. Universally Composable Quantum Multi-party Computation. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, Lecture Notes in Computer Science, pages 486–505, Berlin, Heidelberg. Springer, 2010. ISBN: 978-3-642-13190-5. DOI: 10.1007/978-3-642-13190-5_25 (cited on pages 7, 140, 170).

[Unr12] Dominique Unruh. Quantum Proofs of Knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, Lecture Notes in Computer Science, pages 135–152, Berlin, Heidelberg. Springer, 2012. ISBN: 978-3-642-29011-4. DOI: 10.1007/978-3-642-29011-4_10 (cited on pages 8, 185).

[Unr14] Dominique Unruh. Quantum Position Verification in the Random Oracle Model. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, Lecture Notes in Computer Science, pages 1–18, Berlin, Heidelberg. Springer, 2014. ISBN: 978-3-662-44381-1. DOI: 10.1007/978-3-662-44381-1_1 (cited on page 6).

[vdWet20] John van de Wetering. ZX-calculus for the working quantum computer scientist. December 27, 2020. arXiv: 2012.13966 [quant-ph] (cited on pages 14, 30).

[VW07] Emanuele Viola and Avi Wigderson. Norms, XOR Lemmas, and Lower Bounds for GF(2) Polynomials and Multiparty Protocols. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*. Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07), pages 141–154, June 2007. DOI: 10.1109/CCC.2007.15 (cited on pages 89, 90).

[VW16] Thomas Vidick and John Watrous. Quantum Proofs. *Foundations and Trends® in Theoretical Computer Science*, 11(1-2):1–215, 2016. ISSN: 1551-305X, 1551-3068. DOI: 10.1561/0400000068 (cited on page 9).

[VZ20] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations. *Quantum*, 4:266, May 14, 2020. ISSN: 2521-327X. DOI: 10.22331/q-2020-05-14-266. arXiv: 1902.05217 (cited on page 9).

[Wat09] John Watrous. Zero-Knowledge against Quantum Attacks. *SIAM Journal on Computing*, 39(1):25–58, January 1, 2009. ISSN: 0097-5397. DOI: 10.1137/060670997 (cited on pages 8, 185).

[Xan] Xanadu. Xanadu Quantum Technologies. Xanadu. URL: http://www.xanadu.ai/ (visited on 01/24/2022) (cited on page 3).

[Zha19] Mark Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, Lecture Notes in Computer Science, pages 239–268, Cham. Springer International Publishing, 2019. ISBN: 978-3-030-26951-7. DOI: 10.1007/978-3-030-26951-7_9 (cited on page 101).

[Zha21] Jiayu Zhang. Succinct blind Quantum computation using a random oracle. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1370–1383. Association for Computing Machinery, New York, NY, USA, June 15, 2021. ISBN: 978-1-4503-8053-9 (cited on page 5).

[ZWD+20] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, December 18, 2020. DOI: 10.1126/science.abe8770 (cited on page 3).