UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Quantum assisted Secure Multiparty Computation

Manuel Batalha dos Santos

Supervisor: Doctor Paulo Alexandre Carreira Mateus
Co-Supervisor: Doctor Armando Nolasco Pinto

Thesis specifically prepared to obtain the PhD Degree in Mathematics

Draft

January 2023

Abstract

Start with no indent.

Then you can write another paragraph.

Key-words: quantum cryptography, quantum oblivious transfer, quantum obliious linear evaluation, secure multiparty computation.

Resumo

Escrever a mesma coisa que está no Abstract, mas em Português.

Palavras-chave: criptografia quântica, passeios quânticos, memórias quânticas, transições de fase topológicas, estados de fronteira

Acknowledgments

Write the acknowledgments here.

I acknowledge Fundação para a Ciência e a Tecnologia (FCT, Portugal) for its support through the PhD grant SFRH/BD/ 144806/2019 in the context of the Doctoral Program in the Information Security (IS). I also acknowledge support from SQIG (Security and Quantum Information Group) in the Instituto de Telecomunicações (IT), Lisbon, namely through UID/EEA/50008/2013. (CHECK THIS)

I dedicate this thesis to my loving wife Teresinha and my two children Henrique and Helena who came to life during this journey to help me finish it.

Contents

List of Figures

List of Tables

List of Abbreviations

A – Alice

B – Bob

BCS – Bardeen-Cooper-Schrieffer

BG – Boltzmann-Gibbs

CS – Chiral symmetry

DTQW – Discrete-time quantum walk

DQPT – Dynamical quantum phase transition

E – Eve

EB – Entanglement based

LE – Loschmidt Echo

MDM – Massive Dirac model

PHS – Particle-hole symmetry

PT – Phase transition

PM – Prepare and measure

SSH – Su-Schrieffer-Heeger

TI – Topological insulator

TSC – Topological superconductor

TRS – Time-reversal symmetry

QKD – Quantum Key Distribution

QW – Quantum walk

Chapter 1

Introduction

The emerging fields of Data Mining and Data Analysis have deeply benefited from the increasing power of computers [3]. However, its need for a massive and methodical collection of data can lead to the complete or partial leak of private sensitive information, such as in the case of the genomics field [4–7]. As a consequence, the aggregation of data from different sources is most of the times blocked due to legally imposed regulations such as the General Data Protection Regulation (GDPR) [8]. Although this has the benefit of protecting people's privacy, it also has the downside of preventing honest players from accessing data necessary to tackle some of the most important issues in our society.

Secure Multiparty Computation

To overcome the privacy-related issues described above, several privacy-enhancing technologies have been proposed [9–11]. One important area of research is Secure Multiparty Computation (SMC). This technology allows a set of n parties P_i to jointly compute some function $f(x_1, ..., x_n) = (y_1, ..., y_n)$ without disclosing their inputs to the other parties. The security requirements of SMC are equivalent to an ideal case, where every party P_i sends his inputs to some independent and trusted third party, who computes $f()$ and sends back to each party their corresponding output.

Since Yao seminal work [12], several SMC protocols have been developed, rendering different framework implementations [13–15]. However, they can generally be separated into two types according to the circuit logic being used: boolean or arithmetic. In each case, the efficiency and security of SMC heavily rely on the efficiency and security of important cryptographic primitives. Boolean-based SMC protocols rely on Oblivious Transfer (OT) [16] and arithmetic-based rely on Oblivious Linear Evaluation (OLE) [17]. Impagliazzo and Rudich [18] proved that both OT and OLE protocols require public cryptography and cannot just rely on symmetric cryptography. This is an unfortunate result both from an efficiency and security perspective. Indeed, symmetric cryptography is lighter than asymmetric cryptography and requires less computational assumptions. Moreover, with the emergence of quantum computers, Shor's algorithm [19] jeopardizes all the current public-key methods based on RSA, Elliptic Curves or Diffie-Hellman, in which many OT and OLE implementations rely on. This puts at risk the deployment of classical OT and OLE, which ultimately leads to the exposure of the SMC parties' private inputs. Thus, it is essential to develop SMC methods secure against quantum computers while not compromising state-of-the-art performance levels.

A Quantum Era

We are now in the beginning of what is known to be the second quantum revolution. Quantum technology has evolved to a point where we can integrate quantum exotic features into complex engineering systems. Most of the applications lie in the field of quantum cryptography, where one thrives to find protocols that offer some advantage over their classical counterparts. As analysed in [20, 21], these advantages can be of two types:

1. Improve the security requirements, rendering protocols that are information-theoretically secure or require fewer computational assumptions;

2. Achieve new primitives that were previously not possible just with classical techniques.

Despite the most famous use-case of quantum cryptography being quantum key distribution (QKD), other primitives play an important role in this quest. Some examples of these cryptographic tasks are bit commitment [22], coin flipping [23], delegated quantum computation [24], position verification [25], and password-based identification [26, 27].

Also, the intrinsic randomness provided by quantum phenomena is an ideal resource to develop quantum communication protocols for oblivious transfer (OT) [28]. Remarkably, there is a distinctive difference between classical and quantum OT from a security standpoint, as the latter is proved to be possible assuming only the existence of quantum-hard one-way functions [29, 30]. This means quantum OT can be based only on symmetric cryptography, requiring weaker security assumptions than classical OT. Moreover, these quantum protocols frequently have a desirable property that guarantees information-theoretic security after the execution of the protocol. This property is commonly called everlasting security. This greatly improves the security of SMC protocols, allowing them to have their security based on symmetric cryptography alone and with this important feature of everlasting security. Regarding oblivious linear evaluation (OLE) primitive, it is known that it can be reduced to OT [31] through classical methods that do not require further assumptions. Therefore, it seems natural to use quantum OT to generate quantum-secure OLE instances.

Contributions and Outline

Despite the many advances, the adoption of quantum cryptography by secure multiparty computation (SMC) systems is still reduced. This is due to the efficiency challenges imposed by quantum technology and the need of high throughput of both OT and OLE primitives in boolean- and arithmetic-based SMC, respectively.

The overall goal of this dissertation is to give one step closer to the adoption of quantum cryptography by SMC systems. We do this with three contributions. In our first contribution, we start the studying of comparing the efficiency of both classical and quantum protocols. Our second contribution is the first quantum OLE protocol which does not rely on OT. Our last contribution is an implementation of a special-purpose SMC system applied to genomics analysis assisted with quantum OT. Along the way, we produced a review dedicated to quantum OT protocols alone. Usually, its analysis is integrated into more general surveys under the topic of "quantum cryptography", leading to a less in-depth exposition of the topic.

We describe the contributions in a bit more detail.

Efficiency of classical and quantum OT protocols. To the best of our knowledge, there is no comparative study on the efficiency of quantum and classical approaches. This is mainly caused by two reasons. From a theoretical perspective, the use of different types of information (quantum and classical) makes it difficult to make a fair comparison based on the protocols' complexity. Also, from a practical standpoint, there is still a discrepancy in the technological maturity between quantum and classical techniques. Quantum technology is still in its infancy, whereas classical processors and communication have many decades of development.

Despite these constraints, we compare the complexity and operations efficiency of classical and quantum protocols. To achieve this, we realize that both classical and quantum protocols can be divided into two phases: offline and online. The offline phase is characterize by the fact that it is independent of the parties' inputs. This means that, from a practical point-of-view, this phase produces the resources necessary to use during the online phase, where we take into consideration the parties' inputs. It can be argued that the offline phase is not so hungry-efficient as the online phase. As a consequence, for comparison purposes, we can focus on the online phase. Fortunately, the online phase of quantum OT is solely based on classical communications. Therefore, it is possible and fair to compare the online phase of both classical and quantum protocols.

We make a detailed comparison between the complexity of the online phase of two state-of-the-art classical OT protocols [2, 32] and an optimized quantum OT protocol. We conclude that the online phase of quantum OT competes with its classical counterparts and has the potential to be more efficient.

Convetion issue: use precomputation phase instead of offline; use transfer phase instead of online.

Quantum oblivious linear evaluation protocol. Our second contribution is a quantum protocol for OLE with quantum universally composable (quantum-UC) security in the $\mathcal{F}_{\mathbf{COM}}-$hybrid model, i.e. when assuming the existence of a commitment functionality, $\mathcal{F}_{\mathbf{COM}}$. To obtain a secure protocol, we take advantage of the properties of Mutually Unbiased Bases in high-dimensional Hilbert spaces with prime and prime-power dimension. Such a choice is motivated by recent theoretical and experimental advances that pave the way for the development and realization of new solutions for quantum cryptography [33–37].

To the best of our knowledge our protocol is the first quantum-UC secure quantum OLE proposal. Moreover, it is not based on any quantum OT implementation which would be the standard approach. We consider the static corruption adversarial model with both semi-honest and malicious adversaries. We develop a weaker version of OLE, which may be of independent interest. We also modify the proposed protocol to generate quantum-UC secure vector OLE (VOLE). We give bounds on the possible size of VOLE according to the security parameters.

Quantum assisted secure multiparty computation. Individuals' privacy and legal regulations demand genomic data be handled and studied with highly secure privacy-preserving techniques. In this contribution, we propose a feasible secure multiparty computation (SMC) system assisted with quantum cryptographic protocols that is designed to compute a phylogenetic tree from a set of private genome sequences. This system adapts several distance-based methods (Unweighted Pair Group Method with Arithmetic mean, Neighbour-Joining, Fitch-Margoliash) into a private setting where the sequences owned by each party are not disclosed to the other members present in the protocol. We do not apply a generic implementation of SMC to the problem of phylogenetic trees. Instead, we develop a tailored private protocol for this use case in order to improve efficiency.

We theoretically evaluate the performance and privacy guarantees of the system through a complexity analysis and security proof and give an extensive explanation about the implementation details and cryptographic protocols. We also implement a quantum-assisted secure phylogenetic tree computation based on the Libscapi implementation of the Yao protocol, the PHYLIP library and simulated keys of two quantum systems: quantum oblivious key distribution and quantum key distribution.[1]. This demonstrates its effectiveness and practicality. We benchmark our implementation against a classical-only solution and we conclude that both approaches render similar execution times. The only difference between the quantum and classical systems is the time overhead taken by the oblivious key management system of the quantum-assisted approach.

The results are presented as follows. We start presenting SMC protocols based on OT and OLE at Chapter 2. Then, at Chapter 3 we introduce some quantum information concepts and security definitions used throughout the thesis. Chapter 4 is devoted to quantum oblivious transfer protocols. Then, in Chapter 5 we compare classical and quantum approaches for OT. In Chapter 6 we present out quantum OLE protocol along with its security proof. Finally, in Chapter 7, we presented our implementation of quantum-assisted SMC system applied to phylogeny analysis.

Published research

This thesis is based on research published in various journals. During my PhD I was involved in the following projects.

- [38] Manuel B. Santos, Paulo Mateus, and Armando N. Pinto. "Quantum Oblivious Transfer: A Short Review". In: Entropy 24.7 (July 2022), p. 945.

- [39] Manuel B. Santos, Armando N. Pinto, and Paulo Mateus. "Quantum and classical oblivious transfer: A comparative analysis". In: IET Quantum Communication 2.2 (May 2021), pp. 42–53.

- [40] Manuel B. Santos, Paulo Mateus, and Chrysoula Vlachou. Quantum Universally Composable Oblivious Linear Evaluation. 2022. DOI: 10.48550/ARXIV.2204.14171. Poster at QCrypt2022.

- [41] Manuel B. Santos et al. "Private Computation of Phylogenetic Trees Based on Quantum Technologies". In: IEEE Access 10 (2022), pp. 38065–38088.

- [42] Manuel B. Santos et al. "Quantum Secure Multiparty Computation of Phylogenetic Trees of SARS-CoV-2 Genome". In: 2021 Telecoms Conference (ConfTELE). IEEE, Feb. 2021.

[1] The code can be accessed at the following repo: https://github.com/manel1874/QSHY/tree/dev-cq-phylip

- [43] Armando N. Pinto et al. "Quantum Enabled Private Recognition of Composite Signals in Genome and Proteins". In: 2020 22nd International Conference on Transparent Optical Networks (ICTON). IEEE, July 2020.

Chapter 4 is based on [38]. Chapter 5 is based on the work developed on both [39] and [41]. Chapter 6 presents all the results from [40]. Finally, Chapter 7 is the combination of [41–43]

Chapter 2

Technical Overview

2.1 Mathematical preliminaries

Recall, we use the notation $s \leftarrow_{\$} S$ to describe a situation where an element s is drawn uniformly at random from the set S.

Throughout this thesis, Alice plays the role of the sender and Bob plays the role of the receiver.

Introduce $\mathcal{O}$ notation. It is used in chapter 4.

2.2 Secure Multiparty Computation

Estrutura da introdução:

- Cometar que não sabemos mais do que o output da computação. Dar o exemplo da média de pesos. 2 pessoas sabemos o resultado. 3 já não. Ainda assim, pode revelar alguma coisa a mais. Note that, pratically, we can put together other PET such as Differential Privacy in oder to do this.

Talk about two approaches: boolean and arithmetic. Discuss the advantages and disadvantages of each.

2.2.1 Boolean approach

Boolean approach is based on the Yao protocol. In order to do it we need OT. We start by presenting OT and then we describe the Yao protocol.

Oblivious Transfer

The study of oblivious transfer (OT) has been very active since its first proposal in 1981 by Rabin [44]. The importance of OT comes from its wide number of applications. More specifically, one can prove that OT is equivalent to the secure two-party computation of general functions [12, 16], i.e. one can implement a secure two-party computation using OT as its building block. Additionally, this primitive can also be used for secure multi-party computation (SMC) [31], private information retrieval [45], private set intersection [46], and privacy-preserving location-based services [47].

Definition:

use the concealing property and the obliviousness property (used in chapter 4)

Small classical review

Base OT vs Extended OT

Yao protocol

Description

Optimizations

Security

Generalizations of Yao: GMW, BMR

2.2.2 Arithmetic approach

Oblivious Linear Evaluation

SPDZ

2.3 Quantum Information

$\mathcal{B}(\mathcal{H})$ is the set of positive semi-definite operators with unitary trace acting on an Hilbert space $\mathcal{H}$. It is used in chapter 3.2.5 Noisy-quantum-storage model

2.3.1 Quantum states representation

2.3.2 Entropy

2.3.3 Two-universal functions

2.3.4 Mutually Unbiased Basis

2.4 Universal Composability

2.5 Functionality definitions

Chapter 3

Quantum Oblivious Transfer

In a recent survey on classical oblivious transfer (OT) [48], all the analysed protocols require some form of asymmetric cryptography. Indeed, in the classical setting, it is impossible to develop information-theoretic secure OT or even reduce it to one-way functions, requiring some public-key computational assumptions. As shown by Impaggliazzo and Rudich [49], one-way functions (symmetric cryptography) alone do not imply key agreement (asymmetric cryptography). Also, Gertner et al. [50] pointed out that since it is known that OT implies key agreement, this sets a separation between symmetric cryptography and OT, leading to the conclusion that OT cannot be generated alone by symmetric cryptography. Otherwise, one could use one-way functions to implement key agreement through the OT construction. This poses a threat to all classical OT protocols [51–53] that are based on mathematical assumptions provably broken by a quantum computer [19]. Besides the security problem, asymmetric cryptography tends to be computationally more complex than symmetric cryptography, creating a problem in terms of speed when a large number of OTs are required. The classical post-quantum approach, thrives to find protocols resistant against quantum computer attacks. However, these are still based on complexity problems and are not necessarily less computationally expensive, than the previously mentioned ones.

In parallel to the classical post-quantum approach, the quantum cryptography community tackled this security issue by presenting some OT protocols based on quantum technologies. Intriguingly enough, more than a decade before the first classical OT by Rabin (1981, [44]) was published, Wiesner proposed a similar concept. However, at the time, it was rejected for publication due to the lack of acceptance in the research community. The first published quantum OT (QOT) protocol, known as the BBCS (Bennett-Brassard-Crépeau-Skubiszewska) protocol [28] was only presented in 1992. Remarkably, there is a distinctive difference between classical and quantum OT from a security standpoint, as the latter is proved to be possible assuming only the existence of quantum-hard one-way functions [29, 30]. This means quantum OT requires weaker security assumptions than classical OT.

In this chapter, we review the particular topic of quantum OT. We mainly comment on several important OT protocols, their underlying security models and assumptions. To the best of our knowledge, there is no prior survey dedicated to quantum OT protocols alone. Usually, its analysis is integrated into more general surveys under the topic of "quantum cryptography", leading to a less in-depth exposition of the topic. For reference, we provide some distinctive reviews on the general topic of quantum cryptography [20, 54–60].

This chapter is divided as follows. We start by giving a brief overview of the impossibility results related to quantum OT. Then, we provide an exposition about some of the most well-known quantum OT protocols based on assumptions. Finally, we give a brief overview of OT protocols not covered throughout this thesis.

3.1 Impossibility results

The beginning of the development of quantum OT (QOT) came hand in hand with the development of quantum bit commitment (QBC). In fact, the first proposed QOT protocol (BBCS [28]) reduces QOT to QBC . This sets a distinctive difference between classical and quantum protocols. Although bit commitment (BC) can be reduced to oblivious transfer (OT) [16], the reverse is not true using only classical communication [61]. Therefore, Yao's proof [62] of BBCS protocol [28] gives quantum communications the enhanced quality of having an equivalence between QOT and QBC - they can be reduced to each other - a relation that is not known in the classical realm.

At the time of the BBCS protocol, the quest for unconditionally secure QOT was based on the possibility of unconditional secure QBC. A year later, Brassard et al. presented a QBC protocol [63] named after the authors, BCJL (Brassard-Crépeau-Jozsa-Langlois). However, this work presented a flawed proof of its unconditional security which was generally accepted for some time, until Mayers spotted an issue on it [64]. Just one year after, Lo and Chau [65], and Mayers [66] independently proved unconditional QBC to be impossible. Nevertheless, the existence of unconditionally secure QOT not based on QBC was still put as an open question [54] even after the so-called no-go theorems [65, 66]. However, Lo was able to prove directly that unconditionally secure QOT is also impossible [67]. He concluded this as a corollary of a more general result that states that secure two-party computations which allow only one of the parties to learn the result (one-side secure two-party computation) cannot be unconditionally secure. Lo's results triggered a line of research on the possibility of two-sided secure two-party computation (both parties are allowed to learn the result without having access to the other party's inputs), which was also proved by Colbeck to be impossible [68] and extended in subsequent works [69–71]. For a more in-depth review of the impossibility results presented by Lo, Chau and Mayers, we refer the interested reader to the following works [61, 72].

Although the impossibility results have been well accepted in the quantum cryptography community, there was some criticism regarding the generality of the results [73–76]. This line of research reflects the view put forward by Yuen [73] in the first of these papers: "Since there is no known characterization of all possible QBC protocols, logically there can really be no general impossibility proof, strong or not, even if it were indeed impossible to have an unconditionally secure QBC protocol." In parallel, subsequent analyses were carried out, reaffirming the general belief of impossibility [77–79]. However, most of the discord has ended with Ariano et al. proof [80] in 2007, giving an impossibility proof covering all conceivable protocols based on classical and quantum information theory. Subsequent work digested Ariano et al. [80] work, trying to present more succinct proofs [81–83] and to translate it into categorical quantum mechanics language [84–86].

Facing these impossibility results, the quantum cryptography community followed two main paths:

1. Develop OT protocols under some assumptions. These could be based on limiting the technological power of the adversary (e.g. noisy-storage model, relativistic protocols, isolated-qubit model) or assuming the security of additional functionalities (e.g. bit commitment).

2. Develop OT protocols with a relaxed security definition. These allow the adversary to extract, with a given probability, some information (partial or total) about the honest party input/output. This approach leads to the concepts of weak OT and weak private database query.

In the next section, we explore protocols that produce a special primitive called *oblivious keys* as an intermediate step.

3.2 BBCS-based protocols

In this section, we explore protocols that circumvent the no-go theorems [65, 66] through assumptions. Some of the presented solutions are based on one-way functions, which are believed to be quantum-hard [29, 30, 87], and others rely on technological or physical limitations of the adversaries [88–93]. The latter are qualitatively different from complexity-based assumptions on which post-quantum protocols rely. Also, all these assumptions have the important property that they only have to hold during the execution of the protocol for its security to be preserved. In other words, even if the assumptions lose their validity at some later point in time, the security of the protocol is not compromised. This property is commonly known as *everlasting* security [94]. Everlasting security is also a major distinctive feature of quantum protocols when compared with classical cryptographic approaches.

We start by presenting the first QOT protocol. Then, we see how this protocol led to the development of two assumption models: $\mathcal{F}_{\mathrm{COM}}-$hybrid model and the limited-quantum-storage model.

3.2.1 BBCS protocol

Notation conflict: $\mathcal{F}$ to denote universal functions

In 1983, Wiesner came up with the idea of *quantum conjugate coding* [95]. This technique is the main building block of many important quantum cryptographic protocols [26, 96, 97], including quantum oblivious transfer [28]. It also goes under the name of *quantum multiplexing* [97], *quantum coding* [98] or *BB84 coding* [61]. In quantum conjugate coding we encode classical information in two conjugate (non-orthogonal) bases. This allows us to have the distinctive property that measuring on one basis destroys the encoded information on the corresponding conjugate basis. So, when bit 0 and 1 are encoded by these two bases, no measurement is able to perfectly distinguish the states. We will be using the following bases in the two-dimensional Hilbert space $\mathcal{H}_2$:

- Computational basis: $+ := \{|0\rangle_+, |1\rangle_+\}$;
- Hadamard basis: $\times := \{|0\rangle_\times, |1\rangle_\times\} = \left\{\frac{1}{\sqrt{2}}\big(|0\rangle_+ + |1\rangle_+\big), \frac{1}{\sqrt{2}}\big(|0\rangle_+ - |1\rangle_+\big)\right\}$.

Throughout this chapter we abuse the notation and consider that the set of bases $\{+, \times\}$ can be associated with the binary set $\{0, 1\}$. $+$ is associated with 0 and $\times$ with 1. This is specially useful to compare strings of bases from different parties, i.e. the XOR operation ($\oplus$) between two vectors $\boldsymbol{\theta}^{\mathsf{A}}, \boldsymbol{\theta}^{\mathsf{B}} \in \{+, \times\}^n$ is defined as the XOR between the corresponding binary vectors $\boldsymbol{\theta}^{\mathsf{A}}, \boldsymbol{\theta}^{\mathsf{B}} \in \{0, 1\}^n$.

Protocol [28]. The first proposal of a quantum oblivious transfer protocol is presented in Figure 3.1 and it is called after its creators, Bennett-Brassard-Crépeau-Skubiszewska (BBCS). It builds on top of the quantum conjugate coding technique. Alice starts by using this encoding to generate a set of qubits that are subsequently randomly measured by Bob. These two steps make up the first phase of the BB84 QKD protocol. For this reason, this is called the *BB84 phase*. Next, both parties use the output bits obtained from Bob and the random elements generated by Alice to share a special type of key, known as *oblivious key*. This is achieved when Alice reveals her bases $\boldsymbol{\theta}^{\mathsf{A}}$ to Bob. Using the oblivious key as a resource, Alice can then obliviously send one of the messages m_0, m_1 to Bob, ensuring that he is only able to know one of the messages. This is achieved using a two-universal family of hash functions $\mathcal{F}$ from $\{0,1\}^{n/2}$ to $\{0,1\}^l$. Recall, we use the notation $s \leftarrow_\$ S$ to describe a situation where an element s is drawn uniformly at random from the set S.

Π^{BBCS} protocol

Parameters: n, security parameter; $\mathcal{F}$ two-universal family of hash functions.
Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

BB84 phase:

1. Alice generates random bits $\boldsymbol{x}^{\mathsf{A}} \leftarrow_\$ \{0,1\}^n$ and random bases $\boldsymbol{\theta}^{\mathsf{A}} \leftarrow_\$ \{+,\times\}^n$. Sends the state $|\boldsymbol{x}^{\mathsf{A}}\rangle_{\boldsymbol{\theta}^{\mathsf{A}}}$ to Bob.

2. Bob randomly chooses bases $\boldsymbol{\theta}^{\mathsf{B}} \leftarrow_\$ \{+,\times\}^n$ to measure the received qubits. We denote by $\boldsymbol{x}^{\mathsf{B}}$ his output bits.

Oblivious key phase:

3. Alice reveals to Bob the bases $\boldsymbol{\theta}^{\mathsf{A}}$ used during the *BB84 phase* and sets his oblivious key to $\mathsf{ok}^{\mathsf{A}} := \boldsymbol{x}^{\mathsf{A}}$.

4. Bob computes $\mathsf{e}^{\mathsf{B}} = \boldsymbol{\theta}^{\mathsf{B}} \oplus \boldsymbol{\theta}^{\mathsf{A}}$ and sets $\mathsf{ok}^{\mathsf{B}} := \boldsymbol{x}^{\mathsf{B}}$.

Transfer phase:

5. Bob defines $I_0 = \{i : \mathsf{e}^{\mathsf{B}}_i = 0\}$ and $I_1 = \{i : \mathsf{e}^{\mathsf{B}}_i = 1\}$ and sends the $(I_b, I_{b\oplus 1})$ to Alice.

6. Alice picks two uniformly random hash functions $f_0, f_1 \in \mathcal{F}$, computes the pair of strings (s_0, s_1) as $s_i = m_i \oplus f_i(\mathsf{ok}^{\mathsf{A}}_{I_{b\oplus i}})$ and sends the pairs (f_0, f_1) and (s_0, s_1) to Bob.

7. Bob computes $m_b = s_b \oplus f_i(\mathsf{ok}^{\mathsf{B}}_{I_0})$.

Alice's output: $\perp$.
Bob's output: m_b.

Figure 3.1: BBCS OT protocol.

Oblivious keys. As we saw in the BBCS protocol, oblivious keys can be used as a resource to produce OT instances. In fact, we can draw a comparison between standard encryption keys and oblivious keys. In the same way as standard keys are the resource that allows the encryption of a specific message, oblivious keys are the resource that enables the performance of OT with messages. In other words, encryption methods consume standard keys, while OT methods consume oblivious keys. The term, oblivious key, was used for the first time by Fehr and Schaffner [99] referring to random OT. However, under a subtle different concept, it was put forth by Jakobi et al. [100] and used to implement private database queries (PDQ). Also, in a recent work, Lemus et al. [101] presented the concept of oblivious key applied to OT protocols. We can define it as follows.

Definition 1 (Oblivious key)**.** *An oblivious key shared between two parties, Alice and Bob, is a tuple* $\mathsf{ok} := \big(\mathsf{ok}^{\mathsf{A}}, (\mathsf{ok}^{\mathsf{B}}, \mathsf{e}^{\mathsf{B}})\big)$ *where* ok^{A} *is Alice's key,* ok^{B} *is Bob's key and* e^{B} *is Bob's signal string.* e^{B} *indicates which indexes of* ok^{A} *and* ok^{B} *are correlated and which indexes are uncorrelated, i.e.* $\mathsf{e}^{\mathsf{B}}_i = 0$ *when the corresponding indexes are correlated and* $\mathsf{e}^{\mathsf{B}}_i = 1$ *when they are not.*

Note that, for some index i, when two index elements $\mathsf{ok}^{\mathsf{A}}_i$ and $\mathsf{ok}^{\mathsf{B}}_i$ are correlated, $\mathsf{ok}^{\mathsf{A}}_i = \mathsf{ok}^{\mathsf{B}}_i$. However, when they are uncorrelated, they are drawn independently. This means that both index elements may either be equal or different. Consider the following oblivious key $\mathsf{ok} = (001101101101, (000101001100, 101000110001))$ as an example. We can check it is a well strucured oblivious key:

$$\left.\begin{array}{r|cccccccccccc} \mathsf{ok}^{\mathsf{A}}: & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \mathsf{ok}^{\mathsf{B}}: & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \mathsf{e}^{\mathsf{B}}: & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right\}\mathsf{ok}$$

It is worth stressing that oblivious keys are independent of the sender's messages m_0, m_1 and are not the same as random OT. In fact, as Alice does not know the groups of indexes I_0 and I_1 computed by Bob after the basis revelation, Alice does not have her messages fully defined. A similar concept was defined by König et al. [90] under the name of *weak string erasure.*

Security. Regarding security, the BBCS protocol is unconditionally secure against dishonest Alice. Intuitively, this comes from the fact that Alice does not receive any information from Bob other than some set of indexes I_0. However, the BBCS protocol is insecure against dishonest Bob. In its original paper [28], the authors describe a memory attack that provides Bob complete knowledge on both messages m_0 and m_1 without being detected. This can be achieved by having the receiver delay his measurements in step 2 to some moment after step 3. This procedure is commonly called the memory attack as it requires quantum memory to hold the states until step 3. The authors suggest that, for the protocol to be secure, the receiver has to be forced to measure the received states at step 2. In the following sections, we present two common approaches to tackle this issue. We may assume the existence of commitments or set physical assumptions that constrain Bob from delaying his measurement.

3.2.2 BBCS in the $\mathcal{F}_{\mathbf{COM}}$−hybrid model

As mentioned in the previous section, a secure BBCS protocol requires Bob to measure his qubits in step 2. In this section, we follow the suggestion from the original BBCS paper [28] and fix this loophole using a commitment scheme. Since we assume we have access to some commitment scheme, we call it $\mathcal{F}_{\mathbf{COM}}$−hybrid model[1].

$\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocol

Parameters: n, security parameter; $\mathcal{F}$ two-universal family of hash functions.
Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

BB84 phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Cut and choose phase:

3. Bob commits to the bases used and the measured bits, i.e. $\mathbf{COM}\big(\boldsymbol{\theta}^{\mathsf{B}}, \boldsymbol{x}^{\mathsf{B}}\big)$, and sends to Alice.
4. Alice asks Bob to open a subset T of commitments (e.g. $n/2$ elements) and receives $\{\theta_i^{\mathsf{B}}, x_i^{\mathsf{B}}\}_{i \in T}$.
5. In case any opening is not correct or $x_i^{\mathsf{B}} \neq x_i^{\mathsf{A}}$ for $\theta_i^{\mathsf{B}} = \theta_i^{\mathsf{A}}$, abort. Otherwise, proceed.

Oblivious key phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Transfer phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Alice's output: $\perp$.
Bob's output: m_b.

Figure 3.2: BBCS OT protocol in the $\mathcal{F}_{\mathbf{COM}}$−hybrid model.

[1] The notation $\mathcal{F}_{\mathbf{COM}}$ is commonly used for ideal functionalities. However, here we abuse the notation by using $\mathcal{F}_{\mathbf{COM}}$ to refer to any commitment scheme (including the ideal commitment functionality).

Protocol. The modified BBCS (Figure 3.2) adds a *cut and choose* phase that makes use of a commitment scheme **COM** to check whether Bob measured his qubits in step 2 or not. It goes as follows. Bob commits to the bases used to measure the qubits in the *BB84 phase* and the resulting output bits. Then,Alice chooses a subset of qubits to be tested and asks Bob to open the corresponding commitments of the bases and output elements. If no inconsistency is found, both parties can proceed with the protocol. Note that the size of the testing subset has to be proportional to n (security parameter), as this guarantees that the rest of the qubits were measured by Bob with overwhelming probability in n.

Security. Formally proving the security of this protocol led to a long line of research [27–30, 62, 99, 102–107]. Earlier proofs from the 90's started by analyzing the security of the protocol against limited adversaries that were only able to do individual measurements [103]. Then, Yao [62] was able to prove its security against more general adversaries capable of doing fully coherent measurements. Although these initial works [62, 103, 104] were important to start developing a QOT security proof, they were based on unsatisfactory security definitions. At the time of these initial works, there was no composability framework [99, 106] under which the security of the protocol could be considered. In modern quantum cryptography, these protocols are commonly proved in some quantum simulation-paradigm frameworks [27, 90, 99, 106]. In these paradigms, the security is proved by showing that an adversary in a real execution of the protocol cannot cheat more than what he is allowed in an ideal execution, which is secure by definition. This is commonly proved by utilizing an entity, called simulator, whose role is to guarantee that a real execution of the protocol is indistinguishable from an ideal execution. Moreover, they measured the adversary's information through average-case measures (e.g. Collision Entropy, Mutual Information) which are proven to be weak security measures when applied to cryptography [108, 109].

More desirable worst-case measures started to be applied to quantum oblivious transfer around a decade later [110, 111]. These were based on the concept of *min-entropy* [108, 109], $H_{\min}$, which, intuitively, reflects the maximum probability of an event to happen. More precisely, in order to prove security against dishonest Bob, one is interested in measuring Bob's min-entropy on Alice's oblivious key ok^{A} conditioned on some quantum side information E he may has, i.e. $H_{\min}(\mathsf{ok}^{\mathsf{A}}|E)$. Informally, for a bipartite classical-quantum state ρ_{XE} the conditional min-entropy $H_{\min}(X|E)$ is given by

$$H_{\min}(X|E)_{\rho_{XE}} := -\log P_{guess}(X|E),$$

where $P_{guess}(X|E)$ is the probability the adversary guesses the value x maximized over all possible measurements. Damgård et al. [27] were able to prove the stand-alone QOT security when equipped with this min-entropy measure and with the quantum simulation-paradigm framework developed by Fehr and Schaffner [99]. Their argument to prove the security of the protocol against dishonest Bob can be summarized as follows. The cut and choose phase ensures that Bob's conditional min-entropy on the elements of ok^{A} belonging to I_1 (indexes with uncorrelated elements between Alice's and Bob's oblivious keys) is lower-bounded by some value that is proportional to the security parameter, i.e. $H_{\min}(\mathsf{ok}^{\mathsf{A}}_{I_1}|E) \geq n\lambda$ for some $\lambda > 0$. Note that this is equivalent to derive an upper bound on the guessing probability $P_{guess}(\mathsf{ok}^{\mathsf{A}}_{I_1}|E) \leq 2^{-n\lambda}$. Having deduced an expression for λ, they proceed by applying a random hash function f from a two-universal family $\mathcal{F}$, $f \leftarrow_{\$} \mathcal{F}$. This final step ensures that $f(\mathsf{ok}^{\mathsf{A}}_{I_1})$ is statistically indistinguishable from uniform (privacy amplification theorem [111–113]). The proof provided by Damgård et al. [27] was extended by Unruh [106] to the quantum Universal Composable (UC) model, making use of ideal commitments. Now, a natural question arises:

Which commitment schemes can be used to render simulation-based security?

Commitment scheme. The work by Aaronson [87] presented a non-constructive proof that "indicates that collision-resistant hashing might still be possible in a quantum setting", giving confidence in the use of commitment schemes based on quantum-hard one-way functions in the $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocol. Hopefully, it was shown that commitment schemes can be built from any one-way function [114–116], including quantum-hard one-way functions. Although it is intuitive to plug in into $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ a commitment scheme derived from a quantum-hard one-way function, this does not necessarily render a simulation-based secure protocol. This happens because the nature of the commitment scheme can make the simulation-based proof difficult or even impossible. For a detailed discussion see [29].

Indeed, the commitment scheme must be quantum secure. Also, the simulator must have access to two intriguing properties: *extractability* and *equivocality*. Extractability means the simulator can extract the committed value from a malicious committer. Equivocal means the simulator can change the value of a committed value at a later time. Although it seems counter-intuitive to use a commitment scheme where we can violate both security properties (hiding and biding properties), it is fundamental to prove its security. Extractability is used by the simulator to prove security against the dishonest sender and equivocality is used by the simulator to prove security against the dishonest receiver. In the literature, there have been some proposals of the commitment schemes COM with these properties based on:

- Quantum-hard one-way functions [29, 30];

- Common Reference String (CRS) model [106, 117];
- Bounded-quantum-storage model [118];
- Quantum hardness of the Learning With Errors assumption [27].

Composability. The integration of secure OT executions in secure multiparty protocols [12] should not lead to security breaches. Although it seems intuitive to assume that a secure OT protocol can be integrated within more complex protocols, proving this is highly non-trivial as it is not clear *a priori* under which circumstances protocols can be composed [119].

The first step towards composability properties is the development of simulation based-security. However, this does not necessarily imply composability (see Section 4.2 of [119] for more details), as a composability framework is also required. In the literature, there have been some proposals for such a framework. In summary, Fehr and Schaffner [99] developed a composability framework that allows sequential composition of quantum protocols in a classical environment. The works developed by Ben-Or and Mayers [120] and Unruh [106, 121] extended the classical Universal Composability model [122] to a quantum setting (quantum-UC model), allowing concurrent composability. Maurer and Renner [123] developed a more general composability framework that does not depend on the models of computation, communication, and adversary behaviour. More recently, Broadbent and Karvonen [86] created an abstract model of composable security in terms of category theory. Up until now, and to the best of our knowledge, the composable security of the protocol $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ was only proven in the Fehr and Schaffner model [99] by Damgård et al. [27] and in the quantum-UC by Unruh [106].

3.2.3 BBCS in the limited-quantum-storage model

In this section, we review protocols based on the limited-quantum-storage model. The protocols developed under this model avoid the no-go theorems because they rely their security on reasonable assumptions regarding the storage capabilities of both parties. Under this model, there are mainly two research lines. One was started by Damgård, Fehr, Salvail and Schaffner [88], who developed the bounded-storage model. In this model, the parties can only store a limited number of qubits. The other research line was initiated by Wehner, Schaffner and Terhal [89], who developed the noisy-storage model. In this model the parties can store *all* qubits. However, they are assumed to be unstable, i.e. they only have imperfect noisy storage of qubits that forces some decoherence. In both models, the adversaries are forced to use their quantum memories as both parties have to wait a predetermined time (Δt) during the protocol.

3.2.4 Bounded-quantum-storage model

In the bounded-quantum-storage model or BQS model for short, we assume that, during the waiting time Δt, the adversaries are only able to store a fraction $0 < \gamma < 1$ of the transmitted qubits, i.e. the adversary is only able to keep $q = n\gamma$ qubits. The parameter γ is commonly called the storage rate.

Protocol. The protocol in the BQS model, $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$, is very similar to the BBCS protocol $\Pi^{\mathbf{BBCS}}$ presented in Figure 3.1. The difference is that both parties have to wait a predetermined time (Δt) after step 2. This protocol is presented in Figure 3.3.

$\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ protocol

Parameters: n, security parameter; $\mathcal{F}$ two-universal family of hash functions.
Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

BB84 phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Waiting time phase:

3. Both parties wait time Δt.

Oblivious key phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Transfer phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Alice's output: $\perp$.
Bob's output: m_b.

Figure 3.3: BBCS OT protocol in the bounded-quantum-storage model.

Security. We just comment on the security against dishonest Bob because the justification for the security against dishonest Alice is the same as in the original BBCS protocol, $\Pi^{\mathbf{BBCS}}$ (see Section 3.2.1).

Under the BQS assumption, the waiting time (Δt) effectively prevents Bob from holding *a large fraction* of qubits until Alice reveals the bases choices $\boldsymbol{\theta}^{\mathsf{A}}$ used during the *BB84 phase*. This comes from the fact that a dishonest Bob is forced to measure a fraction of the qubits, leading him to lose information about Alice's bases $\boldsymbol{\theta}^{\mathsf{A}}$.

More specifically, Damgård et al. [111] showed that, with overwhelming probability, the loss of information about Alice's oblivious key ($\mathsf{ok}^{\mathsf{A}}_{I_1}$) is described by a lower bound on the min-entropy [57]

$$H_{\min}(\mathsf{ok}^{\mathsf{A}}_{I_1}|E) \geq \frac{1}{4}n - \gamma n - l - 1.$$

Similarly to the $\mathcal{F}_{\mathbf{COM}}-$hybrid model, the min-entropy value has to be bounded by a factor proportional to the security parameter n. To render a positive bound, we derive an upper bound on the fraction of qubits that can be saved in the receiver's quantum memory, while preserving the security of the protocol, i.e. $\gamma < \frac{1}{4}$.

The above upper bound was later improved by König et al. [90] to $\gamma < \frac{1}{2}$. The authors also showed that the BQS model is a special case of the noisy-quantum-storage model. Subsequently, based on higher-dimensional mutually unbiased bases, Mandayam and Wehner [124] presented a protocol that is still secure when an adversary cannot store even a small fraction of the transmitted pulses. In this latter work, the storage rate γ approaches 1 for increasing dimension.

Composability. The initial proofs given by Damgård et al. [88, 111] were only developed under the stand-alone security model [125]. In this model the composability of the protocol is not guaranteed to be secure. These proofs were extended by Wehner and Wullschleger [125] to a simulation-based framework that guarantees sequential composition. Also, in a parallel work, Fehr and Schaffner developed a sequential composability framework under which $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ is secure considering the BQS model.

The more desirable quantum-UC framework was extended by Unruh and combined with the BQS model [118]. In Unruh's work, he developed the concept of BQS-UC security which, as in UC security, implies a very similar composition theorem. The only difference is that in the BQS-UC framework we have to keep track of the quantum memory-bound used by the machines activated during the protocol. Under this framework, Unruh follows a different approach as he does not use the protocol $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ (Figure 3.3). He presents a BQS-UC secure commitment protocol and composes it with the $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocol (Figure 3.2) in order to get a constant-round protocol that BQS-UC-emulates any two-party functionality.

3.2.5 Noisy-quantum-storage model

The noisy-quantum-storage model, or NQS model for short, is a generalization of the BQS model. In the NQS model, the adversaries are allowed to keep any fraction ν of the transmitted qubits (including the case $\nu = 1$) but their quantum memory is assumed to be noisy [90], i.e. it is impossible to store qubits for some amount of time (Δt) without undergoing decoherence.

More formally, the decoherence process of the qubits in the noisy storage is described by a completely positive trace preserving (CPTP) map (also called channel) $\mathcal{C} : \mathcal{B}(\mathcal{H}_{\text{in}}) \to \mathcal{B}(\mathcal{H}_{\text{out}})$, where $\mathcal{H}_{\text{in/out}}$ is the Hilbert space of the stored qubits before (in) and after (out) the storing period Δt and $\mathcal{B}(\mathcal{H})$ is the set of positive semi-definite operators with unitary trace acting on an Hilbert space $\mathcal{H}$. $\mathcal{C}$ receives a quantum state $\rho \in \mathcal{H}_{\text{in}}$ at time t and outputs a quantum state $\rho' \in \mathcal{H}_{\text{out}}$ at a later time $t + \Delta t$.

With this formulation, we can easily see that the BQS model is a particular case of the NQS. In BQS, the channel is of the form $\mathcal{C} = \mathbb{1}^{\otimes \nu n}$, where the storage rate ν is the fraction of transmitted qubits stored in the quantum memory. The most studied scenario is restricted to $n-$fold quantum channels, i.e. $\mathcal{C} = \mathcal{N}^{\otimes \nu n}$ [89, 90, 126], where the channel $\mathcal{N}$ is applied independently to each individual stored qubit. In this particular case, it is possible to derive specific security parameters.

Protocols. The protocol from BQS model $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ is also considered to be secure in the NQS model [126]. However, the first proposed protocol analysed in this general NQS model was developed by König et al. [90]. This protocol draws inspiration from the research line initiated by Cachin, Crépeau and Marcil [127] about classical OT in the bounded-classical-storage model [128, 129]. Similar to these works [127–129], the protocol presented by König et al. [90] uses the following two important techniques in its classical post-processing phase: encoding of sets and interactive hashing. The former is defined as an injective function $\mathsf{Enc} : \{0,1\}^t \to T$, where T is a set of all subsets of $[n]$ with size $n/4$. The latter is a two-party protocol between Alice and Bob with the following specifications. Bob inputs some message W^t and both parties receive two messages W_0^t and W_1^t such that there exists some $b \in \{0,1\}$ with $W_b^t = W^t$. The index b is unknown to Alice, and Bob has little control over the choice of the other message W^t, i.e. it is randomly chosen by the functionality.

In this section, we only present the naïve protocol presented in the original paper [90] as it is enough to give an intuition on the protocol. Although both $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathbf{nqs}}$ protocols are different, we keep a similar notation for a comparison purpose. The protocol $\Pi^{\mathbf{BBCS}}_{\mathbf{nqs}}$ (Figure 3.4) goes as follows. The first two phases (*BB84* and *Waiting time*) are the same as in $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ (Figure 3.3). Then, both parties generate a very similar resource to oblivious keys, named *weak string erasure* (WSE). After the WSE generation, Alice also holds the totality of the key ok^{A}, while Bob holds a fourth of this key, i.e. the tuple $(I, \mathsf{ok}^{\mathsf{B}} := \mathsf{ok}^{\mathsf{A}}_I)$ where I is the set of indexes they measured in the same basis and its size is given by $|I| = \frac{n}{4}$. Then, along with a method of encoding sets into binary strings, both parties use interactive hashing to generate two index subsets, I_0 and I_1. The two subsets (I_0 and I_1) together with two 2−universal hash functions are enough for Alice to generate her output messages (m_0, m_1) and for Bob to get his bit choice along with the corresponding message (b, m_b). For more details on the protocols for encodings of sets and interactive hashing, we refer to Ding et al. [128] and Savvides [129].

Security. Based on the original BQS protocol (Figure 3.3), the first proofs in the NQS model were developed by Schaffner, Wehner and Terhal [89, 130]. However, in these initial works, the authors only considered individual-storage attacks, where the adversary treats all incoming qubits equally. Subsequently, Schaffner [126] was able to prove the security of $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ against arbitrary attacks in the more general NQS model defined by König et al. [90].

In this more general NQS model, the security of both protocols $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathbf{nqs}}$ (Figures 3.3 and 3.4) against a dishonest receiver depends on the ability to set a lower-bound on the min-entropy of the "unknown" key $\mathsf{ok}^{\mathsf{A}}_{I_{1-b}}$ given the receiver's quantum side information. His quantum side information is given by the output of the quantum channel $\mathcal{C}$ when applied to the received states. More formally, one has to lower-bound the expression $H_{\min}\left(\mathsf{ok}^{\mathsf{A}}_{I_{1-b}} | \mathcal{C}\left(Q_{\text{in}}\right)\right)$, where Q_{in} denotes the subsystem of the received states before undergoing decoherence. It is proven [90] that this lower-bound depends on the receiver's maximal success probability of correctly decoding a randomly chosen n-bit string $x \in \{0,1\}^n$ sent over the quantum channel $\mathcal{C}$, i.e. $P^{\mathcal{C}}_{\text{succ}}(n)$.

For particular channels $\mathcal{C} = \mathcal{N}^{\otimes \nu}$, König et al. [90] concluded that security in the NQS model can be obtained in case

$$c_{\mathcal{N}} \cdot \nu < \frac{1}{2},$$

where $c_{\mathcal{N}}$ is the classical capacity of quantum channels $\mathcal{N}$ satisfying a particular property (strong-converse property).

3.2.6 Experimental attacks

Although QKD and QOT protocols are proved to be theoretically secure, experimental implementations may come with loopholes that allow to break their security. This mismatch between theory and practice comes from the fact that theoretical proofs usually assume that the physical apparatus of honest parties cannot be hacked. However, imperfections in both the generation and measurement the qubits can be exploited in multiple ways to perform quantum attacks. We refer the interested reader to proper review articles [131, 132] on QKD attacks and possible mitigation measures. Here, we briefly discuss the

Naïve $\Pi_{\mathbf{nqs}}^{\mathbf{BBCS}}$ **protocol**

Parameters: n, security parameter; $\mathcal{F}$ two-universal family of hash functions.
Alice's input: $\perp$.
Bob's input: $\perp$.

BB84 phase: Same as in $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

Waiting time phase: Same as in $\Pi_{\mathbf{bqs}}^{\mathbf{BBCS}}$ (Figure 3.3).

Weak String Erasure phase: Similar to *Oblivious key phase* of $\Pi^{\mathbf{BBCS}}$ (Figure 3.1).

4. Alice reveals to Bob the bases $\boldsymbol{\theta}^{\mathsf{A}}$ used during the *BB84 phase* and sets her oblivious key to $\mathsf{ok}^{\mathsf{A}} := \boldsymbol{x}^{\mathsf{A}}$.
5. Bob computes $\mathsf{e}^{\mathsf{B}} = \boldsymbol{\theta}^{\mathsf{B}} \oplus \boldsymbol{\theta}^{\mathsf{A}}$. Then, he defines $I = \{i : \mathsf{e}_i^{\mathsf{B}} = 0\}$ and sets $\mathsf{ok}^{\mathsf{B}} := \boldsymbol{x}_I^{\mathsf{B}}$.
6. If $|I| < n/4$, Bob randomly adds elements to I and pads the corresponding positions in ok^{B} with 0s. Otherwise, he randomly truncates I to size $n/4$, and deletes the corresponding values in ok^{B}.

Interactive hashing phase:

7. Alice and Bob execute interactive hashing with Bob's input W to be equal to a description of $I = \mathsf{Enc}(W)$. They interpret the outputs W_0 and W_1 as descriptions of subsets I_0 and I_1 of $[n]$.

Transfer phase:

5. Alice generates random $f_0, f_1 \leftarrow_{\$} \mathcal{F}$ and sends them to Bob.
6. Alice computes the pair of messages (m_0, m_1) as $m_i = f_i(\mathsf{ok}_{I_i}^{\mathsf{A}})$.
7. Bob computes $b \in \{0,1\}$ by comparing $I = I_b$ and computes $m_b = f_b(\mathsf{ok}_I^{\mathsf{B}})$.

S **output:** $(m_0, m_1) \in \{0,1\}^l$ (two messages).
R **output:** (b, m_b) where $b \in \{0,1\}$ (bit choice).

Figure 3.4: BBCS OT protocol in the noisy-quantum-storage model.

impact of these attacks on BBCS-based QOT protocols.

QOT attacks

It is important to stress that there is a fundamental difference between QKD and QOT protocols. In QKD, both parties can cooperate to detect an external attack, whereas, in QOT, both parties are distrustful of each other. Moreover, QKD external attacks pressupose that the adversary has physical access to the quantum channel and is able to play some sort of man-in-the-middle attack. Regarding QOT protocols, both parties are already linked by a quantum channel. Therefore, in principle, QOT attacks require less engineering effort to succeed as the adversary is already using the quantum channel.

According to the security properties of QOT, Alice must not know Bob's bit b and Bob must not know m_{1-b}. Regarding BBCS-based QOT protocols, its security depends on the security requirements of oblivious keys. Informally, this means that Alice must not be able to know which set of indexes is known by Bob (i.e. e^{B}) and Bob must have limited knowledge on Alice's key (i.e. ok^{A}). These two pieces of information (e^{B} and ok^{A}) can be easily deduced if the adversary has access to the quantum bases used by the other party ($\boldsymbol{\theta}^{\mathsf{A}}$ or $\boldsymbol{\theta}^{\mathsf{B}}$). Indeed, Alice gets e^{B} by computing $\boldsymbol{\theta}^{\mathsf{B}} \oplus \boldsymbol{\theta}^{\mathsf{A}}$ and Bob gets ok^{A} by measuring all the qubits with Alice's bases $\boldsymbol{\theta}^{\mathsf{A}}$. Therefore, the main aim of the adversary is to use his quantum channel to gain some information (or control) about the set of bases used by the other. Two of the most common attacks on quantum systems are faked-state attacks [133] (FSA) and trojan-horses attacks [134] (THA). The former targets measurement apparatus only and the latter can target both preparation and measurement apparatus. In a prepare-and-measure setting, FSA can only be used by Alice (sender) while THA can be used by both. For the sake of exposition, let us see how these two approaches can be used to attack both $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocols. The attacks on $\Pi^{\mathbf{BBCS}}_{\mathbf{nqs}}$ follow the same reasoning but the notation vary slightly.

We denote by $\tilde{\boldsymbol{\theta}}^{\mathsf{B}}_{J} \leftarrow \mathcal{A}_{\mathbf{qok}}(J)$ Alice's quantum hacking procedure ($\mathcal{A}_{\mathbf{qok}}(J)$) that breaks the security requirements of oblivious keys and provides her with Bob's bases ($\tilde{\boldsymbol{\theta}}^{\mathsf{B}}_{J}$) from index set J. Similarly for Bob, i.e. $\tilde{\boldsymbol{\theta}}^{\mathsf{A}}_{J} \leftarrow \mathcal{B}_{\mathbf{qok}}(J)$.

FSA attacks. These attacks can be performed with well crafted optical signals that allow Alice to take control over Bob's measurement outcomes. In summary, as described by Jain et al. [135], when both parties' bases coincide, Bob's detector clicks; when these are orthogonal, he gets no detection event ($\perp$). In other words, Alice forces Bob to only use the measurements where their bases coincide. So, the indexes corresponding to no detection events will be discarded by both parties whereas the others will be used in the rest of the protocol. This way, Alice gains full knowledge about Bob's bases and can easily distinguish I_0 from I_1. Note that Alice does not have to attack all measurement turns. She only needs one successful FSA to guess one basis. This happens with high

$\Pi^{\mathsf{A}}_{\mathbf{FSA}}$ attack

Alice's input: set of indexes J of size q.

1. Alice performs some faked-state attack $\left\{\tilde{\theta}^{\mathsf{B}}_j\right\}_{j\in J} \leftarrow \mathcal{A}_{\mathbf{qok}}(J)$ where $\tilde{\theta}^{\mathsf{B}}_j \in \{+,\times\}$ or $\tilde{\theta}^{\mathsf{B}}_j = \perp$.
2. If $\exists j \in J$ such that $\tilde{\theta}^{\mathsf{B}}_j \neq \perp$:
   (a) $b = 0$ if $j \in I_b$, ;
   (b) $b = 1$ if $j \notin I_b$.
3. Otherwise, sets $b = \perp$.

Alice's output: b.

Figure 3.5: Alice faked-state attack to $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocols.

probability in the number of attacks q,

$$Pr[\text{Success Alice attack in } q \text{ rounds}] = 1 - \left(\frac{1}{2}\right)^q.$$

From this basis, Alice can deduce to which set (I_0 or I_1) the corresponding index (j) belongs. As Bob computes his message m_b with the set where their basis coincide, and since Alice computes both messsages m_0 and m_1 out of both sets, she can determine Bob's message m_b. Indeed, m_b will be the message that comes from the set where j belongs. The attack $\Pi^{\mathsf{A}}_{\mathbf{FSA}}$ against both $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ is summarized in Figure 3.5.

THA attacks. These types of attacks are performed by sending bright pulses into the equipment under attack and scanning through the different reflections to obtain the bases used. Likewise the FSA, Alice only needs to successfully find one basis used by Bob. By comparing her basis and Bob's basis to that particular turn, she can find Bob's bit b. This attack $\Pi^{\mathsf{A}}_{\mathbf{THA}}$ is summarized in Figure 3.6.

Bob's attack through THA is more challenging. Not only he has to successfully guess *all* Alice's bases, he also has to be able to correctly measure the corresponding qubits after leaking the sender's bases. Without the help of quantum memories, this procedure is much more difficult to succeed. Bob's attack $\Pi^{\mathsf{B}}_{\mathbf{THA}}$ is summarized in Figure 3.7.

$\Pi^{\mathsf{A}}_{\mathbf{THA}}$ attack

Alice's input: one index element, j.

1. Alice performs some trojan-horse attack $\left\{\tilde{\theta}^{\mathsf{B}}_j\right\} \leftarrow \mathcal{A}_{\mathbf{qok}}(i)$ where $\tilde{\theta}^{\mathsf{B}}_j \in \{+, \times\}$.
2. Alice compares the received basis $\tilde{\theta}^{\mathsf{B}}_j$ with her corresponding base θ^{A}_j. Denote by $\tilde{\mathsf{e}}^{\mathsf{B}}_j := \tilde{\theta}^{\mathsf{B}}_j \oplus \theta^{\mathsf{A}}_j$.
3. Upon receiving I_b from R:
   (a) $b = \tilde{\mathsf{e}}^{\mathsf{B}}_j$ if $j \in I_b$;
   (b) $b = 1 - \tilde{\mathsf{e}}^{\mathsf{B}}_j$ if $j \notin I_b$.

Alice's output: b.

Figure 3.6: Alice trojan-horse attack to $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocols.

$\Pi^{\mathsf{B}}_{\mathbf{THA}}$ attack

Parameters: n, security parameter..

1. Bob performs some trojan-horse attack to all qubits sent by Alice, i.e. $\left\{\tilde{\theta}^{\mathsf{A}}_i\right\}_{i\in[n]} \leftarrow \mathcal{B}_{\mathbf{qok}}([n])$ where $\tilde{\theta}^{\mathsf{A}}_i \in \{+, \times\}$.
2. Bob measures the received states $\left|\boldsymbol{x}^{\mathrm{A}}\right\rangle_{\boldsymbol{\theta}^{\mathrm{A}}}$ with the correct bases, $\left\{\tilde{\theta}^{\mathsf{A}}_i\right\}_{i\in[n]}$.

Bob's output: ok^{A}.

Figure 3.7: Bob trojan-horse attack to $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ and $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ protocols.

Countermeasures

We have seen how two well-known quantum hacking techniques can undermine the security of oblivious keys and, consequently, the security of oblivious transfer. Fortunately, there are some countermeasures that can be applied that prevent such attacks from breaking the system's security. These countermeasures can be divided into two categories: security patches that tackle specific vulnerabilities and novel schemes that allow faulty devices.

Regarding the two presented possible attacks, it is commonly possible to implement security patches that prevent them. FSA can be prevented by placing an additional detector (usually called watchdog) at the entrance of the receiver's measurement device. This detector monitors possible malicious radiation that blinds his detector. Also, THA can be blocked by an isolator placed at both parties entrance devices. However, as mentioned by Jain et al. [135] these two countermeasures only prevent these attacks perfectly in case the isolators and watchdogs work at all desired frequencies, which is not the case in practice.

There is a research line focused on the study of security patches for each technological loophole [136]. However, this approach pursues the difficult task of approximating the experimental implementations to the ideal protocols. It would be more desirable to develop protocols that already consider faulty devices and are robust against any kind of quantum hacking attack. This is the main goal of device-independent (DI) cryptography, where we drop the assumption that quantum devices cannot be controlled by the adversary and we treat them simply as black-boxes [137, 138]. Here, we give a general overview of the state-of-the-art of DI protocols. For a more in-depth description, we refer to the corresponding original works.

Kaniewski-Wehner DI protocol [139]. The first DI protocol of QOT was presented in a joint work by Kaniewski and Wehner [139] and its security proof was improved by Ribeiro et al. [140]. The protocol was proved to be secure in the noisy-quantum-storage (NQS) model as it uses the original NQS protocol $\Pi_{\mathbf{nqs}}^{\mathbf{BBCS}}$ (Figure 4) for trusted devices. It analyzes two cases leading to slightly different protocols.

First, they assume that the devices have the same behaviour every time they are used (memoryless assumption). This assumption allows for testing the devices independently from the actual protocol, leading to a DI protocol in two phases: device-testing phase and protocol phase . Under this memoryless assumption, one can prove that the protocol is secure against general attacks using proof techniques borrowed from [90]. Then, they analyse the case *without* the memoryless assumption. In that case, it is useless to test the devices in advance as they can change their behaviour later. Consequently, the structure of the initial DI protocol (with two well-separated phases) has to be changed to accommodate this more realistic scenario. That is, the rounds for the device-testing phase have to be intertwined with the rounds for the protocol phase.

As a common practice in DI protocols, the DI property comes from some violation of Bell inequalities [141], which ensures a certain level of entanglement. This means that, in the protocol phase, the entanglement-based variant of $\Pi_{\mathbf{nqs}}^{\mathbf{BBCS}}$ must be used. Here, the difference lies in the initial states prepared by Alice, which, for this case, are maximally entangled states $|\Phi^+\rangle\langle\Phi^+|$, where $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The Bell inequality used in this case comes from the Clauser-Holt-Shimony-Horne (CHSH) inequality [142].

Broadbent-Yuen DI protocol [143]. More recently, Broadbent and Yuen [143] used the $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ (Figure 3) to develop a DI protocol in the BQS model. Similar to Kaniewski and Wehner's work, the protocol is secure under the memoryless assumption. However, they do not require non-communication assumptions that ensure security from Bell inequality violations. Instead of using the CSHS inequality, their work uses a recent self-testing protocol [144, 145] based on a post-quantum computational assumption (hardness of Learning with Errors (LWE) problem [146]).

Ribeiro-Wehner MDI protocol [147]. Ribeiro and Wehner [147] developed an OT protocol in the measurement-device-independent (MDI) regime [148] to avoid the technological challenges in the implementation of DI protocols [149]. In this regime, two parties perform QOT with untrusted measurement devices while trusting their sources. In addition, this work was motivated by the fact that, so far, there is no security proof in the DI setting. Furthermore, many attacks on the non device-independent protocols affect the measurement devices rather than the sources [150]. The presented protocol follows the research line of König et al. [90] and start by executing a weak string erasure in the MDI setting (MDI-WSE phase). For this reason, it is also proved to be secure in the NQS model.

The initial MDI-WSE phase goes as follows. Both Alice and Bob send random states $|\boldsymbol{x}^{\mathsf{A}}\rangle_{\boldsymbol{\theta}^{\mathsf{A}}}$ and $|\boldsymbol{x}^{\mathsf{B}}\rangle_{\boldsymbol{\theta}^{\mathsf{B}}}$, respectively, to an external agent that can be controlled by the dishonest party. The external agent performs a Bell measurement on both received states and announces the result. Bob flips his bit according to the announced result to match Alice's bits. Then, both parties follow the $\Pi^{\mathbf{BBCS}}_{\mathbf{nqs}}$ protocol (Figure 4) from the waiting time phase onward. A similar protocol was presented by Zhou et al. [151] which additionally takes into account error estimation to improve the security of the protocol.

Chapter 4

Classical and quantum oblivious transfer

Secure multiparty computation (SMC) has the potential to be a disruptive technology in the realm of data analysis and computation. It enables several parties to compute virtually any function while preserving the privacy of their inputs. However, most of its protocols' security and efficiency relies on the security and efficiency of oblivious transfer (OT). For this reason, it is fundamental to understand the pros and cons of classical and quantum approaches. In this chapter, we start by analysing both the security and efficiency of classical OT protocols. Then, we compare these classical protocols with their quantum analog. However, we note that classical and quantum approaches use different information medium. Also, classical technology is indeed much more mature than quantum technology. These two observations make it dubious how to perform such a comparison.

In Chapter 3, we reviewed several quantum OT protocols and, in particular, we explored BBCS-based QOT protocols. Beyond being resistant to quantum computer attacks, these protocols provide a practical way to perform OT within SMC. These are divided into two independent phases: oblivious key phase and transfer phase. The first phase corresponds to a precomputation phase that uses quantum technologies and is independent of the parties input elements (m_0, m_1 and b). The second phase only uses classical communication and is based on the precomputed elements from the first phase (oblivious keys). It can be argued that the precomputation phase is not so hungry-efficient as the transfer phase. This comes from the fact that the precomputation is independent of the parties' inputs and, thus, can be performed way before starting an SMC execution. Since the classical OT protocols can also be divided into these two phases, we can compare the transfer phase of both quantum and classical approaches. Furthermore, we do not need quantum equipment to be run concurrently witht he SMC execution.

4.1 Classical oblivious transfer

Let us start by presenting the Bellare-Micali (BM) OT protocol [1] based on public key Diffie-Hellman. This exposition aims to shed some light on the issues related to classical OT implementations. The security and efficiency issues explored in this section also apply to most of the major classical protocols [51–53].

We consider $\mathbb{G}_q$ to be a subgroup of $\mathbb{Z}_p^*$ with generator g and order q, where p is prime and $p = 2q+1$. Also, we assume public knowledge on the value of some constant $C \in \mathbb{G}_q$. This constant guarantees that Bob follows the protocol. Also, for simplicity, we assume the protocol uses a random oracle described as a function H. For comparison purposes with quantum OT version presented in Chapter 3, we split the BM OT protocol into two phases: precomputation phase and transfer phase. The first phase sets the necessary resources to execute the oblivious transfer in the second phase. The BM OT protocol Π_{BM} is shown in Fig. 4.1.

Π_{BM} protocol

Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

(Precomputation phase)

1. Bob randomly generates $k \in \mathbb{Z}_q$ and computes g^k.
2. Alice randomly generates $r_0, r_1 \in \mathbb{Z}_q$ and computes g^{r_0} and g^{r_1}.

(Transfer phase)

3. Bob sets $\mathsf{pk}_b := g^k$. Also, he computes $\mathsf{pk}_{b\oplus 1} = C \cdot \mathsf{pk}_b^{-1}$.
4. Bob sends both public keys $(\mathsf{pk}_0, \mathsf{pk}_1)$ to Alice.
5. Alice checks if $(\mathsf{pk}_0, \mathsf{pk}_1)$ were correctly generated by computing their product: $C = \mathsf{pk}_0 \times \mathsf{pk}_1$.
6. Alice computes and sends to Bob the two tuples: $E_0 = (g^{r_0}, H(\mathsf{pk}_0^{r_0}) \oplus m_0)$ and $E_1 = (g^{r_1}, H(\mathsf{pk}_1^{r_1}) \oplus m_1)$ for some hash function H.
7. Bob is now able to compute $H(\mathsf{pk}_b^{r_b})$ and recover m_b.

Alice's output: $\perp$.
Bob's output: m_b.

Figure 4.1: Bellare-Micali classical OT protocol divided into two phases [1].

4.1.1 Security issues

The Bellare-Micali OT protocol is secure if it complies with both the concealing and obliviousness property. The former is achieved because Bob does not send any information that reveals his input bit choice b to Alice. The latter relies on Alice's ability to keep her randomly generated elements r_0 and r_1 private. Thus, the obliviousness property is compromised if Bob is able to compute the discrete logarithm of g^{r_i} for $i = 0, 1$ (discrete logarithm problem).

The hardness of the discrete logarithm problem on cyclic groups is the basis of several other important protocols. Thus, it is crucial to understand its security limits. Nevertheless, it remains to be proven whether, given a general cyclic group $\mathbb{G}_q$ with generator g and order q, there exists a polynomial-time algorithm that computes r from g^r, where $r \in \mathbb{Z}_q$. Indeed, the BM OT protocol's security relies on the assumption that Bob has limited computational power and is not able to compute the discrete logarithm of a general number.

Although the general discrete logarithm problem is not known to be tractable in polynomial-time, there are specific cases where it is possible to compute it efficiently. This leads to some classical attacks where the structure of the cyclic group considered is not robust enough. As an example, if a prime p is randomly generated without ensuring that $p - 1$ contains a big prime p_b in its decomposition, it is possible to use a divide-and-conquer technique [152] along with some other methods (Shank's method [153], Pollard's rho [154], Pollard's lambda [154]) to solve the discrete logarithm problem. In this case, the computation time will only depend on the size of p_b. So, the smaller the prime p_b, the faster the algorithm can be. In order to avoid these types of attacks, it is recommended to use safe primes, i.e. $p = 2q + 1$ prime where q is also prime. However, it is computationally more expensive to find safe primes because they are less frequent when compared with prime numbers. Beyond the cyclic group structure, it is also important to find big enough prime numbers p. Otherwise, it is possible to compute the discrete logarithm in an acceptable time. As reported in [155], after one week of precomputation, it is possible to compute the discrete logarithm in a 512-bit group in one minute by using the number field sieve algorithm. So, by following this method, after a week-long computation, Bob would be able to find both messages m_0 and m_1 of the BM OT protocol in one minute. In an SMC scenario based on the Yao approach [11], where each OT performed corresponds to one input bit of Alice and the chosen group parameters are fixed, Bob would be able to get the keys corresponding to both 0 and 1 bit and, consequently, he would be able to discover all Alice's inputs. Therefore, at the expense of efficiency, it is necessary to use big enough prime numbers (2048-bit or larger), for which these classical attacks could not be feasibly implemented.

We have just seen specific examples where it is possible to break the security of OT protocol using classical techniques. However, it is known that it is possible to break the general discrete logarithm problem with a quantum computer. In 1995, Peter Shor published a quantum algorithm that is able to solve both prime factorization and discrete logarithm problems in polynomial-time [19]. This remarkable finding poses a threat to most of our currently deployed asymmetric cryptographic protocols (Rivest-Shamir-Adleman, elliptic-curve cryptography and Diffie-Hellman key exchange) as they have their security based on these computational assumptions. Therefore, in the BM OT protocol Bob would be able to perform two attacks with the help of a quantum computer:

Quantum attack 1:

1. Bob computes the discrete logarithm of $g^{r_{b\oplus 1}}$ received from Alice using Shor's algorithm, i.e. $r_{b\oplus 1} = \log_g g^{r_{b\oplus 1}}$.
2. Bob is then able to compute $H\big((g^{r_b})^k\big) = H(\mathsf{pk}_b^{r_b})$ and $H(\mathsf{pk}_{b\oplus 1}^{r_{b\oplus 1}})$ and get both messages m_b and m_{b-1}.

Quantum attack 2:

1. Bob computes the discrete logarithm of $\mathsf{pk}_{b\oplus 1}$ with the Shor's algorithm, i.e. $s = \log_g \mathsf{pk}_{b\oplus 1}$.
2. Bob is then able to compute $H\big((g^{r_b})^k\big) = H(\mathsf{pk}_b^{r_b})$ and $H\big((g^{r_{b\oplus 1}})^s\big) = H(\mathsf{pk}_{b\oplus 1}^{r_{b\oplus 1}})$ and get both messages m_b and $m_{b\oplus 1}$.

In the research literature, there are mainly two approaches to tackle this issue: the development of protocols with assumptions on the computational power of quantum computers or the development of protocols that make use of quantum technology. The former is known as post-quantum cryptography [156] and its public-key cryptography protocols are generally more demanding due to the nature of the computational assumptions used. It is also worth stressing that these computational assumptions are still unproven and have survived just a few years of scrutiny, rendering it likely to be attacked in the near future. The latter is known as quantum cryptography [157]. It provides solutions without relying on asymmetric cryptography but it drastically increases the cost of technological equipment required. Finally, it is important to note that, in general, quantum protocols do not suffer from *intercept now - decipher later* attack (everlasting security) because they base their security on quantum theory. On the contrary, this possible threat is always present in protocols based on computational assumptions.

4.1.2 Efficiency issues

In the previous section, we noted that every mitigation process used to increase security would bring a downside in efficiency: generating safe primes is more demanding, computing bigger exponents and module primes is heavier in general, and using post-quantum solutions require stronger computational assumptions and thus tends to increase the computational complexity.

Now, let us understand the efficiency limitations of the BM OT protocol. We start by looking at the operations used in the protocol (random number generation, modular multiplication, modular inversion, modular exponentiation, hash function evaluation, XOR operation) from which the most demanding operation is modular exponentiation. For this reason, the complexity of BM OT heavily depends on the complexity of modular exponentiation. The number of modular exponentiations executed in each phase is summarised in the Table 4.1.

	Alice	Bob
Precomputation phase	2	1
Transfer phase	2	1

Table 4.1: Number of modular exponentiations in the BM protocol for each phase.

One of the most efficient methods to compute general modular exponentiation with $n-$bit numbers is through a square-and-multiply algorithm along with Karatsuba multiplication. The former method takes $\mathcal{O}(n)$ multiplications and the latter has complexity $\mathcal{O}(n^{1.58})$. Thus, the overall method takes $\mathcal{O}(n^{2.58})$ $n-$bit operations [**?**]. To set an overestimation on the OT generation rate, let us only consider the time (in CPU cycles) required to compute all modular exponentiation operations. We can use the following expression:

$$\left(\frac{C_{mexp}}{C_{cycles}} \times N_{mexp}\right)^{-1} \tag{4.1}$$

where C_{mexp} is the number of CPU cycles required to compute one modular exponentiation, C_{cycles} is the CPU frequency (number of cycles per second) and N_{mexp} is the number of modular exponentiations performed in the OT implementation. This expression only renders an overestimation because it depends on both the implementation of the modular exponentiation operation and the CPU frequency used.

Considering a standard CPU operating around 2.5 GHz ($C_{cycles} = 2.5 \times 10^9$ cycles per second) and a very efficient implementation of modular exponentiation [**?**] ($C_{mexp} \sim$

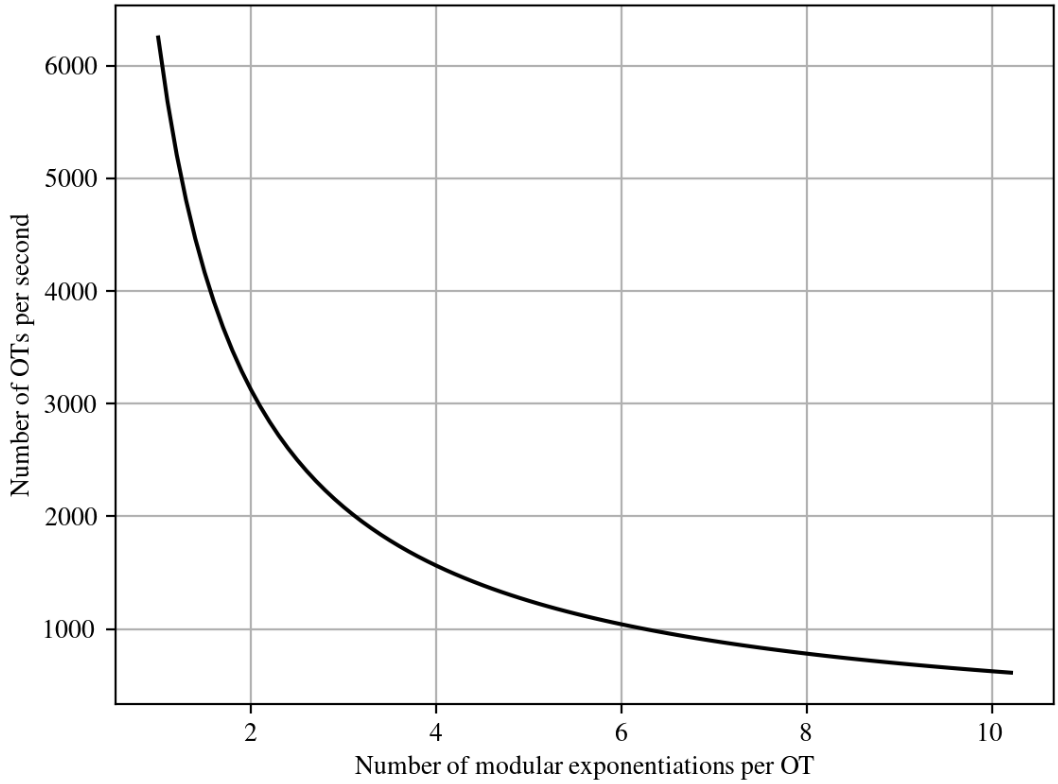

Figure 4.2: Plot of expression (4.1) on the overestimation of OT rate against the number of modular exponentiation operations required per OT.

400 000 CPU cycles), the BM OT protocol would be able to perform at most ~ 1041 BM OTs in one second as represented in Fig. 4.2. Note that this is a very loose overestimation of the number of OT per second. Here, we just took into consideration the computational complexity of modular exponentiation, and we assumed that all the other operations do not have a big impact on the computation time. So, we can conclude that the real OT rate must be well below this threshold. As reported in [2], it takes around 18 ms to generate a similar OT protocol: Naor-Pinkas OT [52], which requires 5 modular exponentiations. This corresponds to a rate of just 56 OT per second.

The OT rates presented above lead to serious constraints on the execution of SMC protocols that rely on OT. The Yao SMC protocol [11] uses boolean circuits to privately compute the desired functionality and requires as many OT as half the number of input wires. Thus, the execution time of the OT phase of the Yao protocol with a 32 000 input boolean circuit would take at least 16 s using our rough OT rate estimation and around 2 min 23 s using Naor-Pinkas OT rate. In a deployment environment where several rounds of the same circuit are evaluated, this approach becomes impractical and higher rates must be achieved.

4.1.3 OT extension protocols

Because most of the required computation to achieve OT comes from asymmetric cryptographic primitives that use modular exponentiation, it would be desirable to substitute it by more efficient methods. Symmetric cryptography has the advantage to be more efficient than asymmetric cryptography. In addition, all known quantum attacks to symmetric cryptography based on the Grover's algorithm only provide a quadratic advantage over classical approaches, which can be mitigated by doubling the size of the symmetric keys [156]. Unfortunately, as we saw in the beginning of Chapter 3, Impagliazzo and Rudich's result [18] implies that OT protocols require asymmetric cryptographic assumptions. This means OT cannot be performed by symmetric cryptographic tools alone.

Nonetheless, researchers developed some OT schemes to circumvent Impagliazzo and Rudich's result using hybrid protocols mixing symmetric and asymmetric cryptography. This idea was introduced by Beaver [**?**], where he showed that it is possible to extend the number of OT using symmetric cryptography when a small number of base OT is created using asymmetric cryptography. Although Beaver's protocol was very inefficient, it paved the way to more efficient implementations [2**? ? ? ?**]. Currently, one of the most efficient protocols is able to generate around 10 million OTs in 2.62 s [2]. Because these protocols use a small number of base OTs and quantum secure symmetric tools, the security of the extended protocol mainly depends on the security of the base OT protocol. Moreover, the protocol that we analyse in Section 4.2.3 [2] is not secure against malicious parties and must only be deployed in a semi-honest environment. Protocols that are secure against malicious parties need an extra consistency check phase which increases their complexity [32**?**] as we see in Section 4.2.4.

4.2 Oblivious transfer complexity analysis

In this section, we compare the complexity of the transfer phase of an optimized version of the BBCS-based QOT protocols ($\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ and $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$) presented before and several well known classical protocols. We start by explaining the optimization.

4.2.1 Optimization

Recall that both $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ and $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ can be divided into two phases: the oblivious key distribution phase (we also call it a *precomputation* phase) and the transfer phase. It is interesting to note that both protocols follow the same steps in the transfer phase. We present the transfer phase of both protocols in Figure 4.3. We slightly rewrite the protocol by using only one hash function (H describes a random oracle) instead of two random hash functions f_0 and f_1. This is done for comparison purposes and because, in practice, H is implemented as a specific hash function, such as SHA.

$\Pi^{\mathbf{BBCS}}$ protocol

Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

Precomputation phase: Alice and Bob generate an oblivious key $(\mathsf{ok}^{\mathsf{A}}, (\mathsf{ok}^{\mathsf{B}}, \mathsf{e}^{\mathsf{B}}))$ according to the corresponding procedure. $\Pi^{\mathbf{BBCS}}_{\mathcal{F}_{\mathbf{COM}}}$ as in Figure 3.2 and $\Pi^{\mathbf{BBCS}}_{\mathbf{bqs}}$ as in Figure 3.3.

Transfer phase:

5. Bob defines $I_0 = \{i : \mathsf{e}^{\mathsf{B}}_i = 0\}$ and $I_1 = \{i : \mathsf{e}^{\mathsf{B}}_i = 1\}$ and sends the pair $(I_b, I_{b\oplus 1})$ to Alice.

6. Alice computes the pair of strings (s_0, s_1) as $s_i = m_i \oplus H(\mathsf{ok}^{\mathsf{A}}_{I_{b\oplus i}})$ and sends to Bob.

7. Bob computes $m_b = s_b \oplus H(\mathsf{ok}^{\mathsf{B}}_{I_0})$.

Alice's output: $\perp$.
Bob's output: m_b.

Figure 4.3: Transfer phase of BBCS-based QOT protocols in the $\mathcal{F}_{\mathbf{COM}}$−hybrid model and bounded-quantum-storage model.

Now, observe that Bob sends two sets $(I_b, I_{b\oplus 1})$ to Alice during the first communication round (Figure 4.3, Step 5). This can be optimized as it is redundant to send both sets of indexes. In fact, with only one set (I_b), Alice is able to know its complement $(\overline{I_b} = I_{b\oplus 1})$). Thus, we end up with the optimized protocol $(\Pi^{\mathbf{BBCS}}_{\mathbf{O}})$ presented in Figure 4.4. This optimized version requires less bandwidth when compared with the initially proposed transfer phase. Now, note that the size of the sets can be identified with a symmetric security parameter κ, as the sets define the keys $(\mathsf{ok}_{I_i},\ i = 0, 1)$ to be used in the hash scheme H. For comparison purposes, we consider that $\kappa = 128$. Furthermore, we can consider the messages, m_0 and m_1, to be garbled circuit's keys. As their size can be $l = 128, 192$ or 256, we consider that $l \sim \kappa$ have the same order of magnitude, meaning they represent the same cost of bits. Therefore, Bob only needs to send l bits to Alice in step 5, leading to an overall reduction of one fourth in the number of bits sent during the transfer phase.

In order to fairly compare the transfer phase of $\Pi^{\mathbf{BBCS}}_{\mathbf{O}}$ protocol with the corresponding

$\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ **protocol**

Alice's input: $(m_0, m_1) \in \{0,1\}^l$ (two messages).
Bob's input: $b \in \{0,1\}$ (bit choice).

Precomputation phase: Alice and Bob generate an oblivious key $(\mathsf{ok}^{\mathsf{A}}, (\mathsf{ok}^{\mathsf{B}}, \mathsf{e}^{\mathsf{B}}))$ according to the corresponding procedure. $\Pi_{\mathcal{F}_{\mathbf{COM}}}^{\mathbf{BBCS}}$ as in Figure 3.2 and $\Pi_{\mathbf{bqs}}^{\mathbf{BBCS}}$ as in Figure 3.3.

Transfer phase:

5. Bob defines $I_0 = \{i : \mathsf{e}_i^{\mathsf{B}} = 0\}$ and $I_1 = \{i : \mathsf{e}_i^{\mathsf{B}} = 1\}$ and **sends only** I_b to Alice.

6. Alice computes the pair of strings (s_0, s_1) as $s_i = m_i \oplus H(\mathsf{ok}_{I_{b\oplus i}}^{\mathsf{A}})$ and sends to Bob.

7. Bob computes $m_b = s_b \oplus H(\mathsf{ok}_{I_0}^{\mathsf{B}})$.

Alice's output: $\perp$.
Bob's output: m_b.

Figure 4.4: Transfer phase of BBCS-based QOT protocols in the $\mathcal{F}_{\mathbf{COM}}$−hybrid model and bounded-quantum-storage model.

phase of other classical protocols, we have to divide the classical protocols in these two phases. We apply the following rule: all the steps used in the protocol that are independent of the messages (m_0 and m_1) and of the bit choice (b) are considered to be part of the precomputation phase, otherwise they are included in the transfer phase. Since the precomputation phase can be executed before the execution of the Yao GC protocol, it is more important to guarantee that the transfer phase has small complexity. Furthermore, we stress that here we will only compare the complexity among the different protocols' transfer phase because their precomputation phase rely on different technologies. Since quantum technologies are still in their infancy and constantly evolving, it is difficult to compare the efficiency with classical approaches. Nevertheless, it is worth noting that the oblivious key phase of $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol is linear in all its security parameters. In fact, as presented by Lemus et al. [101], the time complexity of $\Pi_{\mathcal{F}_{\mathbf{COM}}}^{\mathbf{BBCS}}$ is of the order $\mathcal{O}(\kappa(2l+t))$, where κ is the security parameter of the hash-based commitments, $2l$ is the number of qubits sent used to directly generate the oblivious keys and t is the number of testing qubits.

4.2.2 Classical OT

In section 4.1, we divided the well known Bellare-Micali protocol in these two phases and we observed that it uses three exponentiations during the transfer phase. In Table 4.2 we present the number of required modular exponentiations and communication rounds during the transfer phase of four well known classical protocols that have their security based on the computational hardness of the Discrete Logarithm problem.

Protocol	Exponentiation	Comm. rounds
EGL [51]	3	2
BM [1]	3	2
NP [52]	2	2
SimpleOT [53]	1	2

Table 4.2: Number of modular exponentiation operations and communication rounds executed during the transfer phase of four other classical protocols.

From Table 4.2, we see that the most efficient protocol (SimpleOT [53]) still requires one exponentiation operation and 2 communication rounds. From the above formula (4.1) and setting $C_{mcyles} = 2.5 \times 10^9$, $C_{mexp} = 400\,000$ and $N_{mexp} = 1$, we get an overestimation of around 6000 OT per second. Comparing with the rate achieved by OT extension protocols (10 million OT in 2.62 s), it is still very inefficient.

This means that the current classical OT protocols have a computational complexity limited by $\mathcal{O}(n^{2.58})$ bit operations due to modular exponentiation. The $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol only depends on simple bit operations (XOR, truncation and comparison), meaning its computational complexity is linear in the length of the messages $\mathcal{O}(n)$.

In addition, it is important to stress that none of the above protocols are secure against quantum computer attacks. In order to have classical OT protocols with this level of security, we need to follow post-quantum approaches which may lead to more demanding operations [**?**]. As reported in [**?**], using Kyber key encapsulation based on the module learning with errors (M-LWE) problem [**?**], it takes 24 ms to generate one OT in a LAN network. This leads to a rate of just 41 OT per second, which is even lower than the rate reported by [2] for the Naor-Pinkas [52]: 56 OT per second. In [**? ?**], the authors present a 1-out-of-n OT based on the NTRU post-quantum encryption system [**?**] and compare it with the SimpleOT [53] version. In this case, although Bob and Alice sides are more efficient individually, the overall NTRU OT protocol is still less efficient. For the highest security level, it takes around 1.372 ms to generate one OT with the post-quantum approach, whereas it takes 0.727 ms using the original SimpleOT protocol. These timings lead to the rates of 728 and 1375 OT per second, respectively. It is important to note that these protocols are still prone to *intercept now - decipher later* attacks since they are based on computational assumptions that are only *believed* (and not proved) to be secure against quantum computer attacks.

4.2.3 OT extension

As we explained in section 4.1.3, several techniques based on an hybrid symmetric-asymmetric approach were developed as a way to increase the OT execution rate. These techniques use a small number κ (= 128) of base OT protocols (e.g. EGL, BM, NP, SimpleOT) and extend this resource to m (= 10 000 000) OT executions, where $m >> \kappa$.

Again, to compare OT extension protocols with $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$, we have to decompose them into the same two phases: precomputation phase and transfer phase. In this section, we make a bit-wise comparison of the communication and computational complexity of m executions of $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ and one OT extension execution because OT extension protocols generate a predetermined number (m) of OTs at once. First, we compare $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ with a semi-honest protocol which we call for short ALSZ13, and then, we compare the quantum version with a maliciously secure protocol denoted by KOS15.

ALSZ13 comparison

Let us consider the OT extension protocol proposed in [2] (ALSZ13) and shown in Figure 4.5. At the time of writing, it reports the fastest implementation: 10 million OT in 2.68 seconds. This protocol is originally divided into two phases: initial OT phase and OT extension phase. Note that these two phases correspond exactly to our division of precomputation and transfer phases. Thus, for comparative purposes, we are only interested in the second phase.

In Tables 4.3 and 4.4, we show the computational and communication complexity of both protocols, respectively. In Table 4.3, PRG stands for pseudorandom generator, κ represents the number of base OTs executed in the OT extension precomputation phase, m is the number of final OTs and l is the length of the OT strings. We consider that $l \sim \kappa$ have the same order of magnitude, meaning they represent the same cost of bits. This is so, because $\kappa = 128$ in [2] and the key length used in the garbled circuits are $l = 128, 192$ or 256. Now, we justify the analysis presented in Tables 4.3 and 4.4.

Regarding the ALSZ13 protocol, for every $1 \leq i \leq \kappa$, Bob computes two PRGs in step 4 and Alice computes one PRG in step 5. This accounts for 3κ PRG executions. For every $1 \leq j \leq m$, Alice computes two hash functions in step 6 and Bob computes one

ALSZ13 OT extensions protocol [2]

Alice's input: m pairs (x_j^0, x_j^1), $\forall 1 \leq j \leq m$ of l-bit strings.
Bob's input: m selection bits $\mathbf{r} = (r_1, ..., r_m)$.

Initial OT phase (Precomputation phase)

1. Alice randomly generates a string $\mathbf{s} = (s_1, ..., s_\kappa)$.

2. Bob randomly chooses κ pairs of κ-bit strings $\{(\mathbf{k}_i^0, \mathbf{k}_i^1)\}_{i=1}^{\kappa}$.

3. Bob and Alice execute κ base OTs, where Alice plays the role of the receiver with input $\mathbf{s}$ and Bob plays the role of the sender with messages $(\mathbf{k}_i^0, \mathbf{k}_i^1)$ $\forall 1 \leq i \leq \kappa$.

OT extension phase (Transfer phase)

4. Bob applies a pseudorandom number generator G to $\mathbf{k}_i^0$, i.e. $\mathbf{t}^i = G(\mathbf{k}_i^0)$. Computes $\mathbf{u}^i = \mathbf{t}^i \oplus G(\mathbf{k}_i^1) \oplus \mathbf{r}$ and sends $\mathbf{u}^i$ to Alice for every $1 \leq i \leq \kappa$.

5. Alice computes $\mathbf{q}^i = (s_i \cdot \mathbf{u}^i) \oplus G(\mathbf{k}_i^{s_i})$.

6. Alice sends (y_j^0, y_j^1) for every $1 \leq j \leq m$, where $y_j^0 = x_j^0 \oplus H(j, \mathbf{q}_j)$, $y_j^1 = x_j^1 \oplus H(j, \mathbf{q}_j \oplus \mathbf{s})$ and $\mathbf{q}_j$ is the j-th row of the matrix $Q = [\mathbf{q}^1|...|\mathbf{q}^\kappa]$. Note that, in practice, it is required to transpose Q to access its j-th row.

7. Bob computes $x_j^{r_j} = y_j^{r_j} \oplus H(j, \mathbf{t}_j)$.

Alice's output: $\perp$.
Bob's output: $(x_1^{r_1}, ..., x_m^{r_m})$.

Figure 4.5: Precomputation and transfer phases of OT extensions protocol presented in [2].

hash function. This accounts for $3m$ hash functions. For every $1 \leq i \leq \kappa$, Bob computes two $m-$bit XOR operations in step 4 and Alice computes one $m-$bit XOR operation. For every $1 \leq j \leq m$, Alice computes two $l-$bit XOR operations in step 6 and Bob computes one $l-$bit XOR operation in step 7. Also, for every $1 \leq j \leq m$, Alice computes one $\kappa-$bit XOR operation in step 6. This accounts for $3m\kappa + 3ml + m\kappa$ bitwise XOR operations. For every $1 \leq i \leq \kappa$, Alice computes one $m-$bit AND operation in step 5. Finally, Alice has to perform a matrix inversion which accounts for around $m \log m$ bit operations. The communication complexity is given by the following elements: Bob sends an $m-$bit vector for every $1 \leq i \leq \kappa$ and Alice sends two $l-$bit messages for every $1 \leq j \leq m$. This accounts for $2ml + m\kappa$ bits sent.

Regarding the $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol, for every execution of the protocol, Alice computes two hash functions in step 6 and Bob computes one hash function in step 7. This accounts for $3m$ hash functions. Also, Alice computes two $l-$bit XOR operations in step 6 and Bob computes one $l-$bit XOR operation in step 7. This accounts for $3ml$ bitwise XOR operations. For every execution of the protocol, Alice performs 2κ bitwise comparisons in step 5. Also, Alice computes two $\kappa-$bit truncation in step 6 and Bob computes one $\kappa-$bit truncation in step 7. The communication complexity is given by the following elements: Bob sends a $\kappa-$bit vector and Alice sends two $l-$bit messages, for every execution of the protocol. This accounts for $2ml + m\kappa$ bits sent.

Operation	ALSZ13	$\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$
PRG (AES)	3κ	-
Hash (SHA-1)	$3m$	$3m$
Bitwise XOR	$3m\kappa + 3ml + m\kappa$	$3ml$
Bitwise AND	κm	-
Matrix transposition	$m \log m$	-
Bitwise comparison	-	$2m\kappa$
Bitwise truncation	-	$3m\kappa$

Table 4.3: Computational complexity comparison between ALSZ13 [2] OT extension protocol and $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol from section 4.2.1.

	ALSZ13	$\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$
Bits sent	$2ml + m\kappa$	$2ml + m\kappa$

Table 4.4: Communication complexity comparison between ALSZ13 [2] OT extension protocol and $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol from section 4.2.1.

We have that the communication complexity is exactly the same in both protocols: $\sim 3ml$. So, the OT extension does not have any advantage over $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ during the communication phase. Regarding their computational complexity, we have to compare binary operations executed betwen each protocol.

Firstly, we can see that $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ transfer phase is asymptotically more efficient than ALSZ13 OT extension transfer phase. The computational complexity of OT extension is not linear in the number of OT executions, $\mathcal{O}(m \log m)$, whereas it is linear in the case of $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$, $\mathcal{O}(m)$. Now, let us compare the binary operations between each protocol. Denote by $B_{\text{op}}^{\text{ALSZ13}}$ and $B_{\text{op}}^{\text{BBCS}}$ the number of binary operations executed by ALSZ13 and $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$, respectively. As both protocols execute $3m$ hash functions,we do not take into account their execution. Also, assuming that $\kappa \sim l$, $B_{\text{op}}^{\text{ALSZ13}}$ is roughly given by,

$$\begin{aligned} B_{\text{op}}^{\text{ALSZ13}} &= 3\kappa + 3m\kappa + 3ml + m\kappa + m\kappa + m\log m \\ &= 8m\kappa + 3\kappa + m\log m \end{aligned}$$

and $B_{\text{op}}^{\text{BBCS}} = 8m\kappa$. Here we simplify and assume that 3κ PRGs executions consume only 3κ bit operations. Therefore, ALSZ13 has more $B_{\text{op}}^{\text{ALSZ13}} - B_{\text{op}}^{\text{BBCS}} \geq m\log m$ binary operations than the transfer phase of $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol.

From this, we conclude that $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ transfer phase competes with the ALSZ13 corresponding phase and has the potential to be more efficient. It is important to stress that $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ efficiency performance of the transfer phase comes along with a drastic increase in the security of the protocol. While ALSZ13 protocol relies on the computational assumptions of the base OT, $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol is proved to be secure against quantum computers. Moreover, while ALSZ13 is a semi-honest protocol (assumes well-behaved parties that follow the protocol), $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol is secure against any corrupted party. Indeed, in order to get a fair comparison, we should consider OT extension protocols that are secure against malicious parties. The work developed in [**?**] presented the first protocol in the malicious scenario, which was latter optimized by KOS15 [32] and ALSZ15 [**?**]. Both optimizations carry out one run of the semi-honest OT extension presented in ALSZ13 plus some consistency checks. The protocol presented in [32] adds to ALSZ13 a *check correlation* phase after the transfer phase and the protocol presented in [**?**] adds a *consistency check* phase during the transfer phase. This means that both malicious protocols' transfer phases have greater computational and communication complexity when compared with ALSZ13. Therefore, we can easily deduce that $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ transfer phase has less computational and communication complexity than its classical equivalents with respect to the adversary model used. Next, we compare the KOS15 protocol [32] and $\Pi_{\mathbf{O}}^{\mathbf{BBCS}}$ protocol.

4.2.4 Oblivious Transfer comparison

To implement practical SMC protocols, we need to be able to execute OT with a rate of the order of millions of OT per second. To reach this rate, classical solutions make use of extension algorithms: generate a small number κ of base OT (precomputation phase as in HQOT) and extend them to m ($\kappa << m$) real OT through symmetric cryptography [**?**] (oblivious transfer phase). Currently, the most efficient OT extension protocols developed in the semi-honest model is reported by [2] (ALSZ13) and in the malicious model it is reported by [**?**] (KOS15). In [**?**], the authors showed that the overall complexity in the transfer phase of ALSZ13 is bigger than that of HQOT. Furthermore, they argued that KOS15 complexity is also bigger than HQOT but do not perform a complexity comparison between them. Here, we analyse the complexity of the KOS15 protocol which is implemented in the Libscapi library and we compare it with HQOT.

KOS15 and HQOT comparison

KOS15 protocol is very similar to ALSZ13 with the addition of a *check correlation* phase. This phase ensures that the receiver is well behaved and does not cheat. The KOS15 protocol that generates m l-bit string OT out of κ base OT with computational security given by κ and statistical security given by w is shown in Figure 4.6. Note that in Figure 4.6 we join all the subprotocols presented in the original paper: $\prod_{\mathrm{COTE}}^{\kappa,m'}$, $\prod_{\mathrm{ROT}}^{\kappa,m}$ and $\prod_{\mathrm{DEROT}}^{\kappa,m}$. Also, they identify $\mathbb{Z}_2^{\kappa}$ with the finite field $\mathbb{Z}_{2^\kappa}$ and use "$\cdot$" for multiplication in $\mathbb{Z}_{2^\kappa}$. For example, the element $\boldsymbol{t}_j$ in $\sum_{j=1}^{m'} \boldsymbol{t}_j \cdot \chi_j$ (Figure 4.6, step 10) should be considered in $\mathbb{Z}_{2^\kappa}$.

Similarly to HQOT, the KOS15 starts with a precomputation phase that can be carried out before the actual computation of the OT protocols. However, in the HQOT, the precomputation phase is based on quantum technologies while the transfer phase is solely based on classical methods. Since it is not clear how to compare quantum and classical protocols, we only focus our comparison on the transfer phase of both protocols.

Note that in the original KOS15 paper [**?**] the computation of pseudorandom generator G is carried out in the OT extension phase. However, these 3κ G computations can be executed during the precomputation phase because they do not depend on the input elements. As mentioned before, the additional steps that KOS15 added to the ALSZ13 protocol are steps $9-11$ (check correlation phase). Here, both parties start by calling a random oracle functionality $\mathcal{F}_{\mathrm{Rand}}(\mathbb{F}_{2^\kappa}^{m'})$ that provides them with equal random values. The receiver has to compute twice m' κ-bit sums, m' κ-bit multiplication and sends 2κ bit (x and t) to the sender. Finally, the sender has to compute m' κ-bit sums and m' κ-bit multiplication. We consider karatsuba method for multiplication with complexity $O(\kappa^{1.585})$ and schoolbook addition with complexity $O(\kappa)$. Therefore, we consider that the sum of two κ takes κ bit operations and the multiplication takes $\kappa^{1.585}$.

Denote by $B_{\mathrm{op}}^{\mathrm{KOS15}}$ and $B_{\mathrm{op}}^{\mathrm{HQOT}}$ the number of binary operations executed by KOS15 and HQOT. Without taking into account the execution of $3m$ hash functions and assuming

Operation	KOS15	QOT
Hash (SHA-1)	$3m$	$3m$
Bitwise XOR	$3\kappa m + 3ml + \kappa$	$3ml$
Bitwise AND	κm	-
Matrix transposition	$m \log m$	-
Bitwise comparison	-	$2ml$
Bitwise truncation	-	$3ml$
κ-bit additon	$3(m + (\kappa + w))\kappa$	-
κ-bit mult	$2(m + (\kappa + w))\kappa^{1.58}$	-

Table 4.5: Computation complexity comparison between KOS15 OT extension and HQOT.

that $\kappa \sim l$, $B_{\text{op}}^{\text{KOS15}}$ is roughly given by,

$$\begin{aligned}
B_{\text{op}}^{\text{KOS15}} &= 3\kappa m + 3ml + \kappa \\
&+ \kappa m + m \log m \\
&+ 3(m + (\kappa + w))\kappa \\
&+ 2(m + (\kappa + w))\kappa^{1.58} \\
&= 10m\kappa + \kappa + m \log m \\
&+ 3\kappa^2 + 3\kappa w \\
&+ 2m\kappa^{1.58} + 2\kappa^{2.58} + 2\kappa^{1.58} w
\end{aligned} \qquad \text{span}$$

and $B_{\text{op}}^{\text{HQOT}} = 8m\kappa$. Therefore, KOS15 has more $B_{\text{op}}^{\text{KOS15}} - B_{\text{op}}^{\text{HQOT}} \geq 4m\kappa$ binary operations than HQOT transfer phase. For this estimation, note that we are considering the lower bound $2m\kappa$ instead of $2m\kappa^{1.58}$ and we are not taking into account the implementation of the random oracle $\mathcal{F}_{\text{Rand}}(\mathbb{F}_{2^\kappa}^{m'})$, which would add an extra cost linear in the number of OT executions.

Regarding the communication complexity, the number of bits sent during both ALSZ15 and HQOT is the same. KOS15 only adds κ bits to the communication in ALSZ15 during the check correlation phase. However, since this overhead is independent of m (number of OT executed) its effect is amortized for big m.

General OT extensions protocol [?]

Sender input: m pairs (x_j^0, x_j^1), $\forall 1 \leq j \leq m$ of l-bit strings.
Receiver input: m selection bits $\boldsymbol{r} = (r_1, ..., r_m)$.
Let $m' = m + (\kappa + w)$.

Initial OT phase (Precomputation phase)

1. Alice randomly generates a string $\boldsymbol{s} = (s_1, ..., s_\kappa)$.
2. Bob randomly chooses κ pairs of κ-bit strings $\{(\boldsymbol{k}_i^0, \boldsymbol{k}_i^1)\}_{i=1}^{\kappa}$.
3. Bob and Alice execute κ base OTs, where Alice plays the role of the receiver with input $\boldsymbol{s}$ and Bob plays the role of the sender with messages $(\boldsymbol{k}_i^0, \boldsymbol{k}_i^1)\, \forall 1 \leq i \leq \kappa$.
4. Bob applies a pseudorandom number generator G to $\boldsymbol{k}_i^0$ and $\boldsymbol{k}_i^1$: $\boldsymbol{t}^i = G(\boldsymbol{k}_i^0)$ and $\boldsymbol{t}_1^i = G(\boldsymbol{k}_i^1)$. Also, set $\boldsymbol{T}^i = \boldsymbol{t}^i \oplus \boldsymbol{t}_1^i$.
5. Alice applies G to $\boldsymbol{k}_i^{s_i}$ and sets $\boldsymbol{g}_i^{s_i} = G(\boldsymbol{k}_i^{s_i})$.

OT extension phase (Transfer phase)
Extend

6. Bob generates random elements r_j, for $r \in [m+1, m']$ and resize $\boldsymbol{r} = (r_1, ..., r_m, r_{m+1}, ..., r_{m'})$.
7. Bob computes $\boldsymbol{u}^i = \boldsymbol{T}^i \oplus \boldsymbol{r}$ and sends $\boldsymbol{u}^i$ to Alice for every $1 \leq i \leq \kappa$.
8. Alice computes $\boldsymbol{q}^i = (s_i \times \boldsymbol{u}^i) \oplus \boldsymbol{g}_i^{s_i}$ for every $1 \leq i \leq \kappa$.

Check correlation

9. Sample $(\chi_1, ..., \chi_{m'}) \leftarrow \mathcal{F}_{\text{Rand}}(\mathbb{F}_{2^\kappa}^{m'})$.
10. Bob computes $x = \sum_{j=1}^{m'} r_j \cdot \chi_j$ and $t = \sum_{j=1}^{m'} \boldsymbol{t}_j \cdot \chi_j$, where $\boldsymbol{t}_j$ is the j-th row of the matrix $[\boldsymbol{t}^1|...|\boldsymbol{t}^\kappa]$ and sends these to Alice.
11. Alice computes $q = \sum_{j=1}^{m'} \boldsymbol{q}_j \cdot \chi_j$, where $\boldsymbol{q}_j$ is the j-th row of the matrix $Q = [\boldsymbol{q}^1|...|\boldsymbol{q}^\kappa]$, and checks that $t = q + r \cdot \boldsymbol{s}$. If the check fails, output Abort, otherwise continue.

Randomize and encrypt

11. Alice sends (y_j^0, y_j^1) for every $1 \leq j \leq m$, where $y_j^0 = x_j^0 \oplus H(j, \boldsymbol{q}_j)$, $y_j^1 = x_j^1 \oplus H(j, \boldsymbol{q}_j \oplus \boldsymbol{s})$.
12. Bob computes $x_j^{r_j} = y_j^{r_j} \oplus H(j, \boldsymbol{t}_j)$.

Sender output: $\perp$.
Receiver output: $(x_1^{r_1}, ..., x_m^{r_m})$.

Figure 4.6: Precomputation and transfer phases of OT extensions protocol presented in [?].

Bibliography

[1] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In *Proceedings on Advances in Cryptology*, CRYPTO -89, pages 547–557, Berlin, Heidelberg, 1989. Springer-Verlag.

[2] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 535–548, New York, NY, USA, 2013. Association for Computing Machinery.

[3] Jun Wang. Personal genomes: For one and for all. *Science*, 331(6018):690–690, 2011.

[4] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, 2008.

[5] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[6] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8):e1000167, August 2008.

[7] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, January 2013.

[8] 2016 reform of eu data protection rules. https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016.

[9] Ninghui Li, Min Lyu, Dong Su, and Weining Yang. Differential privacy: From theory to practice. *Synthesis Lectures on Information Security, Privacy, and Trust*, 8(4):1–138, October 2016.

[10] Frederik Armknecht, C. Boyd, Christopher Carr, K. Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. A guide to fully homomorphic encryption. *IACR Cryptol. ePrint Arch.*, 2015:1192, 2015.

[11] A. C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, 1982.

[12] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, October 1986.

[13] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proceedings of the nineteenth annual ACM conference on Theory of computing - STOC '87*. ACM Press, 1987.

[14] Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic encryption and multiparty computation. In *Advances in Cryptology – EUROCRYPT 2011*, pages 169–188. Springer Berlin Heidelberg, 2011.

[15] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Lecture Notes in Computer Science*, pages 643–662. Springer Berlin Heidelberg, 2012.

[16] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing - STOC '88*. ACM Press, 1988.

[17] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Lecture Notes in Computer Science*, pages 643–662. Springer Berlin Heidelberg, 2012.

[18] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 44–61, New York, NY, USA, 1989. Association for Computing Machinery.

[19] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[20] Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, 78(1):351–382, December 2015.

[21] A. N. Pinto, N. A. Silva, A. Almeida, and N. J. Muga. Using quantum technologies to improve fiber optic communication systems. *IEEE Communications Magazine*, 8(51):42–48, August 2013.

[22] Andre Chailloux and Iordanis Kerenidis. Optimal bounds for quantum bit commitment. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. IEEE, October 2011.

[23] André Chailloux and Iordanis Kerenidis. Optimal quantum strong coin flipping. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, October 2009.

[24] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, October 2009.

[25] Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014*, pages 1–18, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[26] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. Secure identification and QKD in the bounded-quantum-storage model. *Theoretical Computer Science*, 560:12–26, December 2014.

[27] Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, pages 408–427, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[28] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 351–366, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[29] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 531–561, Cham, 2021. Springer International Publishing.

[30] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 467–496, Cham, 2021. Springer International Publishing.

[31] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, October 2016.

[32] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure ot extension with optimal overhead. In *Advances in Cryptology - CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 724–741. Springer, August 2015. Date of Acceptance: 08/05/2015.

[33] H. Bechmann-Pasquinucci and W. Tittel. Quantum cryptography using larger alphabets. *Phys. Rev. A*, 61:062308, May 2000.

[34] Thomas Durt, Berthold-Georg Englert, Ingemar Bengtsson, and Karol Zyczkowski. On mutually unbiased bases. *International Journal of Quantum Information*, 08(04):535–640, 2010.

[35] Tian Zhong, Hongchao Zhou, Robert D Horansky, Catherine Lee, Varun B Verma, Adriana E Lita, Alessandro Restelli, Joshua C Bienfang, Richard P Mirin, Thomas Gerrits, Sae Woo Nam, Francesco Marsili, Matthew D Shaw, Zheshen Zhang, Ligong Wang, Dirk Englund, Gregory W Wornell, Jeffrey H Shapiro, and Franco N C Wong. Photon-efficient quantum key distribution using time–energy entanglement with high-dimensional encoding. *New Journal of Physics*, 17(2):022002, 2015.

[36] Frédéric Bouchard, Natalia Herrera Valencia, Florian Brandt, Robert Fickler, Marcus Huber, and Mehul Malik. Measuring azimuthal and radial modes of photons. *Opt. Express*, 26(24):31925–31941, Nov 2018.

[37] Mirdit Doda, Marcus Huber, Gláucia Murta, Matej Pivoluska, Martin Plesch, and Chrysoula Vlachou. Quantum key distribution overcoming extreme noise: Simultaneous subspace coding using high-dimensional entanglement. *Phys. Rev. Applied*, 15:034003, Mar 2021.

[38] Manuel B. Santos, Paulo Mateus, and Armando N. Pinto. Quantum oblivious transfer: A short review. *Entropy*, 24(7):945, July 2022.

[39] Manuel B. Santos, Armando N. Pinto, and Paulo Mateus. Quantum and classical oblivious transfer: A comparative analysis. *IET Quantum Communication*, 2(2):42–53, May 2021.

[40] Manuel B. Santos, Paulo Mateus, and Chrysoula Vlachou. Quantum universally composable oblivious linear evaluation, 2022.

[41] Manuel B. Santos, Ana C. Gomes, Armando N. Pinto, and Paulo Mateus. Private computation of phylogenetic trees based on quantum technologies. *IEEE Access*, 10:38065–38088, 2022.

[42] Manuel B. Santos, Ana C. Gomes, Armando N. Pinto, and Paulo Mateus. Quantum secure multiparty computation of phylogenetic trees of SARS-CoV-2 genome. In *2021 Telecoms Conference (ConfTELE)*. IEEE, February 2021.

[43] Armando N. Pinto, Laura Ortiz, Manuel Santos, Ana C. Gomes, Juan P. Brito, Nelson J. Muga, Nuno A. Silva, Paulo Mateus, and Vicente Martin. Quantum enabled private recognition of composite signals in genome and proteins. In *2020 22nd International Conference on Transparent Optical Networks (ICTON)*. IEEE, July 2020.

[44] Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[45] Yan-Cheng Chang. Single database private information retrieval with logarithmic communication. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, pages 50–61, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[46] Michele Orrù, Emmanuela Orsini, and Peter Scholl. Actively secure 1-out-of-n ot extension with application to private set intersection. In Helena Handschuh, editor, *Topics in Cryptology – CT-RSA 2017*, pages 381–396, Cham, 2017. Springer International Publishing.

[47] Bo Bi, Darong Huang, Bo Mi, Zhenping Deng, and Hongyang Pan. Efficient LBS security-preserving based on NTRU oblivious transfer. *Wireless Personal Communications*, 108(4):2663–2674, May 2019.

[48] Vijay Kumar Yadav, Nitish Andola, Shekhar Verma, and S Venkatesan. A survey of oblivious transfer protocol. *ACM Computing Surveys*, January 2022.

[49] Russel Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 44–61, New York, NY, USA, 1989. Association for Computing Machinery.

[50] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc, 2000.

[51] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, June 1985.

[52] M. Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA '01*, 2001.

[53] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In *Proceedings of the 4th International Conference on Progress in Cryptology – LATINCRYPT 2015 - Volume 9230*, page 40–58, Berlin, Heidelberg, 2015. Springer-Verlag.

[54] Gilles Brassard and Claude Crépeau. 25 years of quantum cryptography. *ACM SIGACT News*, 27(3):13–24, September 1996.

[55] G. Brassard. Brief history of quantum cryptography: a personal perspective. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*. IEEE, 2005.

[56] Jörn Müller-Quade. Quantum cryptography beyond key exchange. *Informatik - Forschung und Entwicklung*, 21(1-2):39–54, September 2006.

[57] Serge Fehr. Quantum cryptography. *Foundations of Physics*, 40(5):494–531, January 2010.

[58] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden. Advances in quantum cryptography. *Advances in Optics and Photonics*, 12(4):1012, December 2020.

[59] Christopher Portmann and Renato Renner. Security in quantum cryptography, 2021.

[60] Shihai Sun and Anqi Huang. A review of security evaluation of practical quantum key distribution system. *Entropy*, 24(2):260, February 2022.

[61] Louis Salvail. *The Search for the Holy Grail in Quantum Cryptography*, pages 183–216. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[62] Andrew Chi-Chih Yao. Security of quantum protocols against coherent measurements. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing - STOC '95*. ACM Press, 1995.

[63] G. Brassard, C. Crepeau, R. Jozsa, and D. Langlois. A quantum bit commitment scheme provably unbreakable by both parties. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*. IEEE, 1993.

[64] Dominic Mayers. The trouble with quantum bit commitment, 1996.

[65] Hoi-Kwong Lo and H. F. Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410–3413, April 1997.

[66] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414–3417, April 1997.

[67] Hoi-Kwong Lo. Insecurity of quantum secure computations. *Physical Review A*, 56(2):1154–1162, August 1997.

[68] Roger Colbeck. Impossibility of secure two-party classical computation. *Physical Review A*, 76(6), December 2007.

[69] Harry Buhrman, Matthias Christandl, and Christian Schaffner. Complete insecurity of quantum protocols for classical two-party computation. *Physical Review Letters*, 109(16), October 2012.

[70] Louis Salvail, Christian Schaffner, and Miroslava Sotáková. Quantifying the leakage of quantum protocols for classical two-party cryptography. *International Journal of Quantum Information*, 13(04):1450041, December 2014.

[71] Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In Amit Sahai, editor, *Theory of Cryptography*, pages 281–296, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[72] Gilles Brassard, Claude Crépeau, Dominic Mayers, and Louis Salvail. A brief review on the impossibility of quantum bit commitment, 1997.

[73] Horace P. Yuen. Unconditionally secure quantum bit commitment is possible, 2000.

[74] Horace P. Yuen. Quantum bit commitment and unconditional security, 2002.

[75] Horace P. Yuen. How to build unconditionally secure quantum bit commitment protocols, 2003.

[76] Chi-Yee Cheung. Quantum bit commitment can be unconditionally secure, 2001.

[77] Jeffrey Bub. The quantum bit commitment theorem. *Foundations of Physics*, 31(5):735–756, 2001.

[78] Chi-Yee Cheung. Secret parameters in quantum bit commitment, 2005.

[79] CHI-YEE CHEUNG. Quantum bit commitment with secret parameters. *International Journal of Modern Physics B*, 21(23n24):4271–4274, September 2007.

[80] Giacomo Mauro D'Ariano, Dennis Kretschmann, Dirk Schlingemann, and Reinhard F. Werner. Reexamination of quantum bit commitment: The possible and the impossible. *Physical Review A*, 76(3), September 2007.

[81] Giulio Chiribella, Giacomo Mauro D'Ariano, and Paolo Perinotti. Probabilistic theories with purification. *Physical Review A*, 81(6), June 2010.

[82] Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, Dirk Schlingemann, and Reinhard Werner. A short impossibility proof of quantum bit commitment. *Physics Letters A*, 377(15):1076–1087, June 2013.

[83] Guang Ping He. Comment on "a short impossibility proof of quantum bit commitment", 2013.

[84] Katriel Cohn-Gordon. Commitment algorithms. Master's thesis, University of Oxford, Oxford, UK, 2012.

[85] Xin Sun, Feifei He, and Quanlong Wang. Impossibility of quantum bit commitment, a categorical perspective. *Axioms*, 9(1):28, March 2020.

[86] Anne Broadbent and Martti Karvonen. Categorical composable cryptography. In Patricia Bouyer and Lutz Schröder, editors, *Foundations of Software Science and Computation Structures*, pages 161–183, Cham, 2022. Springer International Publishing.

[87] Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing - STOC '02*. ACM Press, 2002.

[88] I.B. Damgard, S. Fehr, L. Salvail, and C. Schaffner. Cryptography in the bounded quantum-storage model. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE, 2005.

[89] Stephanie Wehner, Christian Schaffner, and Barbara M. Terhal. Cryptography from noisy storage. *Physical Review Letters*, 100(22), June 2008.

[90] Robert Konig, Stephanie Wehner, and Jürg Wullschleger. Unconditional security from noisy quantum storage. *IEEE Transactions on Information Theory*, 58(3):1962–1984, March 2012.

[91] Yi-Kai Liu. Building one-time memories from isolated qubits. In *Proceedings of the 5th conference on Innovations in theoretical computer science*. ACM, January 2014.

[92] Damián Pitalúa-García. Spacetime-constrained oblivious transfer. *Physical Review A*, 93(6), June 2016.

[93] Adrian Kent. Location-oblivious data transfer with flying entangled qudits. *Physical Review A*, 84(1), July 2011.

[94] Dominique Unruh. Everlasting multi-party computation. *Journal of Cryptology*, 31(4):965–1011, March 2018.

[95] Stephen Wiesner. Conjugate coding. *ACM SIGACT News*, 15(1):78–88, January 1983.

[96] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science*, 560:7–11, December 2014.

[97] Charles H. Bennett, Gilles Brassard, Seth Breidbart, and Stephen Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology*, pages 267–275. Springer US, 1983.

[98] Charles H. Bennett, Gilles Brassard, and Seth Breidbart. Quantum cryptography II: How to re-use a one-time pad safely even if p=NP. *Natural Computing*, 13(4):453–458, October 2014.

[99] Serge Fehr and Christian Schaffner. Composing quantum protocols in a classical environment. In Omer Reingold, editor, *Theory of Cryptography*, pages 350–367, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[100] Markus Jakobi, Christoph Simon, Nicolas Gisin, Jean-Daniel Bancal, Cyril Branciard, Nino Walenta, and Hugo Zbinden. Practical private database queries based on a quantum-key-distribution protocol. *Physical Review A*, 83(2), February 2011.

[101] Mariano Lemus, Mariana F. Ramos, Preeti Yadav, Nuno A. Silva, Nelson J. Muga, André Souto, Nikola Paunković, Paulo Mateus, and Armando N. Pinto. Generation and distribution of quantum oblivious keys for secure multiparty computation. *Applied Sciences*, 10(12):4080, June 2020.

[102] C. Crepeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*. IEEE, 1988.

[103] D. Mayers and L. Salvail. Quantum oblivious transfer is secure against all individual measurements. In *Proceedings Workshop on Physics and Computation. PhysComp '94*. IEEE Comput. Soc. Press, 1994.

[104] Dominic Mayers. Quantum key distribution and string oblivious transfer in noisy channels. In *Advances in Cryptology — CRYPTO '96*, pages 343–357. Springer Berlin Heidelberg, 1996.

[105] Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. Computational collapse of quantum state with application to oblivious transfer. In Moni Naor, editor, *Theory of Cryptography*, pages 374–393, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[106] Dominique Unruh. Universally composable quantum multi-party computation. In *Advances in Cryptology – EUROCRYPT 2010*, pages 486–505. Springer Berlin Heidelberg, 2010.

[107] Niek J. Bouman and Serge Fehr. Sampling in a quantum population, and applications. In *Advances in Cryptology – CRYPTO 2010*, pages 724–741. Springer Berlin Heidelberg, 2010.

[108] Mario Berta, Matthias Christandl, Roger Colbeck, Joseph M. Renes, and Renato Renner. The uncertainty principle in the presence of quantum memory. *Nature Physics*, 6(9):659–662, July 2010.

[109] Marco Tomamichel and Renato Renner. Uncertainty relation for smooth entropies. *Physical Review Letters*, 106(11), March 2011.

[110] Renato Renner. Security of quantum key distribution, 2006.

[111] Ivan B. Damgård, Serge Fehr, Renato Renner, Louis Salvail, and Christian Schaffner. A tight high-order entropic quantum uncertainty relation with applications. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 360–378, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[112] Renato Renner and Robert König. Universally composable privacy amplification against quantum adversaries. In *Theory of Cryptography*, pages 407–425. Springer Berlin Heidelberg, 2005.

[113] Renato Renner. Security of quantum key distribution, 2005.

[114] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991.

[115] Johan HÅstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, January 1999.

[116] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07*. ACM Press, 2007.

[117] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 19–40, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[118] Dominique Unruh. Concurrent composition in the bounded quantum storage model. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 467–486, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[119] Jörn Müller-Quade and Renato Renner. Composability in quantum cryptography. *New Journal of Physics*, 11(8):085006, August 2009.

[120] Michael Ben-Or and Dominic Mayers. General security definition and composability for quantum & classical protocols, 2004.

[121] Dominique Unruh. Simulatable security for quantum protocols, 2004.

[122] Ran Canetti. Universally composable security. *Journal of the ACM*, 67(5):1–94, October 2020.

[123] Ueli Maurer and Renato Renner. Abstract cryptography. In Bernard Chazelle, editor, *The Second Symposium on Innovations in Computer Science, ICS 2011*, pages 1–21. Tsinghua University Press, 1 2011.

[124] Prabha Mandayam and Stephanie Wehner. Achieving the physical limits of the bounded-storage model. *Physical Review A*, 83(2), February 2011.

[125] Stephanie Wehner and Jürg Wullschleger. Composable security in the bounded-quantum-storage model. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 604–615, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[126] Christian Schaffner. Simple protocols for oblivious transfer and secure identification in the noisy-quantum-storage model. *Physical Review A*, 82(3), September 2010.

[127] C. Cachin, C. Crepeau, and J. Marcil. Oblivious transfer with a memory-bounded receiver. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. IEEE Comput. Soc, 1998.

[128] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In Moni Naor, editor, *Theory of Cryptography*, pages 446–472, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[129] George. Savvides. *Interactive hashing and reductions between Oblivious Transfer variants*. PhD thesis, McGill University, School of Computer Science, 2007.

[130] Christian Schaffner, Barbara M. Terhal, and Stephanie Wehner. Robust cryptography in the noisy-quantum-storage model. *Quantum Inf. Comput.*, 9(11&12):963–996, 2009.

[131] Hoi-Kwong Lo, Marcos Curty, and Kiyoshi Tamaki. Secure quantum key distribution. *Nature Photonics*, 8(8):595–604, July 2014.

[132] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden. Advances in quantum cryptography. *Advances in Optics and Photonics*, 12(4):1012, December 2020.

[133] Vadim Makarov and Dag R. Hjelme. Faked states attack on quantum cryptosystems. *Journal of Modern Optics*, 52(5):691–705, March 2005.

[134] N. Gisin, S. Fasel, B. Kraus, H. Zbinden, and G. Ribordy. Trojan-horse attacks on quantum-key-distribution systems. *Physical Review A*, 73(2), February 2006.

[135] Nitin Jain, Birgit Stiller, Imran Khan, Dominique Elser, Christoph Marquardt, and Gerd Leuchs. Attacks on practical quantum key distribution systems (and how to prevent them). *Contemporary Physics*, 57(3):366–387, March 2016.

[136] Nitin Jain, Birgit Stiller, Imran Khan, Dominique Elser, Christoph Marquardt, and Gerd Leuchs. Attacks on practical quantum key distribution systems (and how to prevent them). *Contemporary Physics*, 57(3):366–387, March 2016.

[137] Dominic Mayers and Andrew Chi-Chih Yao. Self testing quantum apparatus. *Quantum Inf. Comput.*, 4(4):273–286, 2004.

[138] Artur K. Ekert. Quantum cryptography based on bell's theorem. *Physical Review Letters*, 67(6):661–663, August 1991.

[139] Jedrzej Kaniewski and Stephanie Wehner. Device-independent two-party cryptography secure against sequential attacks. *New Journal of Physics*, 18(5):055004, May 2016.

[140] Jérémy Ribeiro, Le Phuc Thinh, Jedrzej Kaniewski, Jonas Helsen, and Stephanie Wehner. Device independence for two-party cryptography and position verification with memoryless devices. *Physical Review A*, 97(6), June 2018.

[141] Antonio Acín, Nicolas Gisin, and Lluis Masanes. From bell's theorem to secure quantum key distribution. *Physical Review Letters*, 97(12), September 2006.

[142] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, October 1969.

[143] Anne Broadbent and Peter Yuen. Device-independent oblivious transfer from the bounded-quantum-storage-model and computational assumptions, 2021.

[144] Tony Metger, Yfke Dulek, Andrea Coladangelo, and Rotem Arnon-Friedman. Device-independent quantum key distribution from computational assumptions. *New Journal of Physics*, 23(12):123021, December 2021.

[145] Tony Metger and Thomas Vidick. Self-testing of a single quantum device under computational assumptions. *Quantum*, 5:544, September 2021.

[146] Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Paper 2015/939, 2015. `https://eprint.iacr.org/2015/939`.

[147] Jeremy Ribeiro and Stephanie Wehner. On bit commitment and oblivious transfer in measurement-device independent settings, 2020.

[148] Hoi-Kwong Lo, Marcos Curty, and Bing Qi. Measurement-device-independent quantum key distribution. *Physical Review Letters*, 108(13), March 2012.

[149] G Murta, S B van Dam, J Ribeiro, R Hanson, and S Wehner. Towards a realization of device-independent quantum key distribution. *Quantum Science and Technology*, 4(3):035011, July 2019.

[150] Shihan Sajeed, Igor Radchenko, Sarah Kaiser, Jean-Philippe Bourgoin, Anna Pappa, Laurent Monat, Matthieu Legré, and Vadim Makarov. Attacks exploiting deviation of mean photon number in quantum key distribution and coin tossing. *Physical Review A*, 91(3), March 2015.

[151] Zishuai Zhou, Qisheng Guang, Chaohui Gao, Dong Jiang, and Lijun Chen. Measurement-device-independent two-party cryptography with error estimation. *Sensors*, 20(21):6351, November 2020.

[152] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

[153] D. Shanks. Class number, a theory of factorization and genera. *In Proc. Symp. Pure Math., Providence, R.I.: American Mathematical Society*, 20:415–440, 1971.

[154] John M. Pollard. Monte Carlo methods for index computation mod p. *Mathematics of Computation*, 32:918–924, 1978.

[155] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. Imperfect forward secrecy: How diffie-hellman fails in practice. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 5–17, New York, NY, USA, 2015. Association for Computing Machinery.

[156] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 2017.

[157] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden. Advances in quantum cryptography. *Adv. Opt. Photon.*, 12(4):1012–1236, Dec 2020.