# Quantum Computation and Privacy

Stephanie Wehner

# Quantum Computation
## and
# Privacy

Master's Thesis in Theoretical Computer Science

Universiteit van Amsterdam
Institute for Logic, Language and Computation

Stephanie Wehner

Supervisor: Prof.dr. H.M. Buhrman
Co-supervisor: Dr. R.M. de Wolf
Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

The investigations were partially supported by the EU fifth framework project RESQ, IST-2001-37559.

Cover design by Frans Bartels.

# Contents

# Preface

*"Les étoiles sont belles, à cause d'une fleur que l'on ne voit pas..."*
*- Le Petit Prince, Antoine de Saint-Exupéry.*

Quantum mechanics is one of the most intriguing subjects to study. The world works inherently differently on very small scales and can no longer be described by means of classical physics corresponding to our everyday intuition. Contrary to classical computing, quantum computation is based on the rules of quantum mechanics. It not only allows for more efficient local computations, but also has far-reaching effects on multi-party protocols. In this thesis, we investigate two cryptographic primitives for privacy protection using quantum computing: private information retrieval and anonymous transmissions.

The goal of private information retrieval (PIR) is to allow a user to retrieve any item from a database while preventing the database from learning the index of the requested entry. One simple solution would be to send the entire database to the user, who can then select the desired entry. This ensures that the database surely cannot learn anything. However, this approach is clearly not very practical. Imagine, for example, that for looking up the arrival time of flight in a database you would have to download all flight records of the world. Can we do better than sending the entire database? The main question of private information retrieval is thus: What is the minimum number of bits we really need to retrieve the database record and ensure the privacy of the user? Or, in other words, what is the lower bound for the communication complexity of PIR? Here we prove new lower bounds for the communication complexity of unconditionally secure *classical* private information retrieval using a novel *quantum* trick. Our result implies that several known PIR schemes are close to optimal. Closely related to the problem of private information retrieval are locally decodable codes (LDC). These are error-correcting codes that allow efficient decoding of individual bits of the encoded data from the codeword, without having to read all of it. Thus we can make a small number of queries to the codeword, which will each give us part of this codeword. We then combine all these parts to reconstruct pieces of the encoded data. This is particularly useful in applications where we wish to encode a large chunk of data, but are only interested in extracting small pieces at a time, for example, we want to encode an entire book, but want to retrieve only a single page. The main question here is: How large does the codeword have to be so that we still stand a chance of reconstructing the piece of data we are interested in? We show new lower bounds for the code length of 2-query LDCs. Our results generalize those of Goldreich et al. [47], who proved roughly the same bounds for only *linear* LDCs and PIRs. Like earlier work by Kerenidis and de Wolf [53], our classical lower bounds are proved using quantum computational techniques. The new trick used here is a tight analysis of how well a 2-input function can be computed from a quantum superposition of both inputs. Thus starting out with a quantum state $|0, a_0\rangle + |1, a_1\rangle$, what are our chances of computing $f(a_0, a_1)$?

We also study the problem of anonymous transmissions. In this setting, members of a group of participants want to send and receive data, without revealing their identity to any other participant or to an outside observer. We present a quantum protocol for sending and receiving classical bits anonymously that is resistant to collusions of participants and, unlike all known classical protocols, prevents later reconstruction of the sender. It appears that entangled quantum states are uniquely suited for anonymous transmissions. We then extend this protocol to provide sender and recipient untraceability for qubits as well. In the process we also introduce a new primitive called anonymous entanglement, which may be useful for many other protocols as well. Our protocol furthermore provides an example where $O(n^2)$ pairwise private shared random bits can be replaced by an $n$-qubit shared entangled state. This is an interesting tradeoff, as the $n$-qubit entangled state is equally shared by everyone, whereas each classical random bit is known to only two participants.

## Outline

Chapter 1 gives an informal introduction to quantum computation. Read Appendix A for an overview of linear algebra used in quantum computing. In Chapter 2 you can find the basic cryptographic terminology, as far as it is necessary to follow the remainder of this text. Part I is concerned entirely with private information retrieval and locally decodable codes. Chapter 3 explains the notion of private information retrieval and gives a number of example protocols to aid your intuition. It also gives the known lower and upper bounds for PIR. Chapter 4 then gives an introduction to locally decodable codes together with two examples. Known lower bounds can also be found in this chapter. The main part of Part I is Chapter 5, where we prove new lower bounds for both private information retrieval and locally decodable codes. Part II deals with the problem of anonymous transmissions. In Chapter 6 you can find an overview of the problem together with the description of a known classical protocol. Finally, Chapter 7 presents a new quantum protocol for anonymously transmitting classical and quantum bits.

## Personal Motivation

When I finished the class of Harry Buhrman two years ago, I decided that I wanted to do my final year project on a topic in the area of quantum computing. Since then I have tried to learn more about the subject of quantum mechanics by following several physics classes, which proved to be extremely interesting. Luckily, I had the opportunity to work at CWI, where I could see how people are conducting research in this area. Ronald de Wolf introduced me to the problem of private information retrieval and helped me work on this topic. Many of our initial approaches to PIR turned out to be flawed and several months were turned into a cycle of excitement and disappointment. This was a great learning experience for me and gave me a peek at what it is like to step outside the borders of our existing knowledge.

I am very happy I found a topic which combined my interest in quantum computing and my other long term interest of security. After quitting my job in the area of practical computer security last year, I had initially decided to say good bye to security problems, for the second time in my life. Nevertheless, I keep gravitating back to security issues. In the area of computing I was always particularly fond of distributed computation and protocols involving multiple participants. Not only does this setting generate many concurrency problems, it also creates interesting scenarios if some of the participants decide not to follow the protocol. Multi-player protocols are much like games, which seem to become a lot more interesting once some of the players turn into adversaries and try to "win" or defeat the underlying protocol.

Frankly, I have to confess that I just greatly enjoyed working on the problems in this text.

## Acknowledgments

First and foremost I would like to express my gratitude to Ronald de Wolf, who introduced me to the problem of private information retrieval and has been an interminable source of advice. For giving me a glimpse of research and for the many enjoyable and occasionally frustrating discussions, which resulted in our post to the quant-ph archive, I am deeply grateful.

Many thanks also to Harry Buhrman for giving me the opportunity to join his group at CWI and to attend the RESQ meeting in Barcelona. Furthermore, I would like to thank everyone at CWI for making this a very enjoyable place to be. Thanks to Leen Torenvliet for taking part in my thesis committee. Thanks also to Artur Ekert for letting me visit the group in Cambridge and everyone there to make my visit such a nice experience.

For advice, and interesting, useful and often fun discussions I would like to thank Barbara Terhal, Jonathan Oppenheim, Steven de Rooij, Rudi Cilibrasi, John Tromp, Troy Lee, Robert Špalek, Wim van Dam, Adrian Kent, Serge Massar and especially Matthias Christandl. Thanks also to Steven for proof-reading this thesis. For guiding me through the jungle of bureaucracy I want to thank Lucia Creveld. Thanks to Marja Hegt for providing me with a constant supply of soda cans. I would also like to thank Andreas Pfitzmann for sending me a copy of [68] and [69], and Frans Bartels for creating the image on the cover.

Furthermore, I want to thank Torsten Grust and Peter Honeyman, who advised me to enter university in the first place. For their friendship I would also like to thank Cor Bosman and Erik Bos. Special thanks to Alex Le Heux for many things exceeding words. Finally, I would like to thank my family and friends for being who they are.

Amsterdam<br>May, 2004.

Stephanie Wehner

# Chapter 1

# Quantum Computing

## 1.1 Introduction

Quantum computing is based on quantum physics, which describes how the world works on very small scales. This contrasts with classical forms of computation, which both in theory and implementation are based on classical physics alone.

In the following we give an overview of quantum computing required for the remainder of this text. We first take a brief look at the history of quantum computation. We then review the quantum mechanical principles underlying quantum computation. Finally, we show how to construct quantum circuits to perform computations on quantum states.

## 1.2 History

In the early 1980's Richard Feynman considered the question of simulating quantum mechanical systems, which appears to be extremely difficult using classical computing models. In particular, he raised the question whether a computer based on the principles of quantum mechanics could perform better at such simulations. This marks the advent of the new field of *Quantum Computation*, which has sparked off considerable interest since. In the mid-eighties David Deutsch went on to develop a digital variant [37]. A plethora of quantum algorithms has been discovered since then, which illustrate the power of this new computing model [65]. The most well known is Peter Shor's algorithm for factoring integers presented in 1994 [74]. This was significant, since factoring is one of the problems considered hard in classical computing and the security of many encryption techniques, such as RSA, rests on this assumption. Once large scale quantum computers are built, these cryptographic systems no longer protect against adversaries. However, Wiesner [85] (1970), and Bennett and Brassard [20] (1984) introduced *quantum cryptography* which does not depend on factoring. Quantum cryptography differs from conventional cryptography in that quantum states are used to exchange secret keys securely. Later on, Lov Grover [48] showed how to find an element in an unsorted list of size $N$ in time proportional to $\sqrt{N}$, which is better than any classical algorithm.

Whereas quantum cryptography is already available commercially [80], large scale quantum computers have yet to be built. So far the largest quantum computer constructed in a lab can only work with 7 qubits [34]. Creating large scale quantum computers confronts researchers with many similar problems as were encountered when classical computing machines were first developed: noise-reduction and error-correction. Physical implementation of quantum computers is a very active field of research. However, it may still take decades before all complications can be overcome. Nevertheless other applications such as quantum cryptography, which do not require a full-blown quantum computer may be possible much earlier.

## 1.3 Quantum Mechanics

Contrary to classical computing we perform computations using quantum states, which have some very intriguing properties, as we will discover shortly. In this section we require some linear algebra, the essentials of which are explained in Appendix A.

### 1.3.1 States

A quantum state can be described by a *state vector* living in a complex vector space, also called the *state space*. Such a complex vector space with inner product is called a *Hilbert space*. We will first examine the basic building block, the quantum bit or qubit, and how these are combined. We then take a look at the concept of superpositions. As it turns out, a qubit can be in several classical states all at once!

**Qubits**

The simplest conceivable quantum state is a single *qubit*. A qubit has a two-dimensional state space, whose orthonormal basis is generally denoted as $\{|0\rangle, |1\rangle\}$. These two basis states can be written as vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

In the course of this text, we will refer to this as the *computational basis* in two dimensions. We can now use this idea to write any arbitrary qubit as a vector in this state space, given by

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

where $\alpha$ and $\beta$ are complex numbers. State vectors are required to be *unit vectors.* This means that they are *normalized*, that is $\| \, |\Psi\rangle \, \| = 1$ or equivalently $|\alpha|^2 + |\beta|^2 = 1$. Intuitively, $\alpha$ and $\beta$ give an indication of “how much” of $|0\rangle$ and $|1\rangle$ is found in $|\Psi\rangle$.

We can now combine such individual qubits to higher dimensional quantum states. Two quantum mechanical systems are combined using the *tensor product*. For example we can write a system of two qubits $|\Psi\rangle = \alpha_1|0\rangle + \alpha_2|1\rangle$ and $|\Phi\rangle = \beta_1|0\rangle + \beta_2|1\rangle$ as

$$|\Psi\rangle \otimes |\Phi\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \otimes \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\beta_1 \\ \alpha_1\beta_2 \\ \alpha_2\beta_1 \\ \alpha_2\beta_2 \end{pmatrix}$$

Instead of $|\Psi\rangle \otimes |\Phi\rangle$, we will also use the shorthand notations $|\Phi\rangle|\Psi\rangle$ and $|\Phi, \Psi\rangle$. Formally the state $|\Psi\rangle|\Phi\rangle$ is called a *product state*. As a simple example consider the 2 qubit state

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Note that not all states in the same state space can be written as the product of two individual states. As we will discover shortly, states which can not be decomposed this way play an important role in quantum computing and are called *entangled* states.

In general, we can write any $d$-dimensional quantum state as a vector in $\mathbb{C}^d$. Here we will make use of the Dirac notation to write

$$|\Psi\rangle = \sum_{i=1}^{d} \alpha_i|\psi_i\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_d \end{pmatrix}$$

A vector written in the form $|\cdot\rangle$ is called a *ket*. Its so-called dual, $\langle\Psi|$, is defined as the conjugate transpose of $|\Psi\rangle$. This way of writing a vector is referred to as *bra*. Together they form the *bra(c)ket* $\langle\Psi|\Psi\rangle$. This denotes the inner product between two state vectors $|\Psi\rangle$ and $|\Phi\rangle$ given by $\langle\Psi|\Phi\rangle$.

It becomes clear that $n$ qubits can have a state space of dimension $\mathbb{C}^{2^n}$, which means the state space grows exponentially with the number of qubits.

### Superposition

As we already noticed, a qubit can be in a linear combination of classical states given by the basis vectors. We call such a linear combination of basis states a *superposition*. So

$$\alpha|0\rangle + \beta|1\rangle$$

is a superposition of $|0\rangle$ and $|1\rangle$. Intuitively, this means that a qubit can be both $|0\rangle$ and $|1\rangle$ at the same time! Not only single qubits can be a superposition of states.

An $n$-qubit state lives in a $d = 2^n$-dimensional vector space with orthonormal basis $\{|\psi_1\rangle, \ldots, |\psi_d\rangle\}$. We call any linear combination of basis states $|\psi_i\rangle$,

$$|\Psi\rangle = \sum_i \alpha_i |\psi_i\rangle,$$

a superposition of states with amplitude $\alpha_i$ for the state $|\psi_i\rangle$. For example the state

$$\frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

is a superposition of the states $|00\rangle$ and $|11\rangle$ with amplitude $1/\sqrt{2}$ for state $|00\rangle$, and amplitude $-1/\sqrt{2}$ for $|11\rangle$.

From the last example, we see that amplitudes can also be negative. Amplitudes only correspond to probabilities of measuring the corresponding state if we square them. In fact, these negative amplitudes are responsible for *interference* effects, which play an important role in quantum computing.

#### Phase

The phase of a state is an important characteristic of quantum mechanics. In general, we distinguish between global and relative phase. Let $|\Psi\rangle$ be any state vector and let $\theta$ denote a real number. We say that $e^{i\theta}|\Psi\rangle$ is equal to $|\Psi\rangle$ *up to a global phase factor*. Looking at how we define measurement below, we immediately see that this global phase cancels out during measurement. Since it has no influence on the measurement result, we say that a global phase has no observable consequence.

On the other hand, we call $e^{i\theta}$ in $\alpha|0\rangle + e^{i\theta}\beta|1\rangle$ a *relative phase*. Thus we can have a different phase associated with each amplitude. This makes a relative phase basis dependent. Unlike the global phase, a relative phase does have observable consequences, which will become important later on.

### 1.3.2 Measurement

As already indicated, we have the option to measure such a quantum state. What exactly does this mean? And in particular, if we could determine the amplitudes $\alpha$ and $\beta$ exactly couldn't a single qubit contain an infinite amount of information?

#### Measurements in the computational basis

Suppose we measure the single qubit state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in the computational basis. We cannot perceive the superposition as such, but will only observe outcomes $|0\rangle$ or $|1\rangle$. When measuring $|\Psi\rangle$ we obtain $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. We also refer to $\alpha$ and $\beta$ as *probability amplitudes.* Thus there are only two outcomes of the measurement and we cannot determine $\alpha$ and $\beta$ itself. Therefore the amount of information we can encode in a qubit is not infinite, as one may think initially. Generally if we speak of measurement, we refer to a measurement in the computational basis unless indicated otherwise.

In general, measuring the state $\sum_{x\in\{0,1\}^n} \alpha_x|x\rangle$ in the computational basis gives $|x\rangle$ with probability $|\alpha_x|^2$. Thus measurement of such a quantum state induces a probability distribution on the classical states $x \in \{0,1\}^n$.

An essential aspect of measurements in quantum mechanics is that they affect the state. Measuring $\alpha|0\rangle + \beta|1\rangle$ in the computational basis will *collapse* it to one of the basis states $|0\rangle$ and $|1\rangle$. This means that if we measure again immediately afterwards, we will obtain the same result: the superposition no longer exists.

### General measurements

In this text, we will need more refined measurements. In general, measurements in quantum mechanics are described by a collection of *measurement operators* $\{M_m\}$. The index $m$ refers to the measurement outcome. These operators can be described by matrices and act on the state space of the system we want to observe. If the system is in the state $|\Psi\rangle$ right before our measurement, then the probability of measuring outcome $m$ is given by

$$p(m) = \langle\Psi|M_m^\dagger M_m|\Psi\rangle.$$

After the measurement, the state will have collapsed to

$$\frac{M_m|\Psi\rangle}{\sqrt{p(m)}}.$$

The measurement operators should satisfy the *completeness relation* $\sum_m M_m^\dagger M_m = I$, where $I$ denotes the identity matrix. This reflects the requirement that the sum of all probabilities $p(m)$ should sum to 1, that is $\sum_m p(m) = 1$.

To get a feel for this formalism, consider the measurement of a single qubit in the computational basis. We can describe such a measurement by operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$. We have $M_0^\dagger M_0 + M_1^\dagger M_1 = M_0 + M_1 = I$. Measuring $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ now gives outcome $|0\rangle$ with probability

$$\begin{aligned} p(0) &= \langle\Psi|M_0^\dagger M_0|\Psi\rangle = \langle\Psi|M_0|\Psi\rangle = \\ &= (\alpha^*\langle 0| + \beta^*\langle 1|)|0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle) = |\alpha|^2\langle 0|0\rangle \\ &= |\alpha|^2 \end{aligned}$$

as expected. We also see that the state after the measurement with outcome 0 is given by

$$\frac{M_0|\Psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}|0\rangle.$$

We can ignore the global $\alpha/|\alpha|$, so the post-measurement state is indeed $|0\rangle$.

### Projective measurements

Projective measurements are a special case of general measurements, which we will make use of later. In contrast to general measurements, a projective measurement follows the additional requirement that the operators $M_m$ are orthogonal projectors. This means that all $M_m$ are Hermitian and $M_m M_m' = \delta_{m,m'} M_m$. Note that the operators $M_0$ and $M_1$ we used earlier to describe our measurement in the computational basis already satisfy this constraint, and can thus also be seen as a projective measurement.

In general, projective measurements are described by an *observable*, $M$, which is an Hermitian operator on the state space we wish to observe. This observable has a spectral decomposition $M = \sum_m m P_m$, where $P_m$ is a projector onto the eigenspace of $M$ with eigenvalue $m$. This may sound rather complicated, however, note that the possible measurement outcomes are simply the eigenvalues of $M$. The probability of observing $m$ is then given by

$$p(m) = \langle\Psi|P_m|\Psi\rangle$$

and the state right after the measurement is

$$\frac{P_m|\Psi\rangle}{\sqrt{p(m)}}.$$

For example we can choose $P_0 = |0\rangle\langle 0|$ and $P_1 = |1\rangle\langle 1|$ to perform a measurement in the computational basis.

An important fact, which will we need in a later chapter, is that it is possible to perform partial measurements on a state. Consider for example the state

$$|\Phi\rangle = \frac{1}{\sqrt{3}}|0\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) + \sqrt{\frac{2}{3}}|1\rangle \otimes |10\rangle.$$

We now want to measure only the first qubit, by applying measurement operators $P_0 = |0\rangle\langle 0| \otimes I^{\otimes 2}$ and $P_1 = |1\rangle\langle 1| \otimes I^{\otimes 2}$. The $\otimes I^{\otimes 2}$ just means we will do nothing to the last 2 qubits. Using the above definition this gives us outcome 0 with probability $p(0) = \langle\Phi|P_0|\Phi\rangle = 1/3$ and 1 with $p(1) = \langle\Phi|P_1|\Phi\rangle = 2/3$ respectively. If we measure 0, the state of the system collapses to

$$\frac{P_0|\Phi\rangle}{\sqrt{\frac{1}{3}}} = |0\rangle \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

However if we measure 1, the post-measurement state will be

$$\frac{P_1|\Phi\rangle}{\sqrt{\frac{2}{3}}} = |1\rangle \otimes |10\rangle.$$

### Distinguishing Quantum States

From these definitions it is also apparent that we can distinguish two quantum states which are orthogonal to each other, as we can then simply choose them as our measurement basis. How about non-orthogonal states? It turns out that if $|\Psi\rangle$ and $|\Phi\rangle$ are non-orthogonal, there is no quantum measurement capable of perfectly distinguishing these two states. To see informally why this is true consider some measurement operator $M_j$ with outcome $j$. Depending on the measurement outcome, we will decide whether the state was $|\Psi\rangle$ or $|\Phi\rangle$ according to a certain rule. For example, we use the rule that if we measure $j$, the state was $|\Psi\rangle$. Note that since the two states are non-orthogonal we can decompose $|\Phi\rangle = \gamma|\Psi\rangle + \eta|\phi\rangle$ into a non-zero component parallel to $|\Psi\rangle$, and a component $|\phi\rangle$, which is orthogonal to $|\Psi\rangle$. Due to this non-zero parallel component, there is a non-zero probability of obtaining measurement outcome $j$ when measuring $|\Phi\rangle$. But then we will erroneously conclude that the state was $|\Psi\rangle$. Thus we cannot distinguish two non-orthogonal states with perfect accuracy.

This is all we will need in this thesis concerning quantum measurements. More in depth information can be found in the book of Nielsen and Chuang [65][Section 2.2.3].

## 1.3.3 Evolution

Now that we looked at measurements, what else can we do with a quantum state? In particular, how could we perform any operations on a qubit before measuring?

Quantum mechanical systems undergo an evolution over time. We can describe the evolution of a closed quantum system by a *unitary transformation*. That is, the state $|\Psi\rangle = (\alpha_1, \ldots, \alpha_d)^T$ of the system at time $t_1$ is related to the state $|\Psi'\rangle = (\beta_1, \ldots, \beta_d)^T$ at time $t_2$ by a unitary operator $U$ such that

$$\begin{pmatrix} \beta_1 \\ \vdots \\ \beta_d \end{pmatrix} = U \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{pmatrix}.$$

$U$ can thus be seen as an $d \times d$ complex valued matrix. Since we want all states to stay normalized, we need $\sum_{i=1}^{d} |\beta_i|^2 = 1$. This condition is satisfied by requiring $U$ to be a unitary transformation. This just means that the inverse $U^{-1}$ of $U$ needs to be equal to its conjugate transpose $U^\dagger$. We thus see that unitary operations are norm preserving. Finally, we have $U^\dagger U = U^{-1}U = I$, thus if we apply $U^{-1}$ to $U|\Psi\rangle$, we move back to the original state. This means that quantum operations are *reversible*. Only measurement operations violate this principle, as we saw earlier. This again contrasts with classical computing. Consider for example the classical OR operation, which maps inputs $1, 0$ and $1, 1$ both to the same output state 1. Clearly this operation is not reversible.

How is this useful for quantum computing? We can perform operations on qubits, by bringing about a certain time evolution. Thus we operate on qubits by choosing a certain unitary operation $U$.

### 1.3.4 Entanglement

Before we can turn to construct quantum circuits, we need to look at what is perhaps the most intriguing property of quantum mechanics. As mentioned earlier, not every higher dimensional quantum system is a product state and can be decomposed into the product of several lower dimensional systems. We call states *entangled*, if they are not product states. Entangled states play a fundamental role in quantum computing. Consider for example the state

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

It is intuitively clear that we cannot decompose this state into two states $|\Psi\rangle$ and $|\Phi\rangle$. If we try to write down such a decomposition

$$\begin{aligned} |\Psi\rangle \otimes |\Phi\rangle &= (\alpha_1|0\rangle + \alpha_2|1\rangle) \otimes (\beta_1|0\rangle + \beta_2|1\rangle) \\ &= \alpha_1\beta_1|00\rangle + \alpha_1\beta_2|01\rangle + \alpha_2\beta_1|10\rangle + \alpha_2\beta_2|11\rangle \\ &= (|00\rangle + |11\rangle)/\sqrt{2} \end{aligned}$$

we see right away that we cannot pick $\alpha_1, \alpha_2, \beta_1, \beta_2$ such that $\alpha_1\beta_2 = \alpha_2\beta_1 = 0$ and $\alpha_1\beta_1 = \alpha_2\beta_2 = 1/\sqrt{2}$ simultaneously.

The state above is commonly referred to as an EPR pair, after Einstein, Podolsky and Rosen who tried to use it to prove Quantum Mechanics incomplete. It has some very peculiar properties. Suppose for example we take the first qubit of the state above and give it to Alice. Similarly, we take the second one and hand it to Bob. Alice now visits the newly established colony on the moon, taking her qubit along. At some point she measures her qubit in the computational basis, which will give her outcomes $|0\rangle$ or $|1\rangle$ with equal probability. However, since the two qubits are entangled, this will collapse the total state to either $|00\rangle$ or $|11\rangle$. This means that, if Bob measures his qubit right afterwards, his outcome is completely determined by Alice's measurement result. It is as if "information" has been transmitted *instantaneously* from Alice to Bob! Thus entanglement allows for *non-local* interactions. Note that the outcome is not fixed before the measurement unlike in the case of classical randomness where Alice and Bob share the outcome of a coin flip.

This intuitively strange property of entangled states makes them a fundamentally new resource which plays a central role in quantum computing. Especially in multiparty cryptographic protocols, entanglement can open up interesting possibilities, as we will see later.

### 1.3.5 Density Matrix Formalism

Above we have formulated quantum states as vectors. An alternative formulation, which we will need, is provided by the *density matrix*. This way of abstraction is sometimes more convenient.

First of all, we need two definitions. A quantum system described by a unit vector as a superposition of basis states is called a *pure state*. If this is not the case, we say that the system is in a *mixed state*. Suppose now that a quantum system is in a number of possible pure state $|\psi_i\rangle$, with respective probabilities $p_i$. The set $\{p_i, |\psi_i\rangle\}$, is called an *ensemble of pure states.* The *density operator* is then defined as

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|.$$

This is also called the *density matrix.* Measurements and unitary evolution can be rephrased in terms of the density operator. Since we will not require this in this text, we refer to the book by Nielsen and Chuang [65] for more information. We only note that a pure state will satisfy $\text{tr}(\rho^2) = 1$, whereas for a mixed state $\text{tr}(\rho^2) < 1$. tr is the trace operation, with $\text{tr}(|a_1\rangle\langle a_2|) = \langle a_1|a_2\rangle$. Furthermore it is important to realize that multiple different ensembles of quantum states may give rise to the same density matrix.

An important tool we will use is the so-called *reduced density matrix.* This becomes especially useful when analyzing multi-party protocols where different parties share part of an entangled state. The reduced density matrix lets us determine the state of a sub-system of a larger quantum system. Suppose we have two physical systems $A$ and $B$, whose common state is described by a density operator $\rho^{AB}$. The reduced density operator for $A$ is then defined as

$$\rho^A = \text{tr}_B(\rho^{AB}),$$

where $\text{tr}_B$ is called the partial trace over system $B$. This is a linear operator defined by

$$\text{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2|\text{tr}(|b_1\rangle\langle b_2|),$$

where $|a_1\rangle$, $|a_2\rangle$ are any two vectors in the state space of $A$ and $|b_1\rangle$,$|b_2\rangle$ any two vectors in the state space of $B$.

As an example, let's analyze the EPR pair shared by Alice and Bob above. This is a pure state, and we can calculate the density matrix $\rho$ as

$$\begin{aligned}\rho &= |\Psi\rangle\langle\Psi| = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|) \\ &= \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|)\end{aligned}$$

If Alice has only the first qubit in her possession, how can we describe her state?

We can *trace out* Bob's part to obtain

$$
\begin{aligned}
\rho_A &= tr_B(\rho) \\
&= \frac{1}{2}(tr_B(|00\rangle\langle 00|) + tr_B(|00\rangle\langle 11|) + tr_B(|11\rangle\langle 00|) + tr_B(|11\rangle\langle 11|)) \\
&= \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|) \\
&= \frac{I}{2}
\end{aligned}
$$

We see that Alice's state is a mixed state, since $\mathrm{tr}((I/2)^2) = 1/2 < 1$. Intuitively this means that this is a state that is not known completely to Alice. Observe that starting out with $(|00\rangle - |11\rangle)/\sqrt{2}$ instead, would give the same reduced density matrix for Alice. We can conclude from this, that without any additional information Alice cannot distinguish these two possibilities given her qubit alone.

## 1.4 Quantum Computation

Similarly to how a classical computer can be modeled as a circuit consisting of wires and classical logic gates [67], a quantum computer can be described as a *quantum circuit* with the help of *quantum gates*. Note that if we talk about wires, we refer to a connection within the model, not to the physical implementation. Just as classical bits are not only represented by signals traveling over electrical wires, we can have qubits implemented as photons move from one location to another through space. Sometimes wires here may just denote the passage of time.

### 1.4.1 Elementary Quantum Gates

Now how can we describe such quantum gates? Since the evolution of a quantum state is given by a unitary transformation, it is natural to describe quantum gates this way. If we talk about a quantum gate, we thus simply refer to a unitary transformation applied to a number of qubits. In the following, we will refer to an input of a circuit as a *register*.

**Single Qubit Gates**

We first consider gates that act on a single qubit. Some of the most important quantum gates are given by the Pauli matrices

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}; Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We refer to $X$ as a *bit flip*, since it maps

$$\begin{aligned} X|0\rangle &= X\begin{pmatrix} 1 \\ 0 \end{pmatrix} = |1\rangle, \\ X|1\rangle &= X\begin{pmatrix} 0 \\ 1 \end{pmatrix} = |0\rangle. \end{aligned}$$

The gate described by $Z$ is called a *phase flip*, since it changes the basis states to

$$\begin{aligned} Z|0\rangle &= |0\rangle \\ Z|1\rangle &= -|1\rangle \end{aligned}$$

We can write $Y = iXZ$ and observe that

$$\begin{aligned} Y|0\rangle &= i|1\rangle, \\ Y|1\rangle &= -i|0\rangle. \end{aligned}$$

One of the most important 1-qubit gates which we will make use of is the Hadamard gate, described by

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Note that we could equally well express this transformation as $H = (X + Z)/\sqrt{2}$. Using the vector representation for qubits we see that the effect of $H$ on the basis states $|0\rangle$ and $|1\rangle$ is given by

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \end{aligned}$$

As we saw earlier, all such unitary transformations are reversible. In particular this means that also all quantum gates must be reversible. So if we apply $H^{-1}$ to the states on the left hand side, we should get back to the original basis states. Noting that $H^{-1} = H$ we indeed obtain

$$\begin{aligned} H\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) &= |0\rangle, \\ H\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) &= |1\rangle. \end{aligned}$$

Looking at the last line, we see that the positive and negative amplitudes of $|0\rangle$ cancel out to give $|1\rangle$, which is an effect known as *interference.* This is analogous to interference of for example light waves, and plays an important role in quantum computing.

If we want to apply two quantum gates $U$ and $V$ on a single qubit sequentially, we denote the combined quantum gate using the *matrix product* of $U$ and $V$. Thus if we first let $V$ act on $|0\rangle$ followed by $U$, the resulting state will be $UV|0\rangle$. To describe the actions of an entire circuit however, we will also need to combine operations acting on different qubits. Consider for example the state $|00\rangle$. We now want $U$ to act on the first qubit, and $V$ on the second one. The effect of the circuit on the state $|00\rangle$ can now be described by the *tensor product* $U \otimes V$ defined in Appendix A. This gives us output $(U \otimes V)|00\rangle = U|0\rangle \otimes V|0\rangle$.

For example, consider a circuit with $n$ inputs, where we apply a Hadamard transform to each input. The effect of this circuit is described by

$$\underbrace{H \otimes H \otimes \ldots \otimes H}_{n\ H\text{'s}} = H^{\otimes n}$$

Suppose we have as input $n$ zeros $|0^n\rangle$. Then the result of this circuit will be $H^{\otimes n}|0^n\rangle = (\sum_{j \in \{0,1\}^n} |j\rangle)/\sqrt{2^n}$, which is the uniform superposition of all possible strings of length $n$. What happens if we use an arbitrary input $|z\rangle$ of $n$ qubits? For this we obtain

$$H^{\otimes n}|z\rangle = \frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} (-1)^{z \cdot j} |j\rangle$$

where $z \cdot j = \sum_{k=1}^{n} z_k j_k$ denotes the inner product between the $n$-bit strings $z$ and $j$. This $n$-fold transformation will turn out to be very useful in quantum algorithms.

### Rotations

A visualization which is very effective in understanding single qubit gates, is to consider the representation of a qubit $\alpha|0\rangle + \beta|1\rangle$ as a point on the unit sphere. To make this work, express the qubit as $e^{i\gamma}(\cos(\theta/2)|0\rangle + e^{i\psi}\sin(\theta/2))|1\rangle$ where $\gamma$, $\theta$ and $\psi$ are real numbers. As we mentioned earlier, we can ignore the global phase, since it has no observable effects. Note that this preserves normalization since $|\alpha|^2 + |\beta|^2 = \cos^2(\theta/2) + \sin^2(\theta/2) = 1$. This is called the *Bloch sphere representation* (Figure 1.1) and a qubit forms a *Bloch vector* $(\cos\psi\sin\theta, \sin\psi\sin\theta, \cos\theta)$.

The usefulness of this representation becomes immediately apparent when we consider the effects of the Hadamard transform on a qubit. Note that $(|0\rangle + |1\rangle)/\sqrt{2}$ can be found in the figure at the intersection of the positive $x$-axis and the sphere. It is then easy to see that we can describe the effect of $H$ on $(|0\rangle + |1\rangle)/\sqrt{2}$ as a rotation around the $y$ axis towards $|1\rangle$, followed by a reflection in the $x$-$y$ plane.

In fact, the Bloch sphere representation allows one to view all single qubit operations as rotations on this sphere. We write $R_s(\theta)$ as the rotation around the axis $s \in \{x, y, z\}$ by the angle $\theta$. The basic rotations around the $x$,$y$ and $z$ axis can be expressed using the Pauli matrices and are given by $R_x(\theta) = e^{-i\theta X/2}$, $R_y(\theta) = e^{-i\theta Y/2}$

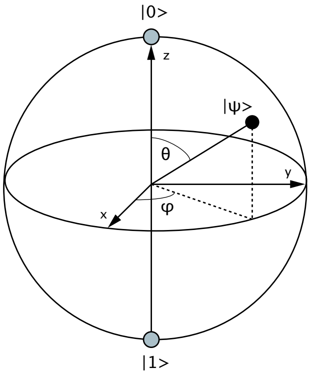

Figure 1.1: Bloch Sphere

and $R_z(\theta) = e^{-i\theta Z/2}$. Especially important for this text will be the rotation around the $z$ axis. We can express it in more detail as

$$R_z(\theta) = e^{-i\theta Z/2} = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} = e^{-i\theta/2} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

Any arbitrary single qubit operation $U$ can be expressed in terms of these rotations as

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta)$$

for some real numbers $\alpha$,$\beta$,$\gamma$ and $\delta$ [65, Theorem 4.1].

### Multi-qubit Gates

So far we have only considered gates acting on a single qubit. We now examine the so-called CNOT (controlled not) gate. It turns out that we can construct any quantum circuit using only single qubit gates and CNOT. The CNOT gate acts on 2 qubits, and will negate the value of the second qubit, if the first qubit is 1. More formally we have

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

which maps

$$\begin{aligned} \text{CNOT}|0\rangle|b\rangle &= |0\rangle|b\rangle \\ \text{CNOT}|1\rangle|b\rangle &= |1\rangle|1-b\rangle \end{aligned}$$

### 1.4.2 Copying Qubits

In classical computing we can easily make a copy of a bit. Can we do the same using quantum gates? It turns out that is impossible to copy a arbitrary qubit! This forms another fundamental difference between classical and quantum computing.

We can verify this intriguing fact by looking at our definition of a quantum gate above. Suppose there is indeed some gate described by a unitary transform $U$ which can copy a pure quantum state $|\psi\rangle$ to another register $|s\rangle$. Thus the effect of $U$ will be $U(|\psi\rangle|s\rangle) = |\psi\rangle|\psi\rangle$. Suppose now that this copying procedure works for two arbitrary pure states $|\psi\rangle$ and $|\phi\rangle$:

$$\begin{aligned} U(|\psi\rangle|s\rangle) &= |\psi\rangle|\psi\rangle \\ U(|\phi\rangle|s\rangle) &= |\phi\rangle|\phi\rangle \end{aligned}$$

If we now take the inner product of these two equations, we obtain $\langle\psi|\phi\rangle = (\langle\psi|\phi\rangle)^2$, since $U^\dagger U = I$. This is only possible if $\langle\psi|\phi\rangle = 0$ or $|\psi\rangle = |\phi\rangle$. Thus either the two states are the same, or they are orthogonal to each other. It is therefore impossible to clone quantum states which are not orthogonal to each other, as for example $|0\rangle$ and $(|0\rangle + |1\rangle)/\sqrt{2}$. This reflects the fact that only orthogonal states can be distinguished from each other with perfect accuracy.

### 1.4.3 Quantum Queries

In this text we will also require the notion of a *quantum query*. Just as in the classical case we can informally specify a query as a question we will ask to a blackbox, which will give us a certain answer in return.

Later on we consider queries with $\ell$-bit answers, where $\ell \geq 1$. For $\Sigma = \{0,1\}^\ell$, a quantum query to a string $y \in \Sigma^m$ is the unitary transformation specified by

$$|j\rangle|z\rangle \mapsto |j\rangle|z \oplus y_j\rangle$$

where $j \in [m]$, $z \in \{0,1\}^\ell$ is called the target register, and $z \oplus y_j$ is the string resulting from the xor of the individual bits of $z$ and $y_j$, i.e. $z \oplus y_j = (z_1 \oplus y_{j,1}) \ldots (z_\ell \oplus y_{j,\ell})$. It is sometimes convenient to get the query result in the phase. To achieve this, define

$$|z_T\rangle = \frac{1}{\sqrt{2^\ell}} \bigotimes_{i=1}^{\ell} (|0\rangle + (-1)^{T_i}|1\rangle)$$

where $T_i$ is the $i$th bit of the $\ell$-bit string $T$. Since $|0 \oplus y_{j,i}\rangle + (-1)^{T_i}|1 \oplus y_{j,i}\rangle = (-1)^{T_i \cdot y_{j,i}}(|0\rangle + (-1)^{T_i}|1\rangle)$, a query maps

$$|j\rangle|z_T\rangle \mapsto |j\rangle(-1)^{T \cdot y_j}|z_T\rangle.$$

### 1.4.4 Example: Quantum Teleportation

As an example of a quantum circuit, we will look at quantum teleportation, since this will become useful for quantum anonymous transmissions later on. This circuit also provides insight into how extremely useful entangled states can be. Other well known circuits implement Shor's factoring algorithm [74] and Grover's search algorithm [48], which finds an element in an unsorted list of size $N$ in time proportional to $\sqrt{N}$. Since we do not make use of these circuits in this text, we refer to the book by Nielsen and Chuang [65] for an excellent introduction to these algorithms.

For quantum teleportation, we will make use of the EPR pair $|E\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ which we encountered earlier. Suppose Alice has the first qubit of this state, and Bob the second. Alice can now use their shared entanglement to send one qubit $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to Bob by transmitting only 2 bits of classical information. Consider the following quantum circuit:

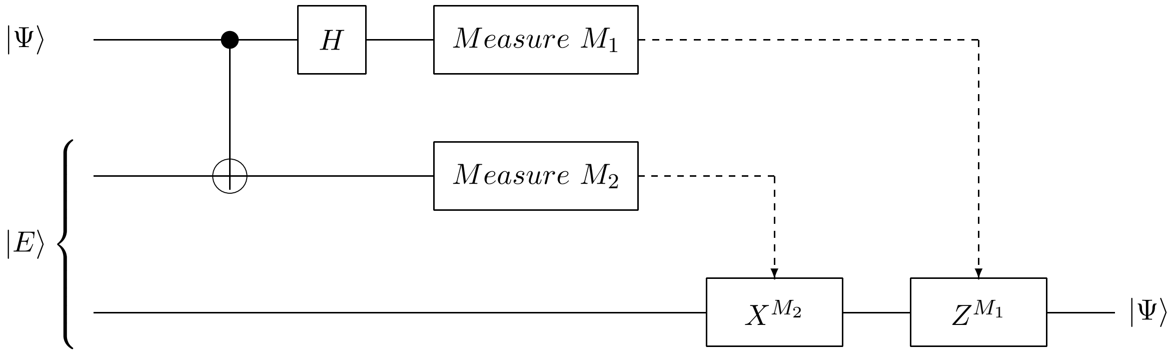

Figure 1.2: Quantum Teleportation Circuit

The input of this circuit is

$$|\Psi\rangle|E\rangle = \frac{1}{\sqrt{2}}\left[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)\right]$$

In Figure 1.2, Alice controls the first two wires and Bob the last one. This corresponds to Alice having the state $|\Psi\rangle$ and the first qubit of the EPR pair $|E\rangle$. Alice now sends her two qubits through a CNOT gate, which changes the state to

$$\frac{1}{\sqrt{2}}\left[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)\right].$$

She then applies a Hadamard transform to the first qubit, giving

$$\begin{aligned} & \frac{1}{2}\left[\alpha(|0\rangle+|1\rangle)(|00\rangle+|11\rangle)+\beta(|0\rangle-|1\rangle)(|10\rangle+|01\rangle)\right]= \\ = \; & \frac{1}{2}\left[|00\rangle(\alpha|0\rangle+\beta|1\rangle)+|01\rangle(\alpha|1\rangle+\beta|0\rangle)+\right. \\ & \left.+|10\rangle(\alpha|0\rangle-\beta|1\rangle)+|11\rangle(\alpha|1\rangle-\beta|0\rangle)\right] \end{aligned}$$

Alice now measures her two qubits, which will collapse the state to one of the four terms in the sum. For example if Alice's measurement result is 00, we can easily see from the equation above that Bob's state is now $|\Psi\rangle$. But this is exactly what Alice was trying to send to Bob! In general we can determine Bob's state, given Alice's measurement outcome:

$$\begin{aligned} 00 &\mapsto I(\alpha|0\rangle+\beta|1\rangle)=|\Psi\rangle \\ 01 &\mapsto X(\alpha|1\rangle+\beta|0\rangle)=|\Psi\rangle \\ 10 &\mapsto Z(\alpha|0\rangle-\beta|1\rangle)=|\Psi\rangle \\ 11 &\mapsto ZX(\alpha|1\rangle-\beta|0\rangle)=|\Psi\rangle \end{aligned}$$

Alice now transmits her measurement outcome $M_1M_2$ to Bob. This allows him to apply the appropriate transform to recover the original state $|\Psi\rangle$, which is given by $Z^{M_1}X^{M_2}$. Note that since Alice still needs to transfer two classical bits before Bob has any information about $|\Psi\rangle$, quantum teleportation does not allow faster than light communication.

### 1.4.5 Summary

We have examined the essential building blocks of quantum computing, which differs fundamentally from classical computing. The state of a quantum computer is described by a quantum state, which can be a linear combination of several classical states. We call such a linear combination a superposition. The dimensionality of the state space grows exponentially with the number of qubits. Operations on quantum states can be described by unitary operators. All such operations are reversible. Only a measurement may irrevocably alter the quantum state. Unlike in classical computing, we cannot copy an arbitrary qubit. Finally, quantum states can be entangled. Such entangled states have a number of interesting properties and form an important resource for quantum computation.

# Chapter 2

# Cryptographic Terminology

Analogous to Chapter 1 stating the basic notions of quantum computing, we will briefly examine the cryptographic notions used in this text. A good in-depth overview can be found in [62].

## 2.1 Terminology

In this text we are concerned with multi-party protocols, where the participants can communicate by sending messages. In this context, we will use the following terminology:

- An *entity*, *party* or *player* is someone or something that sends, receives and manipulates information. You and your computer, for example, can both be entities in a protocol.
- A *sender* is an entity that is the legitimate transmitter of information.
- A *receiver* or *recipient* is an entity that is the legitimate recipient of information.

Another notion we will make use of is that of a *channel*, which allows one entity to transmit information to another. We will also speak of a *secure channel* to denote a channel whose contents are not accessible to an adversary.

An *adversary* or *attacker* is an entity that tries to defeat the security provided by the protocol. Suppose, for example, that the sender tries to transmit an encrypted message to the recipient. An adversary defeats the security of the encryption protocol if he can read the transmitted message. We distinguish two classes of attackers:

- A *passive attacker* is restricted to observing the channel. That is he is allowed to read messages passing over the network, but he cannot make any modifications.
- An *active attacker* can observe the channel and also add, delete or modify any messages that pass over the channel.

Note that in many protocols an adversary can simultaneously play the role of either sender or receiver. In particular we say that

- A participant is *honest* or *correct*, if he follows the protocol.
- He is *dishonest*, *malicious* or *corrupted*, if he does not follow the protocol.

If multiple participants work together to defeat the security of the protocol we speak of a *collusion* of participants.

## 2.2 Security Models

Several different models exists to evaluate the security of cryptographic protocols.

### 2.2.1 Information Theoretic Security

The first model is that of *information-theoretic security.* We also refer to this as *unconditional security* or, in the context of encryption systems, *perfect secrecy.* Unconditional security sounds great, but what exactly does it mean? A protocol that is unconditionally secure, is secure even if an attacker is granted unlimited computational resources. We happily provide him with the most powerful computer on earth and as much memory space as he wants. The main question of unconditional security is thus whether the attacker obtains enough information to defeat the security of the system. We can express this notion using the concept of entropy from information theory [73]. The entropy of a random variable $X$ is defined as

$$H(X) = -\sum_{x} p(X = x) \log p(X = x)$$

where $p(X = x)$ is the probability that $X$ is in the state $x$. Entropy is a measure of information and uncertainty about $X$. In our setting for example, let $E$ denote the messages exchanged by a cryptographic protocol and let $M$ be the item the protocol could be concealing. For each item $m$ in the set of all possible items, there is a probability $p(M = m)$ that $M = m$ is the real item. For example, $E$ can be an encrypted message and $m$ the unencrypted message contents. If we are certain that $M = m$ is the true message, we have $p(m) = 1$ and our uncertainty is minimal: $H(M) = 0$. If, on the other hand, we have no idea which message could possibly be hidden in $E$ and we thus consider all possible decryptions to be equally likely, our uncertainty $H(M)$ is maximized. $H(M|E)$ denotes the uncertainty about $M$ given $E$. More formally

$$H(M|E) = -\sum_{m}\sum_{e} p(M = m, E = e) \log p(M = m|E = e).$$

The mutual information

$$I(M, E) = H(M) - H(M|E)$$

then represents how much information an adversary can gain about the true $M$ by collecting the message exchange $E$. A system is unconditionally secure if the knowledge of $E$ does not decrease the uncertainty about $M$, that is $H(M|E) = H(M)$. For unconditionally secure protocols we thus have $I(M,E) = H(M) - H(M) = 0$. Intuitively this means that even if the attacker collects all messages during the execution of the protocol, he still cannot learn anything more about $M$!

Information theoretic security is clearly desirable and corresponds to the intuitive notion of "secure". Why would we even want to consider other models? Unfortunately, information-theoretic security is hard to achieve in practice. For example, it is easy to show that for message encryption, the key used to encrypt the message must be as long as the message itself to achieve perfect secrecy. Furthermore, once such a key has been used, we are never allowed to use it again. Clearly this poses some practical problems as to how we should distribute such keys. It has even been shown that using only classical communication it is impossible for two remote parties to establish such a key from scratch [20].

### 2.2.2 Computational Security

Most forms of practical cryptography are therefore based on what is called *computational security*. In this security model, we do not grant an adversary unlimited computational resources. Instead, we are concerned with the amount of computation required to break the security of a system. We say that a system is *computationally secure*, if the perceived level of computation necessary to defeat it exceeds the computational resources of any hypothetical adversary by a comfortable margin. The adversary is thereby allowed to use the best known attacks against the system. Closely related are the concepts of complexity-theoretic security and provable security. In this text, we will use the term computational security to encompass both notions. In complexity-theoretic security, the adversary is modeled as having only polynomial computational power. This means that any attacks involve time and space polynomial in the size of the underlying security parameters of the system. In the setting of provable security, the difficulty of defeating the system's security is proven to be as difficult as solving a well-known problem which is thought to be hard. Note that this does not prove the protocol to be unconditionally secure, but only makes a statement of equivalence between the security of the protocol and a hard to compute problem. In practice, these are often number-theoretic problems such as factoring. Note that for example in the case of factoring, it is not known whether these problems are truly difficult to solve classically. In quantum computing, factoring is not a hard assumption any more. Thus "provable" just means the security is proven given certain hardness assumptions.

## 2.3 Resources

In general, we here speak of *randomness* to denote a string $s$ generated by the outcomes of a fair coin flip. We say that a participant $P$ "knows" $M$, if he has no uncertainty about $M$ or, more formally, $H(M|P) = 0$. A participant "does not know" $M$, if his uncertainty about $M$ is maximized. In this context we will use the following terminology:

- We speak of *private randomness*, if a participant holds randomness that is not known to any of the other participants of the protocol.
- We speak of *shared randomness*, if two or more participants know the same randomness.
- We also talk about *private shared randomness*, if two or more participants have shared randomness which is not known to the remaining participants. We also call this a *key*. If only two participants share such a key, we also speak of *pairwise private shared randomness*. Unless otherwise indicated, in this text we always mean pairwise private shared randomness when we speak of private shared randomness.

## 2.4 Summary

We have described the basic cryptographic notions and terminology necessary for the understanding of this text. In particular, we have examined the difference between information-theoretic and computational security. We also introduced the different types of randomness, which we will refer to frequently.

# Part I

# Private Information Retrieval and Locally Decodable Codes

# Chapter 3

# Private Information Retrieval

## 3.1 Introduction

Private information retrieval enables a user to retrieve an entry from a database, while hiding the index of the requested entry. One straightforward solution for this problem would be to send the entire database to the user, who can then simply select the desired item. Whereas this ensures that the database can never learn the index of the requested entry, it is clearly not very efficient. Can we do with less communication? The main question of private information retrieval is thus its communication complexity.

There are numerous practical applications where PIR could play an important role. Consider for example a database containing medical information. To protect the privacy of the patients, it is desirable for the database to remain ignorant about their inquiries. Another practical application could be a database containing patent information. Imagine for example a scientist who just made a great invention, say "round wheels are best", and wants to patent it. For this he will consult the database to check whether there are any existing patents covering his invention. However, asking the database for "round wheels" already gives away part of his brilliant idea. A malicious database could now apply for a patent itself. And even if his query does not give away any vital information, the database will at least learn about the area of research and perhaps start its own investigations. Using private information retrieval, the user is able to retrieve existing patents from the database, without letting it know which ones. A similar application is a database containing pharmaceutical information used for research or databases consulted by stock traders. If the database for example learns which particular stocks a certain trader is interested in, it may decide to buy or sell itself. Private information retrieval also plays a role in anonymous publishing where it is referred to as query anonymity [39]. There it can be used in combination with anonymous transmissions which we will consider in a later chapter. Several other applications are given by Asonov [8].

A similar problem was first considered in a complexity-theoretic context under the name of instance hiding by Abadi, Feigenbaum and Killian [1] and later Beaver et al. [14]. Private information retrieval was then introduced in 1995 by Chor, Goldreich, Kushilevitz, and Sudan [33], and has received considerable attention since. In the PIR model we view the database as an $n$-bit string $x = x_1, \ldots, x_n \in \{0,1\}^n$, where the user wants to retrieve the $i$-th bit $x_i$. The database is not allowed to learn anything about $i$, while allowing the user to compute $x_i$ from its reply. Note that in particular, this means that the database is also not allowed to learn which items the user is *not* interested in. For example, it may not learn that $i \neq 3$. How do we fit real world databases into this model, after all a single database entry may be longer than 1 bit? In this case, we can query the entry of length $z$ bit by bit: we query $z$ individual $n$-bit databases.

Clearly just transmitting the entire database as suggested earlier would be the simplest approach. This, however, is infeasible in practice. It has been shown that achieving communication of less than $n$ bits with a single server is impossible when information-theoretic security is required [33]. This is also the case for quantum PIR [53]. Interestingly however, it is possible to reduce the amount of communication in this setting, if we allow the database to be replicated over multiple servers each holding a copy of $x$. Relaxing the privacy constraint to computational security, it is possible to achieve sub-linear communication using only a single server.

**Outline**

We first take a look at information-theoretic PIR and examine a number of existing protocols for this problem. Known lower bounds for this problem are stated, which we will improve upon in Chapter 5. We then briefly review the case of computationally secure PIR. In this thesis, we are especially interested in the case of information-theoretic PIR, since most known computationally secure PIR schemes can easily be broken once a quantum computer is built. We give known upper and lower bounds for this problem and review several PIR schemes which illustrate how multiple servers can be used to reduce the amount of communication.

## 3.2 Information-Theoretic PIR

### 3.2.1 Definition

Formally we can define information-theoretic PIR as follows:

**Definition 3.2.1** *A one-round, $(1-\eta)$-secure, $k$-server* private information retrieval (PIR) *scheme for a database $x \in \{0,1\}^n$ with recovery probability $1/2 + \varepsilon$, query size $t$, and answer size $\ell$, consists of a randomized algorithm (user) and $k$ deterministic algorithms $S_1, \ldots, S_k$ (servers), such that*

1. *On input $i \in [n]$, the user produces $k$ $t$-bit queries $q_1, \ldots, q_k$ and sends these to the respective servers. The $j$th server sends back an $\ell$-bit string $a_j = S_j(x, q_j)$. The user outputs a bit $f(a_1, \ldots, a_k)$ where $f$ depends on $i$ and his randomness.*

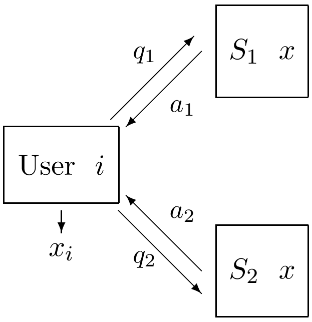

Figure 3.1: 2-server Private Information Retrieval

2. *For every* $x \in \{0,1\}^n$ *and* $i \in [n]$ *we have* $\Pr[f(a_1, \ldots, a_k) = x_i] \geq 1/2 + \varepsilon$.

3. *For all* $x \in \{0,1\}^n$, $j \in [k]$, *and any two indices* $i_1, i_2 \in [n]$, *the two distributions on* $q_j$ *(over the user's randomness) induced by* $i_1$ *and* $i_2$ *are* $\eta$*-close in total variation distance.*

*We say that the scheme* uses $b$ bits, *if the user only uses* $b$ *predetermined bits from each query answer of length* $\ell$*: he outputs* $f(a_{1|S_1}, \ldots, a_{k|S_k})$ *where the sets* $S_1, \ldots, S_k$ *are of size* $b$ *each and are determined by* $i$ *and the user's randomness.*

*The scheme is called* linear, *if for every* $j$ *and* $q_j$ *the* $j$*th server's answer* $S_j(x, q_j)$ *is a linear combination (over* $GF(2)$*) of the bits of* $x$.

The setting $\eta = 0$ corresponds to the case where the server gets no information at all about $i$. All known non-trivial PIR schemes have $\eta = 0$, perfect recovery ($\varepsilon = 1/2$), and only one round of communication. Servers are not allowed to communicate. We furthermore assume a secure channel between the user and the servers, i.e. a server cannot monitor transmissions to and from another server.

Using $k \geq 2$ non-communicating servers allows for PIR with less than $n$ bits of communication. Each of the $k$ servers has a copy of the $n$-bit database $x$. The individual server should learn nothing about $i$, even if it has unlimited computational resources. Since the $k$ servers are not allowed to communicate with each other, this gives information-theoretic privacy for the user. To retrieve an item from the database, the user is allowed to send a query $q_j$ to database $j$, which will send back an answer $a_j$. The user now selects $b$ bits of each answer and combines them to compute the value of $x_i$. We can visualize this for the 2-server case in Figure 3.1.

### 3.2.2 Example Protocols

To illustrate how multiple servers can be used to reduce the communication complexity of PIR, we take a look at some of the known PIR schemes [33].

### Square scheme

We first examine a very simple 2-server PIR scheme having $\eta = 0$ and $\varepsilon = 1/2$. The database $x = x_1 \ldots x_n$ is arranged in a square:

$$x = \begin{pmatrix} x_1 & x_2 & & \cdots & x_{\sqrt{n}} \\ x_{\sqrt{n}+1} & \ddots & & & x_{2\sqrt{n}} \\ \vdots & & x_i & & \vdots \\ \vdots & \cdots & \cdots & \cdots & x_n \end{pmatrix}$$

The index $i$ can now be described by two coordinates $(i_1, i_2)$. The user picks a random string $A \in \{0,1\}^{\sqrt{n}}$, and sends $\sqrt{n}$-bit queries $q_1 = A$ and $q_2 = A \oplus e_{i_1}$ to the two servers, respectively. The first server returns the $\sqrt{n}$-bit answer $a_1 = q_1 \cdot C_1, \ldots, q_1 \cdot C_{\sqrt{n}}$, where $q_1 \cdot C_c$ denotes the inner product mod 2 of $q_1$ with the $c$th column of $x$. The second server sends $a_2$ analogously. The user selects the bit $q_1 \cdot C_{i_2}$ from $a_1$ and $q_2 \cdot C_{i_2}$ from $a_2$ and xors these two bits to get $(A \cdot C_{i_2}) \oplus ((A \oplus e_{i_1}) \cdot C_{i_2}) = e_{i_1} \cdot C_{i_2} = x_i$. This scheme has query and answer length $t = \ell = \sqrt{n}$ and uses $b = 1$ bits from each answer.

### Combinatorial Cubes

Instead of arranging $x$ in a square, we can also associate it with a $d$-dimensional cube $[c]^d$ [33]. For this protocol, we require $k = 2^d$ servers initially. The number of servers can later be reduced using a method based on covering codes.

Assume without loss of generality that $n = c^d$, which allows us to arrange $x$ in a $d$-dimensional cube where each index $i \in [n]$ can now be described by coordinates $(i_1, \ldots, i_d) \in [c]^d$. For reasons that will become apparent shortly, it is convenient to describe all $2^d$ servers by their binary index in $\{0,1\}^d$. Let $e_{i_j}$ denote the $[c]$-bit string corresponding to the singleton set $S = \{i_j\}$

**Protocol 1: PIR Combinatorial Cubes**

**1:** The user $U$ uniformly and independently picks $d$ strings $S_1^0, S_2^0, \ldots, S_d^0 \subseteq [c]$.

**2:** $U$ now constructs $d$ strings in $[c]$ by setting $S_j^1 = S_j^0 \oplus e_{i_j}$ for each $j \in [d]$. $U$ now has $d$ pairs of strings $(S_1^0, S_1^1), \ldots, (S_d^0, S_d^1)$.

**3:** $U$ sends a single string of each pair to each of the $2^d$ servers: To server with id $D = \sigma_1, \ldots, \sigma_d \in \{0,1\}^d$ he sends query $q_D = S_1^{\sigma_1}, \ldots, S_d^{\sigma_d}$.

**4:** Server $D$ now computes the xor of all the bits in the subcube defined by $S_1^{\sigma_1}, \ldots, S_d^{\sigma_d}$,

$$a_D = \bigoplus_{j_1 \in S_1^{\sigma_1}, \ldots, j_d \in S_d^{\sigma_d}} x_{j_1, \ldots, j_d},$$

and sends $a_D$ back to the $U$.

**5:** $U$ now computes: $x_i = \bigoplus_{D \in \{0,1\}^d} (a_D)$

Comparing the different subcubes, it is easy to see that only index $(i_1, \ldots, i_d)$ occurs in exactly one subcube. All other indices appear in an even number of subcubes. Thus the bits corresponding to those indices cancel in the sum and what is left is just $x_{i_1,\ldots,i_d} = x_i$. The privacy of this protocol follows from the fact that the user chooses $d$ $c$-bit strings completely at random. The difference between the string $S_j^0$ and $S_j^1$ is exactly one bit, corresponding to the index $i_j$. Each server, however, receives only one of these strings and not both. Thus the server simply sees a uniformly random $(d \cdot c)$-bit string as a query, from which he cannot infer anything.

To each of the $k = 2^d$ servers we send $d$ $c$-bit strings, giving a query length of $t = d \cdot c$. The length of the answers is $\ell = 1$. Thus the total amount of communication is given by $k(t + \ell) = 2^d(d \cdot c + 1) = 2^d(dn^{1/d} + 1)$. Using the fact that $k = 2^d$ this construction uses $\Omega(k \log(k) n^{1/\log k} + k)$ bits of communication.

The number of servers can be reduced to $k < 2^d$ using a method derived from covering codes. Note that since the difference between two strings $S_j^0$ and $S_j^1$ is only one bit, a server with id $D$ can effectively simulate any other server whose binary id $D'$ has Hamming distance $d(D, D') = 1$. Server $D$ can simply flip each successive bit in each $S_j^{\sigma_j^D}$ where $\sigma_j^D \neq \sigma_j^{D'}$. For example, for a cube of dimension $d = 3$, server $D = 000$ who received query $S_1^0, S_2^0, S_3^0$, can emulate server 100 by flipping each bit $m$ in $S_1^0$ and computing the xor of of all bits in that subcube. It then sends each of the $c$ extra bits back to the user, who can then select the bit of the reply corresponding to $m = i_1$. Likewise server 000 can simulate servers 010 and 001, and server 111 can simulate servers 011,101 and 110. We can therefore reduce the number of servers from $2^3$ to 2. In addition to simulating 3 other servers, each server computes the answer corresponding to the original query and xors it with each bit in the query reply. The length of the answers is now $\ell = 3n^{1/3}$. The user selects the $b = 3$ bits from each answer corresponding to the answers he would have received from the simulated servers. Generally Chor et al. [33] show that this method can be used to obtain a $k$-server PIR with communication complexity $k + (2^d + (d-1) \cdot k) \cdot n^{1/d}$ using a $k$-word covering code for $\{0,1\}^d$. A covering code, $C_d$, with radius one for $\{0,1\}^d$ is a collection of codewords $C_d = \{c_1, c_2, \ldots, c_k\} \subseteq \{0,1\}^d$, such that all words at Hamming distance one from the codewords cover the entire space $\{0,1\}^d$. Let $B(s,1)$ be the set of all $d$-bit strings at Hamming distance one from $s$. Then $\{0,1\}^d \subseteq \cup_{c_j \in C_d} B(c_j, 1)$. As noted earlier, a server can simulate all servers whose id is at Hamming distance one from its own server id. Using only $k$ codewords $c_j$ from a covering code as $k$ server ids allows all remaining servers to be simulated. In the example above, we used the covering code $C_3 = \{(0,0,0),(1,1,1)\}$ of $\{0,1\}^3$ with radius one.

The only remaining question is, how big does $k$ have to be in order to cover the entire space $\{0,1\}^d$? It has been shown that $k \leq 2^d/(d+1)$ for most cases [33]. This shows that the simulation method is most interesting, if $d$ is small. The amount of communication is then in $O(k \log k n^{1/(\log k + \log\log k)})$ [33], [42].

### 3.2.3 Upper Bounds

In their original paper [33], Chor et al. suggested a scheme using $O(k \log k \; n^{1/\log k})$ bits of communication, which we examined in Section 3.2.2. Using covering codes, this can be extended to $O(k \log k \; n^{1/(\log k + \log\log k)})$ [33]. They also give an $O(\log n)$-server PIR with communication complexity $O(\log^2 n \log\log n)$. Finally, their scheme based on polynomials uses $O(k^2 \log(k) n^{1/k})$ bits of communication [42]. Ambainis [4], however, constructed a much more efficient scheme for $k > 2$ servers based on the 2-server scheme from [33] using recursion. A $(k+1)$-server scheme is built from an existing $k$-server scheme. This construction gives a protocol with $O(2^{k^2} n^{1/(2k-1)})$ bits of communication. Itoh [51] gives a slightly different PIR scheme with complexity $O(k! n^{1/(2k-1)})$. Yet more efficient schemes based on the idea of representing the database by polynomials, result in a scheme with complexity $O(k^3 n^{1/(2k-1)})$. Protocols using this representation are based on a generalization of a multi-party communication protocol by Babai, Kimmel, and Lokam [12] in the simultaneous message model. Beimel, Ishai and Kushilevitz first considered this approach in [32] and [16]. Together with Raymond these authors use this representation to construct a $k$-server protocol with $n^{O(\log\log k/(k \log k))}$ bits of communication [18].

We summarize the known upper bounds in Table 3.1 based on [18], [42] and [33].

| Tool | 2 DB | 3 DB | 4 DB | $k$ DB | Reference |
|---|---|---|---|---|---|
| Square/Cube | $n^{1/2}$ | - | $n^{1/2}$ | $k \cdot n^{1/\log k}$ | [33] |
| Covering Codes | $n^{1/3}$ | - | $n^{1/4}$ | $k \log k \cdot n^{1/(\log k + \log\log k)}$ | [33] |
| Polynomials | $n^{1/2}$ | $n^{1/3}$ | $n^{1/4}$ | $k^2 \log k \cdot n^{1/k}$ | [32] |
| Recursion | $n^{1/3}$ | $n^{1/5}$ | $n^{1/7}$ | $2^{k^2} n^{1/(2k-1)}$ | [4] |
| Linear Algebra | $n^{1/3}$ | $n^{1/5}$ | $n^{1/7}$ | $k! n^{1/(2k-1)}$ | [51] |
| Polynomials | $n^{1/3}$ | $n^{1/5}$ | $n^{1/7}$ | $k^3 n^{1/(2k-1)}$ | [17], [16] |
| Polynomials | $n^{1/3}$ | $n^{1/5.25}$ | $n^{1/7.87}$ | $n^{O(\log\log k/(k\log k))}$ | [18] |

Table 3.1: PIR Communication Complexity, up to a constant factor

### 3.2.4 Lower Bounds

Without the privacy condition, we require exactly $\log n + 1$ bits to transmit the index $i$ and receive $x_i$ as an answer. The communication complexity of private information retrieval can thus only be worse. But how much worse exactly?

Much effort has gone into determining good lower bounds for PIR, however the problem appears to be rather tricky [42]. Chor et al. [33] showed that for $k = 1$ servers, we require at least $n$ bits of communication. They also showed that for a *linear* PIR scheme with $k = 2$ servers and single bit answers, where the user makes one query to each server requires a query length of $n - 1$ bits. Beigel, Fortnow and Gasarch [15] showed that if we do not make any restrictions on the query, the query length must be at least $n - 2$ bits. If the server's answers are longer than 1 bit the following bounds are known: Goldreich, Karloff, Schulman and Trevisan showed that for 2-server *linear* PIR, with $t$-bit queries and $\ell$-bit answers, where the user only looks at $b$ predetermined positions in each answer, we have query length $t = \Omega(n/\ell^b)$ [47]. Kerenidis and de Wolf [53] showed that for *general* PIR, the query length satisfies $t = \Omega(n/2^{6\ell})$.

For the case of more than 2 servers, Mann [60] showed that we require at least $(k^2/(k-1) - \varepsilon)\log n$ bits of communication when we are using $k$ servers. For two servers this means the lower bound becomes $4 \log n$. This has been improved to $4.4 \log n$ [53].

In Chapter 5 we generalize the results of Goldreich et al. [47] and improve the results of Kerenidis and de Wolf [53].

## 3.3 Computationally Secure PIR

We can decrease the number of databases necessary to achieve sub-linear communication by relaxing the security of our model. If we are content with computationally secure PIR (CPIR), Chor and Gilboa [30] showed that for any $\varepsilon > 0$, there exists a 2-server PIR with communication complexity $O(n^\varepsilon)$ if one-way functions exist. The existence of one-way functions implies the existence of a pseudo-random generator $G$ which expands seeds of length $s(n) = n^{\delta}$ to length $n$, for any constant $\delta > 0$. To obtain a communication complexity of $O(n^{\varepsilon})$ the authors use $\delta = 1/t$ where the parameter $t$ is the first integer larger than $2/\varepsilon - 3$. The security of the system is based on the assumption that the expanded strings are pseudo-random with respect to poly($n$)-distinguishers: an adversary only has computational resources in poly($n$). This form of CPIR is referred to as CPIR with replication.

Contrary to the information-theoretically secure setting, computationally secure PIR also allows for sub-linear communication using only a single server without replication. This makes CPIR more applicable in practice. The security of these schemes, however, rests on the assumption that certain computational problems are infeasible for the server. So now we no longer grant the server unlimited computational power. In practice, the security of all known CPIR schemes rests on number theoretic assumptions, most notably the Quadratic Residuosity Problem [55] and the $\phi$-hiding assumption [25]. The Quadratic Residuosity Problem is to determine whether $z$ is a quadratic residue modulo $m$, where $\gcd(z, m) = 1$ and $z, m \in \mathbb{N}$. Recall that $z$ is a quadratic residue modulo $m$ if there exists an integer $a$ such that $a^2 \equiv z \mod m$. The $\phi$-hiding assumption also depends on a number theoretic concept. Recall that $\phi$ is the Euler totient function, where $\phi(m)$ gives the number of integers $k$ such that $0 < k < m$ and $\gcd(m, k) = 1$, i.e. $m$ and $k$ are relatively prime. Note that computing $\phi(m)$ on input $m$ is just as hard as factoring $m$ [25]. We say that a composite integer $m$ $\phi$-hides a prime $p$, if $p|\phi(m)$. The $\phi$-hiding assumption then states that it is computationally infeasible to decide whether a small prime $p$ divides $\phi(m)$, where $m$ is a composite integer of unknown factorization. More information about number theoretic concepts in use can be found in [62] and [72].

Much work has been done in the area of CPIR and we only give a brief overview here, as our focus rests on information-theoretic PIR. Kushilevitz and Ostrovsky make use of the Quadratic Residuosity Problem [55] to construct, for any $\varepsilon > 0$, a 1-server CPIR with $O(n^{\varepsilon})$ bits of communication. Their scheme has an additional communication complexity polynomial in a security parameter $t$, but since $t$ is quite small for all practical purposes, it is accounted for in $O(n^{\varepsilon})$ [25]. The same communication complexity is achieved by a protocol proposed by Yamamura and Saito [87] based on the subgroup membership problem, which generalizes the quadratic residuosity problem. In a later paper, Kushilevitz et al. [56] also show that one-way trapdoor permutations are sufficient for 1-server PIR. Their protocol also requires $O(n^{\varepsilon})$ of communication. Cachin, Micali and Stadler [25] give a 1-server PIR with communication complexity $O(\log^4 n)t$, where $t$ is a security parameter. Lipmaa [58] more recently constructed a 1-server PIR based on the Damgård-Jurik public key cryptographic system which further improves the communication complexity to $O(\log^2(n)) \cdot t$ for a security parameter $t$ which is small for all practical purposes. Other implementations have been considered by Chang [26] and Stern [77]. We summarize the known upper bounds for CPIR in Table 3.2.

| Assumption | 1 DB | 2 DB | Reference |
|---|---|---|---|
| Pseudo-Random Generator | | $O(n^\varepsilon)$ | [30] |
| Quadratic Residuosity | $O(n^\varepsilon)$ | | [55] |
| One-Way Trapdoor Permutation | $O(n^\varepsilon)$ | | [56] |
| Subgroup Membership | $O(n^\varepsilon)$ | | [87] |
| $\phi$-hiding Assumption | polylog$(n)$ | | [25] |
| Damgård-Jurik Public Key | $O((\log(n)\log\log(n)))$ | | [58] |

Table 3.2: CPIR Communication Complexity

## 3.4 Related Protocols and Extensions

There are several extensions to PIR and a number of related protocols. We briefly review some of them here for completeness' sake. We will only be concerned with the standard PIR model later on in this text.

### 3.4.1 $t$-private PIR

So far we assumed that the servers do not communicate with each other. In the setting of $t$-private PIR, we allow up to $t$ of the $k$ servers to collude. This means they are allowed to work together and pool their information. More formally we call a PIR scheme $t$-private, if no subset of $t$ servers can determine anything about $i$ [32]. The schemes we considered earlier are thus 1-private.

In their original conference paper [32], Chor et al. presented a $t$-private PIR with $O(tn^{t/k})$ bits of communication also based on polynomials. Since then Beimel, Ishai and Kushilevitz [17] have constructed a $t$-private PIR, for any $\varepsilon > 0$ and $k > t \geq 1$, where the user sends $O(\log n)$ bits to each server and receives $O(n^{t/k+\varepsilon})$ bits in return. Beimel et al. [16] have also given a scheme with complexity $O(n^{1/\lfloor(2k-1)/t\rfloor})$. More recently Blundo, D'Arco and DeSantis [22] gave a different $t$-private PIR with complexity $O(k\sqrt{n})$ for $t \leq k-1$. Their scheme is somewhat similar to the square scheme we examined earlier, however instead of a perfect square, they divide the database up into small blocks (i.e. columns). Their construction is interesting if $t > k/2$.

### 3.4.2 Robust PIR

What if some of the servers break down or send back the wrong answers? The PIR schemes considered earlier offer no protection against such faults. Beimel and Stahl [2] define a $k$-out-of-$m$ PIR as an $m$-server PIR, which still gives the correct answer even if only $k$ of the $m$ servers send back a reply. So far, all schemes we considered were $k$-out-of-$k$ PIRs. In particular, they show that if there exists a 2-out-of-2 PIR with $c$ bits of communication, then there exists a 2-out-of-$m$ PIR with communication $O(c \cdot m \log m)$. Since we have a 2-server PIR with $O(n^{1/3})$ bits of communication, this implies that there is a 2-out-of-$m$ PIR with complexity $O(n^{1/3} m \log m)$.

### 3.4.3 PIR with Preprocessing

So far we have only considered communication complexity. In practice, however, we may also be interested in reducing the workload of a server. In the schemes presented earlier, the database has to compute large sums over many of the $n$ bits of the database. Beimel, Ishai and Malkin [19] suggested that the workload may be reduced by preprocessing at the expense of extra storage. For example in the cube scheme above, we could precompute the xor of all possible cubes of the database.

The authors present several methods to solve this problem. In particular, they construct for any constants $k \geq 2$ and $\varepsilon > 0$: A $k$-server variant of their polynomial schemes with communication complexity $O(n^{1/(2k-1)})$ where the work of the database is $O(n/(\varepsilon \log n)^{2k-2})$ and $O(n^{1+\varepsilon})$ additional bits of storage are required; A $k$-server protocol with $O(n^{1/k+\varepsilon})$ communication and work and $n^{O(1)}$ storage; A protocol with a polylogarithmic number of servers, polylogarithmic communication and work, and $O(n^{1+\varepsilon})$ storage. And finally a computationally private $k$-server protocol with $O(n^{\varepsilon})$ communication, $O(n^{1/k+\varepsilon})$ work and $n^{O(1)}$ storage.

From the first two items, we can see that there is a tradeoff between communication complexity and the workload imposed on the server. The authors furthermore show that the product of extra storage required by the servers and the expected amount of work is at least linear in the database size $n$.

### 3.4.4 Retrieving Other Types of Data: Blocks and Keyword Searches

As we mentioned earlier, a straightforward way to allow each database item to be larger than one bit, say $m$ bits instead, is to execute $m$ instances of PIR. Thereby we retrieve the entire block bit by bit. A better scheme has been suggested by Chor et al. [33]. Suppose our database contains a total of $n$ bits which we break up into $n/s$ blocks of size $s$ each. Also suppose that there exists some $k$-server PIR with block size $s = 1$, query length $t$ and answer length $\ell$. Then there exists a $k$-server PIR with block size $s > 1$ where the user sends $t$ bits to each server, who sends $s\ell$ bits as a reply.

Chor, Gilboa and Naor also consider Private Information Retrieval by keywords [31]. In this setting the user can search the database for specific keywords, instead of an index $i$. This is very useful in the case that the user does not actually know the index. Imagine for example an online search engine such as Google. Clearly you would not want to remember individual indices, even if you could obtain them somehow. More formally, PIR by Keyword assumes a database containing $n$ strings of length $s$. The user can now query for a certain keyword $w \in \{0,1\}^s$. The user now wants to determine if $w$ is in this list of strings, without the database learning anything about $w$. The authors give solutions to this problem with complexity $\sum_{q=1}^{\log n} C(s, 2^{q-1}, k)$ where $C(s, n, k)$ is the communication complexity of a $k$-server PIR with an $n$-block database of $s$-bit blocks.

### 3.4.5 Repudiative Information Retrieval

A slightly weaker scenario has been considered by Asonov and Freytag [10] in the form of Repudiative Information Retrieval (RIR). Here the server is allowed to gain some information about the desired index $i$. However it must be possible for the user to later deny any claims that he retrieved a particular item. More formally the repudiation property is preserved if and only if for all $i$ we have $0 < \Pr(q = i | I_{revealed}) < 1$, where $I_{revealed}$ is the information obtained by the database, $q$ denotes the queried index. The probability is taken over the protocols behaviour. The distribution depends on the actual RIR scheme used. For RIR that is equal to information-theoretic PIR this is the uniform distribution.

### 3.4.6 Symmetrically-Private Information Retrieval

So far the user was allowed to learn more about the database than the entry he asked for. Our only requirement was that the database learns nothing about the identity of the queried item. In Symmetrically-Private Information Retrieval (SPIR), we additionally also demand that the user learns exactly the item he asked for and no more. In this setting, the privacy constraint is thus extended to the data. SPIR has first been considered by Gertner, Ishai, Kushilevitz and Malkin [44] [59], who present a $k$-server SPIR scheme with communication complexity $O(n^{1/(2k-1)})$. The same authors also show there is no information theoretically secure SPIR with $k$ non-communicating databases, even if the databases are allowed to hold private and independent random inputs, and the user is honest. This impossibility result holds even if multiple rounds of interactions between the databases and the user are allowed. In particular, this means that single database SPIR is impossible. However, a protocol for PIR can be extended to SPIR when the servers are allowed private *shared* randomness which is hidden from the user. Kerenidis and de Wolf give a honest user *quantum* protocol which does not make use of any shared randomness [54].

A 1-server SPIR scheme is also known as $\binom{n}{1}$-OT (1-out-of-$n$ oblivious transfer), and has received considerable attention in the literature. A $k$-server SPIR corresponds to distributed $\binom{n}{1}$-OT. Whereas the goal is the same, research on SPIR focuses on the communication complexity of the problem, whereas research about OT is primarily concerned with cryptographic security and how OT relates to other cryptographic primitives.

A similar problem has also been considered as early as 1987 by Brassard, Robert and Crépeau under the name of All-Or-Nothing Disclosure of Secrets [24]. Here the scenario is phrased in terms of a vendor and a buyer: The vendor wants to sell information to the buyer, however, the buyer should obtain exactly one item of information and not more. For example the buyer could have items such as “where is Bin Laden” available for purchase. The buyer wants to obtain information from the vendor, without revealing what information he is interested in. After all, the vendor may otherwise add another item to his information shop, such as “people who want to know where Bin Laden is”. Another computationally secure protocol for this problem was presented later by Stern [77]. This problem was already partially solved in 1970 by Wiesner [85] fulfilling the privacy requirements for the vendor. The vendor can encode and send up to three messages using quantum states, no two of which can be received by the buyer.

### 3.4.7 Further PIR Variations

Another variant of PIR introduces a third party that can help facilitate the protocol, without learning anything by itself. This is a general technique introduced by Beaver [13] also applied outside the realm of PIR. DiCrescenzo et al. [38] first applied this technique to PIR and thereby created Commodity Based PIR. Here the third party provides the server and the user with a shared random string. The length of this string is also referred to as the commodity complexity. The goal is to reduce direct communication between the user and the server at the expense of a higher commodity complexity. The server and the user can make use of the newfound resource to perform a $k$-server PIR with communication complexity $O(\log n)$ and commodity complexity $O(n^{1/(k-1)})$.

Gertner et al. [43] consider a $k$-server PIR scheme, where the database is not replicated over all $k$ servers. Their aim is to keep the database $x$ itself hidden from up to $t$ colluding database servers.

Finally in the realm of practical PIR applications, Smith and Safford [75] consider a PIR using additional specialized hardware in the form of Secure Coprocessors. These tamper-proof devices are trusted to carry out computations without the possibility of an adversary interfering, even if the adversary has physical access to the device. The user thereby sends its query encrypted with the public key of the device to the server. The key idea is that only the device can decrypt the users query. It now reads the entire database, encrypts the requested item with a key supplied by the user and sends it back. This scheme is thus computationally secure. Asonov et al. [9] considered the use of preprocessing using Secure Coprocessors as well. This has important practical implications, since such devices are currently available on the market.

### 3.4.8 Quantum PIR

Finally, a quantum PIR is characterized by a quantum server and communication over a quantum channel. The definition for PIR above can thus be generalized to the quantum case. Kerenidis and de Wolf [53] constructed a 2-server *quantum* PIR with communication complexity $O(n^{3/10})$ based on a 4-server *classical* PIR with binary answers.

## 3.5 Summary

We have explored the notion of private information retrieval. PIR allows a user to retrieve an entry from a database, without revealing the index of this entry. This cryptographic primitive forms an important building block for privacy-sensitive applications. Even though sub-linear communication is not possible in the setting of information-theoretic security when using only a single server, we saw that using multiple servers does give an improvement. We have also examined known lower bounds for this problem, which we will improve upon in Chapter 5. However, before we can prove these new bounds, we first turn to a coding method closely related to PIR: locally decodable codes.

# Chapter 4

# Locally Decodable Codes

## 4.1 Introduction

Error-correcting codes allow reliable transmission and storage of information in noisy environments. Consider for example transmissions over a wireless network. During transit some of the bits may be corrupted, which we would like to restore using an error-correcting code. Likewise the data stored on a CD or your hard drive is protected against noise by such codes. In these applications we usually encode each block of data separately. This has the advantage that we can decode the data block by block depending on what portion of the data we are interested in. However, data will be lost forever if one block is corrupted in such a way we can no longer reconstruct it. Consider for example a game contained on a single CD. If one block of data, viz. one part of the executable is missing, the user will be unable to run the program.

A different approach is to encode the entire body of data in a single codeword and not split it up in blocks first. This would make our encoding more resilient against errors. The disadvantage of this approach is that in order to restore any part of the data, we would need to read the entire codeword at least once. Clearly this is not very practical. Imagine for example that to read a single file in your home directory we would need to examine the entire contents of the hard drive.

*Locally decodable codes* (LDCs) differ from standard error-correcting codes. Here any bit can be reconstructed by reading only a few randomly chosen locations in the codeword. Thus a locally decodable code $C : \{0,1\}^n \to \Sigma^m$ over alphabet $\Sigma$ is an error-correcting code that allows efficient decoding of individual bits of the encoded information: given any string $y$ that is sufficiently close to the real codeword $C(x)$, we can probabilistically recover any bit $x_i$ of the original input $x$, while only looking at $k \leq m$ positions of $y$. The code length $m$ measures the cost of the encoding, while $k$ measures the efficiency of decoding individual bits. The main complexity question of interest here is the tradeoff between the code length, $m$, and the number of queries, $k$. Think of applications encoding a large chunk of data in order to protect it from noise, where we are only interested in extracting small pieces at a time. Imagine for example an encoding of all books in a library, where we would like to retrieve only the first paragraph of this text.

**Outline**

We first define LDCs and smooth codes. Then we take a look at known upper and lower bounds for this problem. In Chapter 5 we improve upon the best known lower bounds. LDCs have a very close relation to the problem of PIR examined in the last chapter. Finally, we illustrate this fact by showing how to construct an LDC starting with a PIR scheme.

## 4.2 Definition

Formally we can define a locally decodable code as follows.

**Definition 4.2.1** $C : \{0,1\}^n \rightarrow \Sigma^m$ *is a* $(k, \delta, \varepsilon)$-locally decodable code (LDC)*, if there exists a classical randomized decoding algorithm* $A$ *with input* $i \in [n]$ *and oracle access to a string* $y \in \Sigma^m$ *such that*

1. *$A$ makes $k$ distinct queries $j_1, \ldots, j_k$ to $y$, non-adaptively, gets query answers $a_1 = y_{j_1}, \ldots, a_k = y_{j_k}$ and outputs a bit $f(a_1, \ldots, a_k)$, where $f$ depends on $i$ and $A$'s randomness.*

2. *For every $x \in \{0,1\}^n$, $i \in [n]$, and $y \in \Sigma^m$ with Hamming distance $d(y, C(x)) \leq \delta m$ we have $\Pr[f(a_1, \ldots, a_k) = x_i] \geq 1/2 + \varepsilon$.*

*Here probabilities are taken over $A$'s internal randomness. For $\Sigma = \{0,1\}^\ell$, we say the LDC* uses $b$ bits*, if $A$ only uses $b$ predetermined bits of each query answer: it outputs $f(a_{1|S_1}, \ldots, a_{k|S_k})$ where the sets $S_1, \ldots, S_k$ are of size $b$ each and are determined by $i$ and $A$'s randomness.*

*The LDC is called* linear*, if $C$ is a linear function over $GF(2)$ (i.e. $C(x + y) = C(x) + C(y)$).*

## 4.3 Smooth codes

Related to locally decodable codes are so called *smooth* codes. These are codes where the decoding algorithm spreads its queries "smoothly" across the codeword, meaning it queries no code location too frequently. Since our proofs will make use of the notion of smooth codes, we review the definition here.

**Definition 4.3.1** $C : \{0,1\}^n \rightarrow \Sigma^m$ *is a* $(k, c, \varepsilon)$-smooth code (SC)*, if there exists a classical randomized decoding algorithm* $A$ *with input* $i \in [n]$ *and oracle access to* $C(x)$ *such that*

1. *$A$ makes $k$ distinct queries $j_1, \ldots, j_k$ to $C(x)$, non-adaptively, gets query answers $a_1 = C(x)_{j_1}, \ldots, a_k = C(x)_{j_k}$ and outputs a bit $f(a_1, \ldots, a_k)$, where $f$ depends on $i$ and $A$'s randomness.*

2. *For every* $x \in \{0,1\}^n$ *and* $i \in [n]$ *we have* $\Pr[f(a_1, \ldots, a_k) = x_i] \geq 1/2 + \varepsilon$.

3. *For every* $x \in \{0,1\}^n$, $i \in [n]$ *and* $j \in [m]$, $\Pr[A$ *queries* $j] \leq c/m$.

*The smooth code* uses $b$ bits*, if $A$ only uses $b$ predetermined bits of each query answer.*

Note that the decoder of smooth codes deals only with valid codewords $C(x)$. The decoding algorithm of an LDC on the other hand can deal with corrupted codewords $y$ that are still sufficiently close to the original. Katz and Trevisan [52, Theorem 1] showed that LDCs and smooth codes are closely related:

**Theorem 4.3.2 (Katz & Trevisan)** *If* $C : \{0,1\}^n \to \Sigma^m$ *is a* $(k, \delta, \varepsilon)$*-locally decodable code, then* $C$ *is also a* $(k, k/\delta, \varepsilon)$*-smooth code (the property of using* $b$ *bits carries over).*

## 4.4 Upper Bounds

### 4.4.1 Examples

To get a better feel for LDCs, we examine two of them in more detail: The Hadamard Code and LDCs based on PIR.

#### Hadamard Code

As an example of a locally decodable code, we take a look at the so called Hadamard code. Here we encode each $x$ of length $n$ by setting the $j$-th bit of the codeword to

$$C(x)_j = j \cdot x \mod 2$$

for all possible $j \in \{0,1\}^n$. Each codeword has then exactly $m = 2^n$ bits.

If we now want to retrieve the $i$-th bit of $x$, we randomly choose a $j \in \{0,1\}^n$ and read bits $C(x)_j$ and $C(x)_{j'}$ with $j' = j \oplus e_i$ from the encoding. We then xor the two replies to get $C(x)_j \oplus C(x)_{j \oplus e_i} = (j \cdot x) \oplus ((j \oplus e_i) \cdot x) \mod 2 = (e_i \cdot x) \mod 2 = x_i$. Figure 4.1 visualizes this concept.

We can thus retrieve $x_i$ perfectly when there is no noise at all, since then $y = C(x)$. What happens if the codeword is corrupted? Suppose we have $d(y, C(x)) = \delta m$. The parameter $\delta$ thus determines the fraction of corrupted bits. Alternatively we can also view $\delta$ as the probability of a bit being corrupted, which gives an expected number of $\delta m$ corrupted bits. Looking at our reconstruction algorithm, it is easy to see that it fails if either $C(x)_j$ or $C(x)_{j \oplus e_i}$ is corrupted. This happens with probability at most $\delta + \delta = 2\delta$. We can therefore reconstruct $x_i$ with probability $1 - 2\delta$.

Since we want a recovery probability of $1/2 + \varepsilon$, we have $\varepsilon = 1/2 - 2\delta > 0$. This means that we expect to reconstruct $x_i$ with a small bias as long as $\delta < 1/4$.

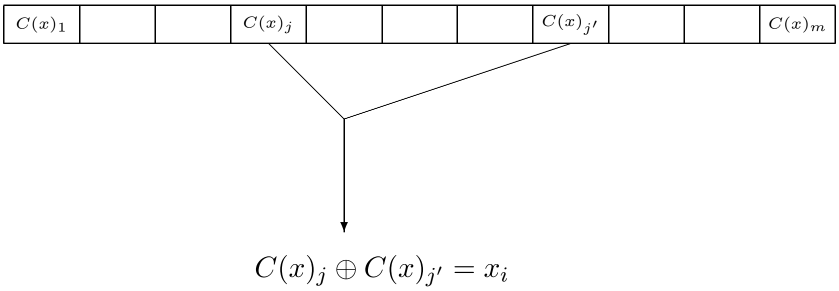

Figure 4.1: 2-query Locally Decodable Code

### LDC based on PIR

As indicated earlier, there is a close relation between LDCs and the notion of PIR. Goldreich et al. [47, Lemma 7.1] formalized this connection in the following lemma:

**Lemma 4.4.1 (GKST)** *Suppose there is a one-round, $(1-\eta)$-secure PIR scheme with two servers, database size $n$, query size $t$, answer size $\ell$, and recovery probability at least $1/2+\varepsilon$. Then there is a $(2,3,\varepsilon-\eta)$-smooth code $C : \{0,1\}^n \rightarrow (\{0,1\}^\ell)^m$, where $m \leq 6 \cdot 2^t$. Furthermore:*

1. *If in the PIR scheme the answer bits are a linear combination of the data, then $C$ is linear.*

2. *If in the PIR scheme, the user only uses $b$ predetermined bits of the $\ell$ bits it receives as an answer to each question, then the same property is true for the decoding algorithm of $C$.*

The main idea behind their proof is to view PIR as an encoding of the database $x \in \{0,1\}^n$. The codeword $C(x)$ then consists of the answers a PIR system could give for each conceivable query to each of the two servers; hence $C(x) \in (\{0,1\}^\ell)^{2 \cdot 2^t}$. The user can now reconstruct the bit $x_i$ from $C(x)$ by looking at the two entries in the codeword corresponding to the queries he would have sent to the two PIR servers to retrieve $x_i$. This gives him $C(x)_{q_0} = a_0$ and $C(x)_{q_1} = a_1$. He can then calculate $x_i$ from answers $a_0$ and $a_1$ as in the PIR protocol. The privacy condition of PIR is translated into the smoothness property of the resulting code. You may wonder why we have $m \leq 6 \cdot 2^t$ above, instead of only a factor of 2. This is due to the fact that some entries are replicated in order to obtain a smooth encoding, i.e. where each entry is uniformly distributed.

Let's see how this works in practice. Consider the simple square scheme PIR we examined in Section 3.2.2. Here the database was arranged into a $\sqrt{n} \times \sqrt{n}$ grid. The user randomly chose $\sqrt{n}$ bits which formed the first query, $q_0$. The second query was then of the form $q_1 = q_0 \oplus e_i$. This means the set of all possible queries is the set of all $\sqrt{n}$-bit strings. We thus have $m = O(2^{\sqrt{n}})$ and $\ell = \sqrt{n}$. The $j$-th entry of the codeword is then given by $C(x)_j \in \{0,1\}^{\sqrt{n}}$ with $C(x)_j = a_j$ the answer of the server, which was the inner product of the query with each column. The user then selects $b = 1$ bits from each $C(x)_j$ and xors them together to get $x_i$ as in the original Square PIR Scheme. We thus see that also the number of bits selected carries over to the smooth code $C : \{0,1\}^n \rightarrow (\{0,1\}^{\sqrt{n}})^{O(2^{\sqrt{n}})}$.

Likewise we can also construct a PIR scheme based on a smooth encoding [52]. Given a $(k, k, \varepsilon)$-smooth code $C : \{0,1\}^n \rightarrow \Sigma^m$, where each position is equally likely to be queried, we can build a $k$-server PIR with query length $\log(m)$ and answer length $\log(|\Sigma|)$ and recovery probability $1/2 + \varepsilon$.

We will make use of this intricate connection later on, to derive new lower bounds for PIR from lower bounds for LDCs.

### 4.4.2 Overview

Several LDC schemes are known. Note that an error-correcting code which produces a single codeword of length $O(n)$ where we have to read the entire codeword to reconstruct a single bit, can be phrased as an LDC with $m = O(n)$ and number of queries $k = m$. The Hadamard code, which we examined above, is an example of an LDC which uses less then $m$ queries. It has a code length of $2^n$. Babai et al. [11] construct an LDC with $m = \text{poly}(n)$ provided that the number of queries is polylogarithmic in $n$. Furthermore, Beimel et al. [18] use the connection between PIR and LDCs to construct a binary code (i.e. where each codeword entry corresponds to 1 bit) of length $2^{n^{O(\log\log k/(k \log k))}}$.

We can summarize these known bounds in a small table:

| Name/Based on | Code length | Reference | Notes |
|---|---|---|---|
| Standard ECC | $O(n)$ | | $k = O(n)$ |
| Hadamard Code | $2^n$ | | k = 2 |
| LDC from PIR | $2^{n^{O(\log\log k/(k \log k))}}$ | [18] | |
| Babai et al. | $\text{poly}(n)$ | [11] | $k = \text{polylog}(n)$ |

Table 4.1: LDC Code Length

## 4.5 Lower Bounds

As mentioned earlier, the main complexity question of interest is the tradeoff between $m$ and $k$. Except for the $k = 2$ case with fairly small alphabet $\Sigma$, no good lower bounds are known.

Katz and Trevisan [52], who first initiated the search for lower bounds for LDCs, showed that for a single query $k = 1$, LDCs do not exist if $n$ is larger than some constant depending on $\delta$ and $\varepsilon$. This reflects the impossibility of a single-server PIR with sub-linear communication and information theoretic security. The same authors also present a super-linear but at most quadratic lower bound for a constant number of queries $k \geq 2$: $m = \Omega(n^{1+1/(q-1)})$.

Goldreich et al. [47] showed that $m = \Omega(2^{\delta\varepsilon n/8})$, if $C$ is a *linear* code. This is an exponential lower bound for linear codes with $k = 2$ queries and constant alphabet. This has been improved for linear binary codes ($\ell = 1$) by Obata [66] to $m \geq 2^{\Omega(\delta n/(1-2\varepsilon))}$, which is optimal. Kerenidis and de Wolf [53] extended the result of Goldreich et al. to *all* codes, using techniques from quantum computing. For alphabet $\Sigma = \{0,1\}^\ell$ their lower bound is

$$m = 2^{\Omega(n/2^{5\ell})}.$$

They also slightly improved the polynomial lower bounds of [52] for $k > 2$. Clearly the above lower bound becomes trivial if each position of the codeword has $\ell \geq \log(n)/5$ bits.

In the next chapter, we will analyze the case where $\ell$ can be much larger, but the decoder uses only $b$ bits out of the $\ell$ bits that a query gives. The $b$ positions that he uses may depend on the index $i$ he is interested in, as well as his randomness. Goldreich et al. [47] also analyzed this situation, and showed the following lower bound for *linear* codes:

$$m = 2^{\Omega(n/\sum_{i=0}^{b}\binom{\ell}{i})}.$$

Note that even though LDC and PIR are closely related, the focus of the lower bounds is slightly different. In the LDC scenario, we are interested in finding a lower bound for the code length $m$ when using a small alphabet. This corresponds to a lower bound for the query length $t$ in PIR as we saw in the previous section. In PIR however, we are additionally interested in a lower bound on the total amount of communication given by the sum of the query and answer length, $(t + \ell)$. In terms of an LDC this means we desire a bound on $\log(m) + \log(|\Sigma|)$.

## 4.6 Summary

We have examined the notion of locally decodable codes, which are error-correcting codes that allow efficient reconstruction of individual bits from a codeword, without having to read all of it. An example of a locally decodable code is the Hadamard code. We can also construct a locally decodable code based on private information retrieval, which illustrates the close connection between the two. Finally, we have introduced the notion of a smooth code, which we will require in the construction of our proofs in the following chapter.

# Chapter 5

# Improved Lower Bounds for LDCs and PIRs

This chapter is based on ”Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval”, S. Wehner and R. de Wolf, quant-ph/0403140

## 5.1 Introduction

Now that we have established the notion of private information retrieval (PIR) and locally decodable codes (LDC), we can turn to proving new lower bounds for both. Here we analyze the case where the answer length $\ell$ is very large, but the decoder uses only $b$ bits out of the $\ell$ bits that a query gives. The $b$ positions that he uses may depend on the index $i$ he is interested in, as well as his randomness. For example, the LDC constructed from the Square PIR scheme in Section 3.2.2 is of this form with $b = 1$. Likewise, the Cube PIR scheme from Section 3.2.2 has a large answer length $\ell$, but the user only selects $b = 3$ bits of each answer. This setting is therefore interesting because many existing constructions are of this form, for quite small $b$. Note again, that Goldreich et al. [47] also analyzed this situation, and showed the following lower bound for *linear* codes:

$$m = 2^{\Omega(n/\sum_{i=0}^{b}\binom{\ell}{i})}.$$

Here we will prove a slightly weaker lower bound for *all* codes:

$$m = 2^{\Omega(n/2^b\sum_{i=0}^{b}\binom{\ell}{i})}.$$

In particular, if $b = \ell$ the bound from [53] is improved to

$$m = 2^{\Omega(n/2^{2\ell})}.$$

We lose a factor of $2^b$ compared to Goldreich et al. This factor can be dispensed with if the decoder outputs the parity of a subset of the bits he receives. All known LDCs are of this type.

Our proofs for LDCs are completely different from the combinatorial approach of Goldreich et al. Similar to [53], we proceed in several steps: (1) we reduce the LDC to a smooth code (2) we then reduce the two classical queries to one *quantum* query and (3) show a lower bound for the induced one-quantum-query-decodable code by deriving a *random access code* from it. We continue by giving improved bounds for PIR, by using the relationship between LDCs and PIR established earlier.

The main novelty is a tight analysis of the following problem, which may be of independent interest: How well can we compute $f(a_0, a_1)$ given a quantum superposition $\frac{1}{\sqrt{2}}(|0, a_0\rangle + |1, a_1\rangle)$ of both halves of the input?

## 5.2 Computing $f(a_0, a_1)$ from Superposed Input

This analysis is the main tool we will require to prove new lower bounds for LDCs and PIRs. Consider the state $|\Psi_{a_0 a_1}\rangle = \frac{1}{\sqrt{2}}(|0, a_0\rangle + |1, a_1\rangle)$ with $a_0, a_1$ both $b$-bit strings. We show that we can compute *any* Boolean function $f(a_0, a_1)$ with bias $1/2^{b+1}$ given one copy of this state. After that we show that bias optimal, if $f$ is the $2b$-bit parity function.

### 5.2.1 Upper bound

The key to constructing the algorithm is the following observation:

**Lemma 5.2.1** *For every function $f : \{0,1\}^{2b} \to \{0,1\}$ there exist non-normalized states $|\varphi_a\rangle$ such that*

$$U : |a\rangle|0\rangle \to \gamma \sum_{w \in \{0,1\}^b} (-1)^{f(w,a)}|w\rangle|0\rangle + |\varphi_a\rangle|1\rangle,$$

*with $\gamma = 1/2^b$, is unitary.*

**Proof.** Let $|\psi_a\rangle = \gamma \sum_{w\in\{0,1\}^b}(-1)^{f(w,a)}|w\rangle|0\rangle + |\varphi_a\rangle|1\rangle$. Recall that $U$ is unitary if and only if $\langle\psi_a|\psi_{a'}\rangle = \delta_{aa'}$ for all $a, a'$. We show that we can choose $|\varphi_a\rangle$ to achieve this.

First, since $\langle w|w'\rangle = \delta_{ww'}$ and $\langle w, 0|\varphi_a, 1\rangle = 0$, we have

$$\langle\psi_a|\psi_{a'}\rangle = \gamma^2 \sum_{w\in\{0,1\}^b} (-1)^{f(w,a)+f(w,a')} + \langle\varphi_a|\varphi_{a'}\rangle.$$

Let $C$ denote the $2^b \times 2^b$ matrix with entries $C_{aa'} = \gamma^2 \sum_{w\in\{0,1\}^b}(-1)^{f(w,a)+f(w,a')}$ where the indices $a$ and $a'$ are $b$-bit strings. From the definition of $C_{aa'}$ we have $|C_{aa'}| \leq 1/2^b$ for $\gamma = 1/2^b$. Then by [50, Corollary 6.1.5], the largest eigenvalue of $C$ is

$$\lambda_{max}(C) \leq \min\left\{\max_a \sum_{a'\in\{0,1\}^b} |C_{aa'}|, \max_{a'} \sum_{a\in\{0,1\}^b} |C_{aa'}|\right\} \leq \sum_{a\in\{0,1\}^b} \frac{1}{2^b} = 1.$$

However, $\lambda_{max}(C) \leq 1$ implies that the Hermitian matrix $I - C$ is positive semidefinite and hence, by [50, Corollary 7.2.11], $I - C = A^\dagger A$ for some matrix $A$. Now define $|\varphi_a\rangle$ to be the $a$th column of $A$. Since the matrix $C + A^\dagger A = I$ is composed of all inner products $\langle\psi_a|\psi_{a'}\rangle$, we have $\langle\psi_a|\psi_{a'}\rangle = \delta_{aa'}$ and it follows that $U$ is unitary. □

Using these observations, we can now prove the following theorem.

**Theorem 5.2.2** *Suppose $f : \{0,1\}^{2b} \to \{0,1\}$ is a Boolean function. There exists a quantum algorithm to compute $f(a_0, a_1)$ with success probability $1/2 + 1/2^{b+1}$ using one copy of $|\Psi_{a_0 a_1}\rangle = \frac{1}{\sqrt{2}}(|0, a_0\rangle + |1, a_1\rangle)$, with $a_0, a_1 \in \{0,1\}^b$.*

**Proof.** First we extend the state $|\Psi_{a_0 a_1}\rangle$ by a $|0\rangle$-qubit. Let $U$ be as in Lemma 5.2.1. Applying the unitary transform $|0\rangle\langle 0| \otimes I^{\otimes b+1} + |1\rangle\langle 1| \otimes U$ to $|\Psi_{a_0 a_1}\rangle|0\rangle$ gives

$$\frac{1}{\sqrt{2}}\left(|0\rangle|a_0\rangle|0\rangle + |1\rangle\left(\frac{1}{2^b}\sum_{w\in\{0,1\}^b}(-1)^{f(w,a_1)}|w\rangle|0\rangle + |\varphi_{a_1}\rangle|1\rangle\right)\right).$$

Define $|\Gamma\rangle = |a_0\rangle|0\rangle$ and $|\Lambda\rangle = \frac{1}{2^b}\sum_w (-1)^{f(w,a_1)}|w\rangle|0\rangle + |\varphi_{a_1}\rangle|1\rangle$. Then $\langle\Gamma|\Lambda\rangle = \frac{1}{2^b}(-1)^{f(a_0,a_1)}$ and the above state is

$$\frac{1}{\sqrt{2}}(|0\rangle|\Gamma\rangle + |1\rangle|\Lambda\rangle).$$

We apply a Hadamard transform to the first qubit to get

$$\frac{1}{2}\left(|0\rangle(|\Gamma\rangle + |\Lambda\rangle) + |1\rangle(|\Gamma\rangle - |\Lambda\rangle)\right).$$

The probability that a measurement of the first qubit yields a 0 is

$$\frac{1}{4}\langle\Gamma + \Lambda|\Gamma + \Lambda\rangle = \frac{1}{2} + \frac{1}{2}\langle\Gamma|\Lambda\rangle = \frac{1}{2} + \frac{(-1)^{f(a_0,a_1)}}{2^{b+1}}.$$

Thus by measuring the first qubit, we obtain the value of $f(a_0, a_1)$ with bias $1/2^{b+1}$. □

To prove that the above algorithm is optimal for the parity function, we need to consider how well we can distinguish two density matrices $\rho_0$ and $\rho_1$. By distinguishing we mean that given an unknown state, we can determine whether it is $\rho_0$ or $\rho_1$. Let $\parallel A \parallel_{tr}$ denote the trace norm of matrix $A$, which equals the sum of its singular values.

**Lemma 5.2.3** *Two density matrices $\rho_0$ and $\rho_1$ cannot be distinguished with probability better than $\frac{1}{2} + \frac{\parallel \rho_0 - \rho_1 \parallel_{tr}}{4}$.*

**Proof.** The most general way of distinguishing $\rho_0$ and $\rho_1$ is a POVM with two operators $E_0$ and $E_1$, such that $p_0 = tr(\rho_0 E_0) \geq 1/2 + \varepsilon$ and $q_0 = tr(\rho_1 E_0) \leq 1/2 - \varepsilon$. Then $|p_0 - q_0| \geq 2\varepsilon$ and likewise, $|p_1 - q_1| \geq 2\varepsilon$. By [65, Theorem 9.1], $\| \rho_0 - \rho_1 \|_{tr} = \max_{\{E_0,E_1\}}(|p_0 - q_0| + |p_1 - q_1|)$ and thus $\| \rho_0 - \rho_1 \|_{tr} \geq 4\varepsilon$. Hence $\varepsilon \leq \| \rho_0 - \rho_1 \|_{tr}/4$. □

**Theorem 5.2.4** *Suppose that $f$ is the parity of $a_0a_1$. Then any quantum algorithm for computing $f$ from one copy of $|\Psi_{a_0a_1}\rangle$ has success probability $\leq 1/2 + 1/2^{b+1}$.*

**Proof.** Define $\rho_0$ and $\rho_1$ by

$$\rho_c = \frac{1}{2^{2b-1}} \sum_{a_0a_1 \in f^{-1}(c)} |\Psi_{a_0a_1}\rangle\langle\Psi_{a_0a_1}|,$$

with $c \in \{0,1\}$. A quantum algorithm that computes the parity of $a_0a_1$ with probability $1/2 + \varepsilon$ can be used to distinguish $\rho_0$ and $\rho_1$. Hence from Lemma 5.2.3 we have $\varepsilon \leq \| \rho_0 - \rho_1 \|_{tr}/4$.

Let $A = \rho_0 - \rho_1$. It is easy to see that the $|0, a_0\rangle\langle 0, a_0|$-entries are the same in $\rho_0$ and in $\rho_1$, so these entries are 0 in $A$. Similarly, the $|1, a_1\rangle\langle 1, a_1|$-entries in $A$ are 0. In the off-diagonal blocks, the $|0, a_0\rangle\langle 1, a_1|$-entry of $A$ is $(-1)^{|a_0|+|a_1|}/2^{2b}$. For $|\phi\rangle = \frac{1}{\sqrt{2^b}} \sum_{w \in \{0,1\}^b} (-1)^{|w|}|w\rangle$ we have

$$|\phi\rangle\langle\phi| = \frac{1}{2^b} \sum_{a_0,a_1} (-1)^{|a_0|+|a_1|}|a_0\rangle\langle a_1|$$

and hence

$$A = \frac{1}{2^b}(|0, \phi\rangle\langle 1, \phi| + |1, \phi\rangle\langle 0, \phi|).$$

Let $U$ and $V$ be unitary transforms such that $U|0, \phi\rangle = |0, 0^b\rangle$, $U|1, \phi\rangle = |1, 0^b\rangle$ and $V|0, \phi\rangle = |1, 0^b\rangle$, $V|1, \phi\rangle = |0, 0^b\rangle$. Then

$$UAV^\dagger = \frac{1}{2^b}(U|0, \phi\rangle\langle 1, \phi|V^\dagger + U|1, \phi\rangle\langle 0, \phi|V^\dagger) = \frac{1}{2^b}(|0, 0^b\rangle\langle 0, 0^b| + |1, 0^b\rangle\langle 1, 0^b|).$$

Since $UAV^\dagger$ is diagonal, its only non-zero singular values are $\sigma_1 = \sigma_2 = 1/2^b$. Hence

$$\| \rho_0 - \rho_1 \|_{tr} = \| A \|_{tr} = \| UAV^\dagger \|_{tr} = \sum_i \sigma_i = \frac{2}{2^b},$$

so $\varepsilon \leq \| \rho_0 - \rho_1 \|_{tr}/4 = 1/2^{b+1}$. □

## 5.3 Lower Bounds for Locally Decodable Codes that Use Few Bits

We now make use of the technique developed above to prove new lower bounds for 2-query LDCs over non-binary alphabets. First we define the notion of a *quantum* smooth code analog to the definition of a *classical* smooth code provided in Section 4.3. We then construct a 1-query quantum smooth code (QSC) from a 2-query classical smooth code (SC), and then prove lower bounds for QSCs. Finally, we will use the connection between smooth codes and LDCs to improve the best known lower bounds for LDCs. In the sequel, we will index the two queries by 0 and 1 instead of 1 and 2, to conform to the two basis states $|0\rangle$ and $|1\rangle$ of a qubit.

### 5.3.1 Quantum Smooth Code

We first define the notion of a 1-query *quantum* smooth code. The following definition is rather ad hoc and not the most general possible, but sufficient for our purposes.

**Definition 5.3.1** *$C : \{0,1\}^n \to \Sigma^m$ is a $(1, c, \varepsilon)$-*quantum smooth code (QSC)*, if there exists a quantum decoding algorithm $A$ with input $i \in [n]$ and oracle access to $C(x)$ such that*

1. *$A$ probabilistically picks a string $r$, makes a query of the form*

$$|Q_{ir}\rangle = \frac{1}{\sqrt{2}}\left(|j_{1r}\rangle \frac{1}{\sqrt{2^b}} \sum_{T \subseteq S_{1r}} |z_T\rangle + |j_{2r}\rangle \frac{1}{\sqrt{2^b}} \sum_{T \subseteq S_{2r}} |z_T\rangle\right)$$

   *and returns the outcome of some quantum measurement on the resulting state.*

2. *For every $x \in \{0,1\}^n$ and $i \in [n]$ we have* $\Pr[A$ *outputs* $x_i] \geq 1/2 + \varepsilon$.

3. *For every $x \in \{0,1\}^n$, $i \in [n]$ and $j \in [m]$,* $\Pr[A$ *queries* $j] \leq c/m$.

*The QSC* uses $b$ bits*, if the sets $S_{1r}, S_{2r}$ have size $b$.*

### 5.3.2 Constructing a 1-query QSC from a 2-query SC

As the first step of our proof, we turn a 2-query classical smooth code into a 1-query quantum smooth code.

**Theorem 5.3.2** *If $C : \{0,1\}^n \to (\{0,1\}^\ell)^m$ is a $(2, c, \varepsilon)$-smooth code that uses $b$ bits, then $C$ is a $(1, c, \varepsilon/2^b)$-quantum smooth code that uses $b$ bits.*

**Proof.** Fix index $i \in [n]$ and encoding $y = C(x)$. The 1-query quantum decoder will pick a random string $r$ with the same probability as the 2-query classical decoder. This $r$ determines two indices $j_0, j_1 \in [m]$, two $b$-element sets $S_0, S_1 \subseteq [\ell]$, and a function $f : \{0,1\}^{2b} \to \{0,1\}$ such that

$$\Pr[f(y_{j_0|S_0}, y_{j_1|S_1}) = x_i] = p \geq \frac{1}{2} + \varepsilon,$$

where the probability is taken over the decoder's randomness. Assume for simplicity that $j_0 = 0$ and $j_1 = 1$, and define $a_0 = y_{j_0|S_0}$ and $a_1 = y_{j_1|S_1}$. We now construct a 1-query quantum decoder that outputs $f(a_0, a_1)$ with probability $1/2 + 1/2^{b+1}$, as follows. The quantum query is

$$|Q_{ir}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle\frac{1}{\sqrt{2^b}}\sum_{T\subseteq S_0}|z_T\rangle + |1\rangle\frac{1}{\sqrt{2^b}}\sum_{T\subseteq S_1}|z_T\rangle\right).$$

The result of this query is

$$\frac{1}{\sqrt{2}}\left(|0\rangle\frac{1}{\sqrt{2^b}}\sum_{T\subseteq S_0}(-1)^{a_0\cdot T}|z_T\rangle + |1\rangle\frac{1}{\sqrt{2^b}}\sum_{T\subseteq S_1}(-1)^{a_1\cdot T}|z_T\rangle\right).$$

We can unitarily transform this to

$$\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle).$$

By Theorem 5.2.2, we can compute a bit $o$ from this such that $\Pr[o = f(a_0, a_1)] = 1/2 + 1/2^{b+1}$. The probability of success is then given by

$$\begin{aligned}
\Pr[o = x_i] &= \Pr[o = f(a_0,a_1)]\cdot\Pr[x_i = f(a_0,a_1)] + \Pr[o \neq f(a_0,a_1)]\cdot\Pr[x_i \neq f(a_0,a_1)]\\
&= \left(\frac{1}{2} + \frac{1}{2^{b+1}}\right)p + \left(\frac{1}{2} - \frac{1}{2^{b+1}}\right)(1-p)\\
&= \frac{1}{2} - \frac{1}{2^{b+1}} + \frac{1}{2^b}p\\
&\geq \frac{1}{2} + \frac{\varepsilon}{2^b}.
\end{aligned}$$

Since no $j$ is queried with probability more than $c/m$ by the classical decoder, the same is true for the quantum decoder. Hence we have constructed a QSC with the appropriate properties. □

### 5.3.3 Improved Lower Bounds for 2-query LDCs over an $\ell$-bit alphabet

Our proof of a lower bound for 2-query LDCs uses the notion of a *quantum random access code*. That is an encoding $x \mapsto \rho_x$ of $n$-bit string $x$ into $m$-qubit states $\rho_x$, such that any bit $x_i$ can be recovered with some probability $p \geq 1/2 + \varepsilon$ from $\rho_x$. For the length of such quantum codes there is a known lower bound [64]:

**Theorem 5.3.3 (Nayak)** *An encoding $x \mapsto \rho_x$ of $n$-bit strings into $m$-qubit states with recovery probability at least $p$ has $m \geq (1 - H(p))n$.*

The main ingredient of our proof is the following lemma, which shows how the query of a QSC gives rise to a quantum random access code. Let $u = \sum_{i=0}^{b} \binom{\ell}{i}$ and define the $\log(u)$-qubit pure states

$$|U(x)_j\rangle = \frac{1}{\sqrt{u}} \sum_{|T| \leq b} (-1)^{T \cdot C(x)_j} |z_T\rangle$$

and the $(\log(m) + \log(u))$-qubit states

$$|U(x)\rangle = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} |j\rangle |U(x)_j\rangle.$$

**Lemma 5.3.4** *Suppose $C : \{0,1\}^n \to (\{0,1\}^\ell)^m$ is a $(1, c, \varepsilon)$-quantum smooth code that uses $b$ bits. Then given one copy of $|U(x)\rangle$, there is a quantum algorithm that outputs 'fail' with probability $1 - 2^{b+1}/(cu)$ with $u = \sum_{i=0}^{b} \binom{\ell}{i}$, but if it succeeds it outputs $x_i$ with probability at least $1/2 + \varepsilon$.*

**Proof.** Let us fix $i \in [n]$. Suppose the quantum decoder of $C$ makes query $|Q_{ir}\rangle$ to indices $j_{0r}$ and $j_{1r}$ with probability $p_r$. Consider the following state

$$|V_i(x)\rangle = \sum_r \sqrt{p_r} |r\rangle \frac{1}{\sqrt{2}} \left( |j_{0r}\rangle |U(x)_{j_{0r}}\rangle + |j_{1r}\rangle |U(x)_{j_{1r}}\rangle \right).$$

We will first show how to obtain $|V_i(x)\rangle$ from $|U(x)\rangle$ with some probability. Rewrite $|V_i(x)\rangle$ to

$$|V_i(x)\rangle = \sum_{j=1}^{m} \alpha_j |\phi_j\rangle |j\rangle |U(x)_j\rangle,$$

where the $\alpha_j$ are nonnegative reals, and $\alpha_j^2 \leq c/(2m)$ because $C$ is a QSC (the $1/2$ comes from the amplitude $1/\sqrt{2}$). Using the unitary map $|0\rangle|j\rangle \mapsto |\phi_j\rangle|j\rangle$, we can obtain $|V_i(x)\rangle$ from the state

$$|V_i'(x)\rangle = \sum_{j=1}^{m} \alpha_j |j\rangle |U(x)_j\rangle.$$

We thus have to show that we can obtain $|V_i'(x)\rangle$ from $|U(x)\rangle$. To this end, define operator

$$M = \sqrt{\frac{2m}{c}} \sum_{j=1}^{m} \alpha_j |j\rangle\langle j| \otimes I$$

and consider a POVM with measurement operators $M^\dagger M$ and $I - M^\dagger M$. These operators are both positive because $\alpha_j^2 \leq c/2m$. Note that, up to normalization, $M|U(x)\rangle = |V_i'(x)\rangle$. The probability that the measurement succeeds (i.e. takes us from $|U(x)\rangle$ to $|V_i'(x)\rangle$) is

$$\langle U(x)|M^\dagger M|U(x)\rangle = \frac{2m}{c}\langle U(x)| \left( \sum_j \alpha_j^2 |j\rangle\langle j| \otimes I \right) |U(x)\rangle = \frac{2}{c}\sum_j \alpha_j^2 = \frac{2}{c}$$

Now given $|V_i(x)\rangle$ we can measure $r$, and then project the last register onto the sets $S_{0r}$ and $S_{1r}$ that we need for $|Q_{ir}\rangle$, by means of the measurement operator

$$|j_{0r}\rangle\langle j_{0r}| \otimes \sum_{T \subseteq S_{0r}} |T\rangle\langle T| + |j_{1r}\rangle\langle j_{1r}| \otimes \sum_{T \subseteq S_{1r}} |T\rangle\langle T|.$$

This measurement succeeds with probability $2^b/u$, but if it succeeds we have the state corresponding to the answer to query $|Q_{ir}\rangle$, from which we can predict $x_i$. Putting everything together, we succeed with probability $(2^b/u) \cdot (2/c)$, and *if* we succeed, we output $x_i$ with probability $1/2 + \varepsilon$. □

We can avoid failures by taking many copies of $|U(x)\rangle$:

**Lemma 5.3.5** *If $C : \{0,1\}^n \to (\{0,1\}^\ell)^m$ is a $(1, c, \varepsilon)$-quantum smooth code, then $|W(x)\rangle = |U(x)\rangle^{\otimes cu/2^{b+1}}$ is a $cu(\log(m) + \log(u))/2^{b+1}$-qubit random access code for $x$ with recovery probability $1/2 + \varepsilon/2$ where $u = \sum_{i=0}^{b} \binom{\ell}{i}$.*

**Proof.** We do the experiment from the previous lemma on each copy of $|U(x)\rangle$ independently. The probability that each experiment fails is $(1-2^{b+1}/(cu))^{cu/2^{b+1}} \leq 1/2$. In that case we output a fair coin flip. If at least one experiment succeeds, we can predict $x_i$ with probability $1/2 + \varepsilon$. This gives success probability at least $\frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \left(\frac{1}{2}\right)^2 = \frac{1}{2} + \frac{\varepsilon}{2}$. □

Armed with these tricks, we can finally prove the lower bound for 2-query smooth codes and LDCs over non-binary alphabets.

**Theorem 5.3.6** *If $C : \{0,1\}^n \to \Sigma^m = (\{0,1\}^\ell)^m$ is a $(2, c, \varepsilon)$-smooth code where the decoder uses only $b$ bits of each answer, then*

$$m \geq 2^{dn - \log(u)}$$

*for* $d = (1 - H(1/2 + \varepsilon/2^{b+1}))2^{b+1}/(cu) = \Theta(\varepsilon^2/(2^b cu))$ *and* $u = \sum_{i=0}^{b} \binom{\ell}{i}$. *In particular,* $m = 2^{\Omega(\varepsilon^2 n/(2^{2\ell}c))}$ *if* $b = \ell$.

**Proof.** Theorem 5.3.2 implies that $C$ is a $(1, c, \varepsilon/2^b)$-quantum smooth code. Lemma 5.3.5 gives us a random access code of $cu(\log(m) + \log(u))/2^{b+1}$ qubits with recovery probability $p = 1/2 + \varepsilon/2^{b+1}$. Finally, the random access code lower bound, Theorem 5.3.3, implies $cu(\log(m) + \log(u))/2^{b+1} \geq (1 - H(p))n$. Rearranging and using that $1 - H(1/2 + \eta) = \Theta(\eta^2)$ gives the result. □

Since a $(2, \delta, \varepsilon)$-LDC is a $(2, 2/\delta, \varepsilon)$-smooth code (Theorem 4.3.2), we obtain the main result:

**Corollary 5.3.7** *If* $C : \{0,1\}^n \to \Sigma^m = (\{0,1\}^\ell)^m$ *is a* $(2, \delta, \varepsilon)$*-locally decodable code, then*

$$m \geq 2^{dn - \log(u)}$$

*for* $d = (1 - H(1/2 + \varepsilon/2^{b+1}))\delta 2^b/u = \Theta(\delta\varepsilon^2/(2^b u))$ *and* $u = \sum_{i=0}^{b} \binom{\ell}{i}$. *In particular,* $m = 2^{\Omega(\delta\varepsilon^2 n/2^{2\ell})}$ *if* $b = \ell$.

In all known non-trivial constructions of LDCs and smooth codes, the decoder outputs the parity of the bits that he is interested in. In this case we can prove a slightly stronger bound.

**Theorem 5.3.8** *If* $C : \{0,1\}^n \to \Sigma^m = (\{0,1\}^\ell)^m$ *is a* $(2, c, \varepsilon)$*-smooth code where the decoder outputs* $f(g(a_{0|S_0}), g(a_{1|S_1}))$*, with* $f : \{0,1\}^2 \to \{0,1\}$ *and* $g : \{0,1\}^b \to \{0,1\}$ *fixed functions, then*

$$m \geq 2^{dn - \log(\ell')}$$

*for* $d = \Omega(\varepsilon^2/(c\ell'))$ *and* $\ell' = \binom{\ell}{b}$.

**Proof.** We can transform $C$ into a smooth code $C' : \{0,1\}^n \to (\{0,1\}^{\ell'})^m$ with $\ell' = \binom{\ell}{b}$ by defining $C'(x)_j$ to be the value of $g$ on all $\binom{\ell}{b}$ possible $b$-subsets of the original $\ell$ bits of $C(x)_j$. Now we are interested in $b' = 1$ bit of each $C'(x)_j$. The result then follows from Theorem 5.3.6. □

## 5.4 Lower Bounds for Private Information Retrieval

### 5.4.1 Lower Bounds for 2-server PIRs that use few bits

Here we derive improved lower bounds for 2-server PIRs from our LDC bounds. We use Lemma 4.4.1 from Goldreich et al. [47, Lemma 7.1] to translate PIR schemes to smooth codes. We then combine this lemma with Theorem 5.3.6 to obtain the following theorem. This slightly improves the lower bound given in [53] and extends it to the case where we only use $b$ bits out of each server reply.

**Theorem 5.4.1** *A classical 2-server* $(1-\eta)$*-secure PIR scheme with* $t$*-bit queries,* $\ell$*-bit answers that uses* $b$ *bits and has recovery probability* $1/2+\varepsilon$ *satisfies*

$$t = \Omega\left(\frac{n(\varepsilon-\eta)^2}{2^b u}\right)$$

*with* $u = \sum_{i=0}^{b}\binom{\ell}{i}$*. In particular, if all bits of the answer are used, then* $t = \Omega(n(\varepsilon-\eta)^2/2^{2\ell})$*.*

**Proof.** Using Lemma 4.4.1 we turn the PIR scheme into a $(2,3,\varepsilon-\eta)$-smooth code $C : \{0,1\}^n \to (\{0,1\}^\ell)^m$ that uses $b$ bits of $\ell$ where $m \leq 6 \cdot 2^t$. From Theorem 5.3.6 we have $m \geq 2^{dn-a}$ with $d = \Theta((\varepsilon-\eta)^2/(2^b u))$. □

If $b$ is fixed, $\varepsilon = 1/2$ and $\eta = 0$, the above bound simplifies to $t = \Omega(n/\ell^b)$, hence

**Corollary 5.4.2** *A 2-server PIR scheme with* $t$*-bit queries and* $\ell$*-bit answers has total communication*

$$C = 2(t+\ell) = \Omega\left(n^{\frac{1}{b+1}}\right).$$

For $b = 1$ this gives $C = \Omega(\sqrt{n})$, which is achieved by the square scheme of Section 3.2.2. For $b = 3$ we get $C = \Omega(n^{1/4})$, which is close to the $C = O(n^{1/3})$ of the cube scheme of Section 3.2.2.

As in Theorem 5.3.8, we can get slightly better bounds for PIR schemes where the user just outputs the parity of $b$ bits from each answer. All known non-trivial PIR schemes have this property.

**Corollary 5.4.3** *If the PIR's user outputs* $f(g(a_{0|S_0}), g(a_{1|S_1}))$*, for fixed* $f$ *and* $g$*, then*

$$t = \Omega\left(\frac{n(\varepsilon-\eta)^2}{\binom{\ell}{b}}\right).$$

## 5.4.2 Weak Lower Bounds for general 2-server PIR

The previous lower bounds on the query length of 2-server PIR schemes were significant only for protocols that use few bits from each answer. Here we slightly improve the best known bound of $4.4\log n$ [53] on the overall communication complexity of 2-server PIR schemes, by combining our Theorem 5.4.1 and Theorem 6 of Katz and Trevisan [52]. We restate their theorem for the PIR setting, assuming for simplicity that $\varepsilon = 1/2$ and $\eta = 0$.

**Theorem 5.4.4 (Katz & Trevisan)** *Every 2-server PIR scheme with* $t$*-bit queries and* $\ell$*-bit answers has*

$$t \geq 2\log\frac{n}{\ell} - O(1).$$

We now prove the following lower bound on the total communication $C = 2(t+\ell)$ of any 2-server PIR scheme with $t$-bit queries and $\ell$-bit answers:

**Theorem 5.4.5** *Every 2-server PIR scheme has total communication*

$$C \geq (5 - o(1)) \log n.$$

**Proof.** We distinguish three cases, depending on the answer length. Let $\delta = \log\log n / \log n$.

**case 1:** $\ell \leq (0.5 - \delta) \log n$. Then from Theorem 5.4.1 we get that $C \geq t = \Omega(n^{2\delta}) = \Omega((\log n)^2)$.

**case 2:** $(0.5 - \delta) \log n < \ell < 2.5 \log n$. Then from Theorem 5.4.4 we have

$$C = 2(t+\ell) > 2\,(2\log(n/(2.5\log n)) - O(1) + (0.5 - \delta)\log n) = (5 - o(1))\log n.$$

**case 3:** $\ell \geq 2.5 \log n$. Then $C = 2(t + \ell) \geq 5 \log n$. □

## 5.5 Quantum PIRs from classical PIR with non-binary answers

Using the tricks employed for LDCs above, we can construct a 2-server *quantum* PIR scheme from a 4-server classical PIR scheme that uses $b$ bits, as follows. The user flips his randomness as in the classical scheme. This fixes queries $q_0,q_1,q_2,q_3$ as well as sets $S_0,S_1,S_2,S_3$ of $b$-bit indices to use from answers $a_0,a_1,a_2,a_3$ respectively. He now picks random permutations $\pi_1,\pi_2,\pi_3$ on the set of all $\binom{\ell}{b}$ $b$-element subsets from an $\ell$-element set, such that $\pi_1(S_0) = S_1$, $\pi_2(S_0) = S_2$ and $\pi_3(S_0) = S_3$. The user then constructs the following quantum state

$$\frac{1}{\sqrt{\binom{\ell}{b}}} \sum_{|T|=b} |T\rangle \left( \frac{1}{\sqrt{2}} (|0\rangle \underbrace{|0\rangle|q_0\rangle|T\rangle}_{\text{Server 1}} \underbrace{|1\rangle|q_1\rangle|\pi_1(T)\rangle}_{\text{Server 2}} + |1\rangle \underbrace{|2\rangle|q_2\rangle|\pi_2(T)\rangle}_{\text{Server 1}} \underbrace{|3\rangle|q_3\rangle|\pi_3(T)\rangle}_{\text{Server 2}}) \right).$$

He keeps the first two registers to himself and sends the rest to the two quantum servers as indicated. Each server now only sees a random mixture over the classical queries and a random $T$. This means that if he were to measure the query, he would only obtain one classical query $q_j$. Thus the privacy of the 4-server scheme carries over to the quantum scheme. Each server tags on $b$ $|0\rangle$-qubits, maps

$$|j\rangle|q_j\rangle|T\rangle|0^b\rangle \mapsto |j\rangle|q_j\rangle|T\rangle|a_{j|T}\rangle$$

and sends everything back. The user now measures the first register, hoping to obtain $T = S_0$. He succeeds with probability $1/\binom{\ell}{b}$. He can then unitarily remove the $S_j$ and $q_j$ to get

$$\frac{1}{\sqrt{2}}(|0\rangle|a_{0|S_0}\rangle|a_{1|S_1}\rangle + |1\rangle|a_{2|S_2}\rangle|a_{3|S_3}\rangle).$$

From this he can compute $f(a_{0|S_0}, a_{1|S_1}, a_{2|S_2}, a_{3|S_3})$ with probability $1/2 + 1/2^{2b+1}$ using Theorem 5.2.2. The expected number of repetitions (in parallel) before success is $\binom{\ell}{b}$. This means that if we can construct 4-server classical PIR schemes where $\binom{\ell}{b}$ is quite small, then we obtain an efficient 2-server quantum PIR scheme. For example, if there exists a 4-server classical PIR scheme with $t, \ell = O(n^{1/8})$, using only $b = 1$ bits from each of the 4 answers, then we obtain a 2-server quantum PIR scheme with an expected number of $O(n^{1/4})$ qubits of communication and recovery probability close to 1. Currently, the best known 2-server quantum PIR communicates $O(n^{3/10})$ qubits [53].

## 5.6 Conclusion and Future Work

In this paper we improved the best known lower bounds on the length of 2-query locally decodable codes and the communication complexity of 2-server private information retrieval schemes. Our bounds are significant whenever the decoder uses only few bits from the two query answers, even if the alphabet (LDC case) or answer length (PIR case) is large. This contrasts with the earlier results of Kerenidis and de Wolf [53], which become trivial even for logarithmic alphabet or answer length, and those of Goldreich et al. [47], which only apply to *linear* schemes.

Still, general lower bounds without constraints on alphabet or answer size completely elude us. Clearly, this is one of the main open questions in this area. Barring that, we could at least improve the dependence on $b$ of our current bounds. For example, a PIR lower bound like $t = \Omega(n/\ell^{\lceil b/2 \rceil})$ might be feasible using some additional quantum tricks. Such a bound for instance implies that the total communication is $\Omega(n^{1/3})$ for $b = 3$, which would show that the Cube scheme of Section 3.2.2 is optimal. Another question is to obtain strong lower bounds for the case of $k \geq 3$ queries or servers. For this case, no superpolynomial lower bounds are known even if the alphabet or answer size is only one bit. Finally, our constructions motivate the search for 4-server classical PIR schemes with fairly large answer length $\ell$, but using very few bits from each answer. As explained in Section 5.5, such schemes would give better 2-server *quantum* PIR schemes.

# Part II

# Anonymous Transmissions

# Chapter 6

# Anonymous Transmissions

## 6.1 Introduction

Primitives to hide the sender and recipient of a transmission have received considerable attention in classical computing. Such protocols allow any member of a group to send and receive messages anonymously, even if all physical transmissions can be monitored.

In the previous part of this text, we have examined the notion of private information retrieval. PIR allows a user to query a database without revealing the requested index to the database. However, the database still learns the identity of the user. This poses a different problem: Consider again the case of a patent database we encountered in Section 3.1. PIR can ensure that the database does not gain any information about the nature of the patent query, but it does not conceal the identity of the user. This may allow a malicious database to conclude that the user has completed some part of his research or gained a new insight, which in itself can be valuable information. It is therefore desirable to hide the identity of the user, as well as the requested index.

Primitives for anonymous transmissions also play an important role in protocols for electronic auctions [76], voting protocols and sending anonymous email [27]. Other applications allow users to access the Internet without revealing their own identity [71], [36] or, in combination with private information retrieval, provide anonymous publishing [39]. Finally, an anonymous broadcast channel which is completely immune to any active attacks, would be a powerful primitive. Alpern and Schneider [3] showed how two parties can use such a channel to perform key-exchange.

### Outline

First we define the notions of anonymity and untraceability. We then take a look at some of the known classical protocols. In particular, we examine the Dining Cryptographers problem [28].

## 6.2 Definitions

In most cryptographic applications, we are interested in ensuring the secrecy of data. Sender and recipient know each other, but are trying to protect their message exchange from prying eyes. Anonymity, however, is the secrecy of identity. Looking at message transmissions in particular, this means that a sender stays *anonymous*, if no one can determine his identity within the set of possible senders. The recipient of the message himself should not learn the sender's identity either. During the following discussions, we will refer to the $n$ members of a group $V$ of senders and recipients as *participants* or *players*. Let $P(s_k)$ be all of the communication during a run of the protocol $P$ where player $k$ sends message $s$. Similarly, let $P(r_k)$ denote a run of the protocol $P$ where player $k$ receives message $r$. Furthermore, let $\Pr(i = k|P(s_k))$ be the probability that player $i$ was the sender given $P(s_k)$, where the probabilities are taken over the protocols behavior. We define for the case of unconditional security

**Definition 6.2.1** *A protocol P allows a sender to be* anonymous*, if for an adversary* $\forall i, j, k \in V : \Pr(i = k|P(s_k)) = \Pr(j = k|P(s_k))$.

Similarly, if we now let $\Pr(i = k|P(r_k))$ denote the probability that player $i$ is the receiver given $P(r_k)$, we can define

**Definition 6.2.2** *A protocol P allows a recipient to be* anonymous*, if for an adversary* $\forall i, j, k \in V : \Pr(i = k|P(r_k)) = \Pr(j = k|P(r_k))$.

We can now define the notion of anonymous message transmissions:

**Definition 6.2.3** Anonymous transmissions *are transmissions of data, where sender and recipient are anonymous.*

Note that protocols to hide the sender and recipient do not protect message contents. It is implicitly assumed that the message data itself does not contain any compromising information. Furthermore, we will make use of the following expressions.

**Definition 6.2.4** *If a participant is anonymous after interactions with the other participants by sending messages, we speak of* untraceability *of a participant.*

**Definition 6.2.5** *A protocol that allows the sender of a message to remain anonymous achieves* sender untraceability*. If the recipient of a message stays anonymous, we speak of* recipient untraceability*.*

Several participants may try to work together in order to trace the sender of a certain message. As in the case of PIR, we call such a collaboration a *collusion* of participants. Here, however, an adversary is additionally allowed to monitor all physical transmissions. This means he can follow the path of all messages, reading and changing them as desired. An attacker that is restricted to observing the network is called *passive*. If he can also change messages passing over the network, we refer to him as an *active* attacker. All known protocols for anonymous transmissions achieving information theoretic security need a reliable broadcast channel. This type of channel is also required by our quantum protocol for anonymous transmissions presented in the next chapter. Achieving broadcast in the presence of dishonest parties turns out to be an extremely difficult problem, even though it seems deceptively easy at first sight. Lamport, Shostak and Pease [57] considered this problem under the name of the Byzantine Generals Problem. More formally, reliable broadcast is defined [40] as

**Definition 6.2.6 (FGMR)** *A protocol among $n$ players such that one distinct player $s$ (the sender) holds an input value $x_s \in D$ (for some finite domain $D$) and all players eventually decide on an output value in $D$ is said to achieve* broadcast *(or* Byzantine Agreement*) if the protocol guarantees that all correct players decide on the same output value $y \in D$, and that $y = x_s$ whenever the sender is correct.*

Lamport et al. [57] showed that perfectly secure broadcast is achievable if and only if less than a third of the players are corrupted. Fitzi and Maurer [41] showed that if broadcast is possible among each subset of three participants, then global broadcast is possible if and only if less than half of the players are corrupted. If broadcast is possible in a stage preceding the actual protocol, Pfitzmann and Waidner [70] showed that broadcast can later be achieved if only a single participant is honest. A slightly weaker notion is that of detectable, or weak broadcast [40]. This variant is sufficient for our purposes.

**Definition 6.2.7 (FGMR)** *A protocol among $n$ players such that one distinct player $s$ (the sender) holds an input value $x_s \in D$ and all players eventually decide on an output value $D \cup \{\perp\}$ (with $\perp \neq D$) is said to achieve* weak broadcast *(or* detectable broadcast*) if the protocol guarantees the following conditions:*

- *If a correct player decides on some value $y \in D$ then all correct players decide on a value in $\{y, \perp\}$.*
- *If the sender is correct then all correct players decide on $y = x_s$.*

We can interpret $\perp$ as “abort”. Thus in detectable broadcast, either all honest players agree on same value $y$ or they abort the protocol. Analog to the this notion of broadcast in message passing networks, we also define:

**Definition 6.2.8** *A protocol achieves* anonymous broadcast *if it achieves broadcast where the sender remains anonymous.*

## 6.3 Known Protocols

A considerable number of schemes have been suggested for anonymous transmissions. They can be divided into three basic classes: First of all there are protocols which employ a trusted third party. In practice this takes the form of a trusted proxy server [6], [49], forwarding messages while masking the identity of the original sender.

Secondly, there are protocols using a chain of forwarding servers. Most notably these are protocols based on so-called mixing techniques introduced by Chaum [27], such as Webmixes [21] and ISDN-Mixes [68]. Here messages are passed through a number of untrusted proxies which reorder the messages; hence the name MixNet. The goal of this reordering is to ensure an observer cannot match in- and outgoing messages and thus cannot track specific messages on their way through the network. Public Key Encryption is then used between the user and the different forwarding servers to hide the contents of a message. Several implemented systems, such as Mixmaster [63], PipeNet [36] and Onion Routing [78] employ layered encryption: the user successively encrypts the message with the public keys of all forwarding servers in the chain. Each server then "peels off" one layer, by decrypting the received data with its own secret key, to determine the next hop to pass the message to. The Crowds [71] system takes another approach. Here each player acts as a forwarding server himself. He either sends the message directly to the destination, or passes it on to another forwarding server with a certain probability. The aim is to make any sender within the group appear equally probable for an outside observer. Various other protocols using forwarding techniques are known. Since our focus lies on the final class of protocols, we restrict ourselves to this brief introduction. More information can be found in the papers by Goldberg and Wagner [46], [45]. An extensive overview of known techniques and protocols is also given in the PhD thesis of D. Martin [61, Chapter 2 and 3].

Finally there is a class of protocols which does not make use of either a trusted third party or other forms of message forwarding. These protocols are based on shared resources and communication among all the participating parties, which makes them difficult to implement on a large scale in practice. Unlike the other classes of protocols, however, they do not suffer from malicious forwarding servers. The most well-known of these is the Dining Cryptographers protocol introduced by Chaum [28] of which we give a brief overview below. A network based on this protocol is also referred to as a DC-net. Small scale practical implementations of this protocol are known [61]. In contrast to schemes based on public key cryptography, a DC-net provides unconditional security.

### 6.3.1 Dining Cryptographers

Since our protocol in the next chapter was inspired by the Dining Cryptographers Problem, we briefly review the original dinner table scenario: A group of cryptographers is assembled around a dinner table in their favorite restaurant. They have already made arrangements with the restaurant owner to pay the bill anonymously. However, they are curious whether one of them is paying or if perhaps the NSA pays. They agree on the following protocol: Each of them secretly flips a coin behind the menu with both of his neighbors at the table and adds the outcomes of both coin flips. He then loudly announces the outcome of the sum. The person paying, however, adds a 1 to his outcome before the announcement. All the dinner guests can now calculate the total sum of the announced bits. The sum equals 1 if one of the cryptographers is paying and 0 otherwise.

Let's see how this works out for cryptographers $A$, $B$ and $C$ in Figure 6.1. Let $r_{i,j}$ denote the outcome of the coin flip between $i$ and $j$. In our example, $B$ wants to anonymously send bit $b = 1$. Everyone computes the sum of their private shared random bits, where $B$ adds $b$ to his sum. They then broadcast their outcomes. Anyone, including $A$, can now compute the total sum of the announcements which gives $a_A \oplus a_B \oplus a_C = b$, since every shared random bit occurs exactly twice in the sum. Anonymity is preserved, since $A$ does not know the value of $r_{B,C}$ and can therefore not determine whether $B$ or $C$ was the sender. The same holds for $C$ by a similar argument.

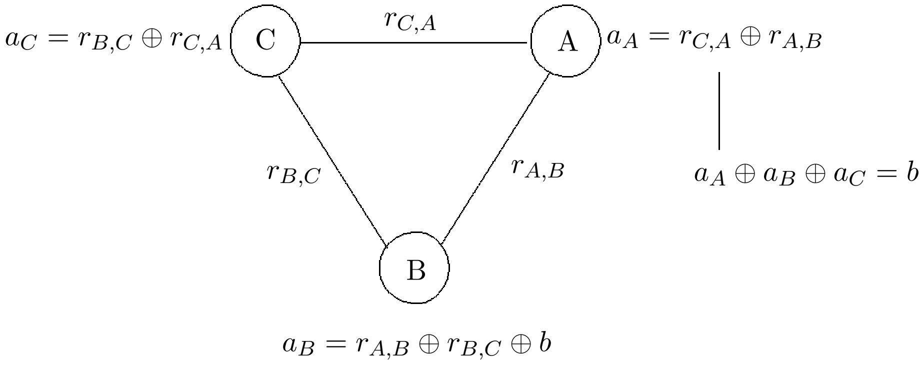
$a_C = r_{B,C} \oplus r_{C,A}$
C
$r_{C,A}$
A
$a_A = r_{C,A} \oplus r_{A,B}$
$r_{B,C}$
$r_{A,B}$
$a_A \oplus a_B \oplus a_C = b$
B
$a_B = r_{A,B} \oplus r_{B,C} \oplus b$

Figure 6.1: Dining Cryptographers Problem

### Protocol

This scenario can be generalized by viewing the participants as nodes in an undirected graph. We will from now on use the notions of "nodes in a key-sharing graph" and "participants" interchangeably. Similarly, we will speak of edges, keys and private shared random bits to denote the same object.

**Definition 6.3.1** *The undirected graph $G = (V, E)$ is called the* key-sharing graph *of a DC-net if*

- *Each node in $V$ represents exactly one of the participants.*

- *There is an edge between two nodes $i$ and $j$ if and only if $i$ and $j$ privately share one random bit $r_{i,j}$.*

For example, in the round-table scenario depicted in Figure 6.1, the set of nodes is $V = \{A, B, C\}$ and the edges are given by $E = \{\{A, B\}, \{B, C\}, \{C, A\}\}$. Each node has degree 2, as each participant shares a random bit with each of his neighbors.

We can summarize this protocol below. Let $n$ be the total number of nodes. Furthermore $d \geq 2$ denotes the degree of the nodes in the key-sharing graph. Let $r_{i,j} = r_{j,i}$ denote the private shared random bit associated with the edge connecting nodes $i$ and $j$. We furthermore assume that participants do not collude.

**Protocol 2: Dining Cryptographers**

```
1: The sending node $s$ computes $a_s = b \oplus \bigoplus_{\{s,k\} \in E}(r_{s,k})$
2: Each other node $j$ computes $a_j = \bigoplus_{\{j,k\} \in E}(r_{j,k})$
3: Each node $j$ broadcasts $a_j$
4: Each recipient now calculates the outcome $b = \bigoplus_{j=1}^{n}(a_j)$
```

Sender untraceability is achieved, since each participant remains uncertain about at least one random bit in each announcement. A formal proof can be found in [28]. Clearly recipient untraceability is already achieved by the broadcast. Everybody can compute the resulting sum, so any one of them could act as the true receiving party. For the announcement we require $n$ uses of a detectable broadcast channel.

Boykin [23] considered a quantum protocol to transmit classical information anonymously where the participants share EPR pairs instead. The necessary private shared random bits are then distilled from the shared EPR pairs.

An important requirement of the DC-net model is a reliable broadcast channel. If one of the participants is able to send a different value to each neighbor, he can potentially subvert the protocol. For example one of the participants could intentionally lie to some of the other participants during the announcement phase. This will simply disrupt the protocol, if messages are flowing in only one direction. If, however, the recipient reacts to messages by sending a reply, this can allow such an active attacker to identify the recipient later on [82]: Consider for example the graph above with participants $A$, $B$ and $C$ and suppose that the receiver simply echoes back any messages he receives. Suppose $A$ is malicious and tries to find out whether the real receiver is $B$ or $C$. Of course, both $B$ and $C$ receive the broadcast outcome, however, we assume that only one of them actually intends to receive messages. For example, one of them could be a server waiting to process requests. In the first round $A$ will make sure he sends a different bit to $B$ and $C$. This means that $B$ and $C$ will compute a different value for outcome $b$. $A$, on the other hand, receives all the correct messages and can compute the proper $b$. In the next round

$B$ or $C$ will reply. $A$ can now simply observe the outcome to determine who was the receiving party in the previous round.

Noting that it is sufficient if all honest parties either receive the same broadcast value or abort, a DC-net is also secure using a detectable broadcast by Maurer et al. [40]. A number of additions have also been made to the original protocol to remove the dependence on a reliable broadcast channel altogether [82], [84]. However, they offer only computational security.

### Collisions and Disruptions

The above protocol works if only a single person tries to send. If multiple participants try to send at the same time, collisions occur. Chaum [28] presents a simple collision detection protocol based on ALOHA [79]: If a participant notices that the outcome of the broadcast is different from what he intended to send, he simply resends after waiting a random number of rounds. Other approaches for collision detection and slot reservation for DC type channels have also been considered by Pfitzmann [69] and Waidner et al. [82].

Another disadvantage of the protocol lies in its vulnerability to denial of service attacks, because a malicious participant can jam the channel by repeated sending of information. Classically various approaches offering computational security have been considered to thwart these attacks [83], [28]. An overview about the problems associated with regulating channel access in a DC-net is also given in the PhD thesis of A. Pfitzmann [69].

### Collusions of Participants

What happens, if some of the $n$ participants decide to work together to trace back the sender? The effect of collusions depends highly on the connectivity of the key-sharing graph $G$. Unless $G$ is fully connected, malicious nodes can partition the graph into multiple disjoint sets. This allows them to compute the sum of announcements in each such partition individually. By comparing the individual sums with the total sum of the network, they can determine which partition of the graph a transmission originated in. To see how this works, consider the following example depicted in Figure 6.2.

Recall that each edge represents one bit of private shared randomness. If $M$ and $E$ now pool their resources and work together, they can compute the announcements of both partitions of the graph separately. In the example of Figure 6.2 they compute the outcome of the right partition $a_B \oplus a_C \oplus a_D \oplus r_{MB} \oplus r_{EC} = b_1$ and the left partition $a_A \oplus r_{AM} \oplus r_{AE} = b_2$. They then compute $b_1 \oplus b_2 = b$ to obtain the overall outcome. From $b_1 = b$ they can now immediately conclude that $A$ sent bit $b$. Thus by partitioning the key-sharing graph, they can restrict the identity of the sender to a much smaller set of participants.

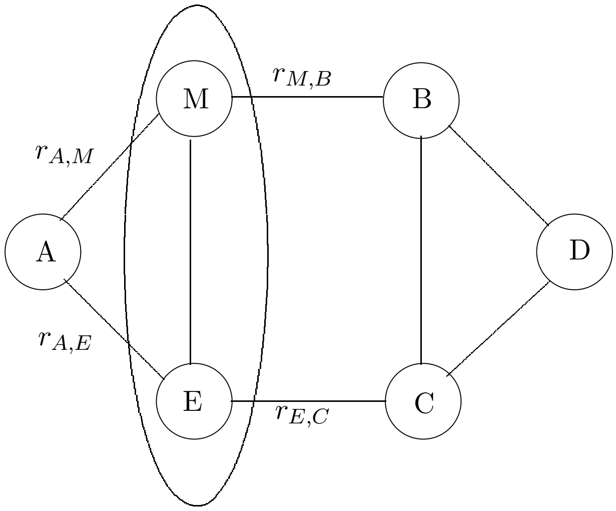

Figure 6.2: Collusions of participants (here M and E) can partition the key-sharing graph

The number of malicious collaborators we can tolerate is thus directly dependent on the form of $G$. Only a fully connected key-sharing graph can withstand partitioning if up to $n-2$ participants are working together.

### Resources

The fundamental resource used by the DC-net is private shared randomness between the participants. How many of these shared random bits do we need?

We first note that for any protocol $P$ that achieves sender untraceability, where the only resource used by the $n$ participating parties is pairwise private shared randomness, the form of the key-sharing graph $G = (V, E)$ is important. More formally we can say that

**Lemma 6.3.2** *In any protocol $P$ to achieve anonymous broadcast among $n$ players, where the only resource available to the participants is pairwise private shared randomness, a broadcast channel and classical communication, a particular collusion of $t$ participants can break the sender's anonymity, if the corresponding collection of $t$ nodes partitions the key-sharing graph $G = (V, E)$.*

**Proof.** $t$ colluding nodes partition the key-sharing graph into $s$ disjoint sets of nodes $\{S_1, \ldots, S_s\} = N$. As there is no edge connecting any of these sets, these sets do not share any private randomness. Now suppose that anonymous broadcast is still possible. Let $S_i, S_j \in N$ be two arbitrary sets of nodes in $N$. Any node $i \in S_i$ can now establish a key with a node $j \in S_j$ using anonymous broadcast [3]:

Nodes $i$ and $j$ each generate $n$ random bits: $r_i^1, \ldots, r_i^n$ and $r_j^1, \ldots, r_j^n$. Node $i$ now announces $n$ messages of the form: "Bit $b_k$ is $r_i^k$" for $1 \leq k \leq n$ using the protocol for anonymous broadcast. Likewise, node $j$ announces "Bit $b_k$ is $r_j^k$" for $1 \leq k \leq n$. Nodes $i$ and $j$ now discard all bits for which $r_i^k = r_j^k$ and use the remaining bits as a key. Note that an adversary can only learn whether $b_k = r_i^k$ or $b_k = r_j^k$ if the two announcements are the same. If $r_i^k \neq r_j^k$, the adversary does not learn who has which bit.

However, since the view of an adversary in our model includes all communication, members of different sets in $N$ cannot establish a shared key to communicate with each other securely using only classical communication [20]. Thus the sender's anonymity can be broken if the graph can be partitioned. □

Furthermore, note that each player $j$ needs to privately share one bit of randomness with at least two other participants or his anonymity can be compromised. We can phrase this in terms of the key-sharing graph as

**Corollary 6.3.3** *Each node $j \in V$ of the key-sharing graph $G = (V, E)$, used by a protocol $P$ for anonymous transmissions, where the only resource available to the $n$ participants is pairwise private shared randomness, a broadcast channel and classical communication, must have degree $d \geq 2$.*

**Proof.** Suppose on the contrary, that an arbitrary node $j$ has degree 1: it has only one outgoing edge to another node $k$. Clearly, node $k$ can partition the key-sharing graph into two disjoint sets $S_1 = \{j\}$ and $S_2 = V \setminus \{j, k\}$. By Lemma 6.3.2, node $k$ can then break $j$'s anonymity. □

**Corollary 6.3.4** *Any protocol $P$ that achieves anonymous broadcast, where no players collude and the only resource available to the $n$ participants is pairwise private shared randomness, a broadcast channel and classical communication, needs at least $n$ bits of pairwise private shared randomness.*

**Proof.** Consider again the key-sharing graph $G = (V, E)$. Suppose on the contrary, that only $k < n$ bits of private shared randomness are used. Then there must be at least one node of degree 1 in the graph. Thus by Corollary 6.3.3 at least $n$ bits of private shared randomness are necessary. □

**Corollary 6.3.5** *Any protocol $P$ that achieves anonymous broadcast and is resistant against collusions of $t < n - 1$ malicious players, where the only resources available to the $n$ participants are pairwise private shared randomness, a broadcast channel and classical communication, needs at least $n(n-1)$ bits of pairwise private shared randomness.*

**Proof.** Again consider the key-sharing graph $G$. Suppose on the contrary, that only $k < n(n-1)$ bits of private shared randomness are used. However, then there are only $k < n(n-1)$ edges in a graph of $n$ nodes. Then $G$ is not fully connected and there is a set of $t = n-2$ colluding nodes which can partition the key-sharing graph. By Lemma 6.3.2, they can then break the sender's anonymity. Thus $n(n-1)$ bits of pairwise private shared random bits are necessary to tolerate up to $t < n-1$ colluding players. □

These observations will become useful in the next chapter, when we consider a different resource which can be used to solve the problem: shared entanglement.

### 6.3.2 Secure Function Evaluation

Chaum and Crépeau [29] later also considered the problem of secure function evaluation (SFE), also called Secure Multi-Party Computation. In this setting, $n$ participants want to compute a function $f$ of their inputs $b_1, b_2, \ldots, b_n$, such that each party learns the value of $f(b_1, \ldots, b_n)$ without revealing more about their input bit then can be inferred from this value itself. The DC-net thus forms a special case of secure function evaluation where $f$ is the parity function. The input of player $i$ is $b_i = 0$, if he doesn't wish to send, and $b_i = 1$ otherwise. Secure Multi-Party computation is without doubt the more interesting problem. Nevertheless, we consider this special case later on, since our protocol is intuitively much simpler then protocols for the general case. Anonymous transmission itself, however, can be a separate primitive which may be implemented by fundamentally different approaches, as practical implementations show. Furthermore, more efficient protocols may be possible in this restricted scenario [81].

Crépeau et al. [35] recently also considered the case of secure function evaluation using quantum states as inputs. Clearly in the quantum setting there is no notion of broadcast as in classical computing.

## 6.4 Summary

We have examined the notion of anonymous transmission, which allows message exchange where sender and recipient remain anonymous. There are numerous classical protocols for this problem, most of which offer computational security. We have also taken a closer look at the Dining Cryptographers protocol, which is unconditionally secure given a reliable broadcast channel. Secure Function Evaluation or Secure Multi-Party Communication allows implementation of a DC-net without detection of disruptors. We note that all classical protocols allow reconstruction of the sender, once all privately held keys are collected by a single party. For example in a DC-net, we can always trace back the sender later on, once we acquire knowledge of all keys $r_{i,j}$ in the network.

# Chapter 7

# A Quantum Protocol for Anonymous Transmissions with Perfect Repudiation

## 7.1 Introduction

Here we introduce new quantum protocols to send and receive classical and quantum bits anonymously. We first consider a protocol that allows $n$ participants to send and receive one bit of classical information anonymously using one shared entangled state $|\Psi\rangle = (|0^n\rangle + |1^n\rangle)/\sqrt{2}$ and $n$ uses of a detectable broadcast channel as defined in Section 6.2. Like the original DC-net, the quantum protocol achieves anonymous broadcast. Our protocol is secure against collusions of up to $n-2$ participants. This means that the collaborators cannot learn anything more by working together and pooling their resources. The main advantage of our protocol over the classical protocols is that it prevents later reconstruction of the sender and thus achieves perfect repudiation: None of the participants are able to prove the identity of a sender at a later point in time. This contrasts with all known classical protocols we examined in the last chapter.

Clearly we would also like to transmit qubits anonymously. We first use our protocol to allow two anonymous parties to establish a shared EPR pair. Finally, we use this form of anonymous entanglement to hide the sender and recipient of an arbitrary qubit. These protocols use the same resource of shared entangled states $|\Psi\rangle$ and a detectable broadcast channel.

If multiple participants want to send simultaneously, collisions can occur. Using the same fundamental resource of shared states, we create a $\lceil \log n \rceil$-round quantum protocol to detect all collisions in advance. Each of these rounds uses one entangled state $|\Psi\rangle$.

Our protocol provides an example of a scenario where we can trade in $O(n^2)$ private pairwise shared random bits used classically for one $n$-qubit shared entangled state $|\Psi\rangle$. Interestingly, this is a state equally shared by all participants, whereas the classical random bits are shared privately between each pair of participants. This shows that, for certain applications, globally shared entanglement may not only be traded in for shared randomness, but also for private shared randomness among the participants.

## 7.2 Preliminaries

### 7.2.1 Quantum Resources

The fundamental resource used in our protocol are $n$-party shared entangled states of the form

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle).$$

By "shared" we mean that each of the $n$ participants holds exactly one qubit of $|\Psi\rangle$. They could have obtained these states at an earlier meeting or distribute and test them later on.

The key observation used in our protocols is the fact that phase flips and rotations applied by the individual participants have the same effect on the global state no matter who applied them. Consider for example the phase flip defined by

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

If player number $i$ applies this transformation to his state, the global transformation is

$$U_i = I^{\otimes i-1} \otimes Z \otimes I^{n-i}.$$

We now have $\forall i, U_i|\Psi\rangle = (|0^n\rangle - |1^n\rangle)/\sqrt{2}$. Note that this effect takes place "instantaneously" and no communication is necessary.

### 7.2.2 Definitions and Model

We use the general notions of anonymity as defined in Section 6.2. Our model assumes that the $n$ participants have access to the following resources

1. An $n$-qubit shared entangled state $|\Psi\rangle = (|0^n\rangle + |1^n\rangle)/\sqrt{2}$.

2. A channel providing at least detectable broadcast.

We require our protocol to satisfy the following conditions

1. If all participants are honest, then the protocol achieves sender and recipient untraceability.

2. If one or more players are malicious, they can at most disrupt the protocol, but are not able to learn anything about the sender's or recipient's identity.

Here, we call a participant "honest" if he follows the protocol, and "dishonest" otherwise as in Section 2.1. If he is trying to disrupt the channel, we will also call him "malicious". We require our protocols to be unconditionally secure, as opposed to protocols where the security is based on computational constraints.

## 7.3 Sending Classical Bits

First of all we present a protocol to transmit a classical bit $b$ anonymously, if all $n$ participants share an $n$-qubit entangled state $|\Psi\rangle$. For now we assume that only one person wants to transmit in each round of the protocol and deal with the case of multiple senders later on.

### 7.3.1 Protocol

Let's return to the original dinner table scenario described in Section 6.3.1. Suppose Alice, one of the dinner guests, wishes to broadcast a bit $b$ anonymously. For this she uses the following protocol:

**Protocol 3: ANON($b$)**
Prerequisite: Shared state $(|0^n\rangle + |1^n\rangle)/\sqrt{2}$

**1:** Alice applies a phase flip $Z$ to her part of the state if $b = 1$ and does nothing otherwise.

**2:** Each participant (incl. Alice):

- Applies a Hadamard transform to his qubit.
- Measures his qubit in the computational basis.
- Broadcasts his measurement result.
- Counts the total number of 1's, $k$, in the $n$ measurement outcomes.
- If $k$ is even, he concludes $b = 0$, otherwise $b = 1$.

### 7.3.2 Analysis

**Correctness**

Since Alice applies the phase flip $Z$ depending on the value of the bit $b$ she wishes to transmit, the participants obtain the state $(|0^n\rangle + |1^n\rangle)/\sqrt{2}$ if $b = 0$ and $(|0^n\rangle - |1^n\rangle)/\sqrt{2}$ if $b = 1$. By tracing out the other players' part of the state as in Section 1.3.5, we can see that no player can determine on his own whether the phase of the global state has changed. We therefore require the players to first apply a Hadamard transform to their qubit. This changes the global state such that we get a superposition of all strings $x \in \{0,1\}^n$ with an even number of 1's for no phase flip and an odd number of 1's if a phase flip has been applied:

$$\begin{aligned} H^{\otimes n}\left(\frac{1}{\sqrt{2}}(|0^n\rangle + (-1)^b|1^n\rangle)\right) &= \frac{1}{\sqrt{2^{n+1}}}\left(\sum_{x\in\{0,1\}^n}|x\rangle + (-1)^b\sum_{x\in\{0,1\}^n}(-1)^{|x|}|x\rangle\right) \\ &= \frac{1}{\sqrt{2^{n+1}}}\sum_{x\in\{0,1\}^n}(1+(-1)^{b\oplus|x|})|x\rangle \end{aligned}$$

This means that we expect an even number of 1's if $b = 0$ and an odd number of 1's if $b = 1$. The players now measure their part of the state, and announce their measurement outcome. This allows each participant to compute the number of 1's in the global outcome, and thus the value of $b$. Broadcasting all measurement results requires $n$ uses of a broadcast channel.

Note that if we had multiple senders, this protocol computes the parity of all their input bits, without revealing their individual inputs. Thus, ANON also performs secure function evaluation on $n$ inputs, if the function is parity.

### Security

As we noticed in Section 7.2, the resulting global state is independent of the identity of the person applying the phase flip. A phase flip is applied locally, so no transmissions are necessary to change the global state. Therefore an adversary monitoring the network cannot learn anything about the identity of the sender.

Suppose that $t$ participants decide to work together to determine the identity of the sender. Here we can tolerate up to $n-2$ colluding participants: by pooling their states and measuring, they will still remain uncertain as to who applies the phase flip. To achieve the same in a DC-net, we require a fully-connected key-sharing graph as noted in Section 6.3.1. Our protocol is therefore secure against all passive attackers.

The fact that collusions do not affect the security of our protocol, also makes it secure against the so-called predecessor attack described in [86]. In this attack, the adversary is allowed to track a stream of communication over multiple rounds of the protocol. Among the existing protocols investigated by these authors, only a fully connected DC-net is resilient to this attack.

The most interesting property of our quantum protocol is that Alice can later always deny she performed the phase flip. There is no record of her activity as in the case of private shared random bits. Thus even if we have a collusion of $n-1$ participants, which together know all keys $r_{ij}$, they cannot prove Alice sent $b$ and not one of their own. This contrasts with all known classical protocols, where once all keys become known to a single entity, it can always determine the sender of a message. An outside entity, who has the ability to force all participants to give up their keys, can trace Alice at any later point in time. The same is true in computationally secure schemes, where public key encryption is used.

Our protocol thus provides irreconstructible untraceability. Whereas this is stronger than the classical protocols, it also makes our protocol more prone to disruptors. Unlike in the classical scenario, we cannot employ mechanisms such as traps suggested by Chaum [28], and Waidner and Pfitzmann [84], to trace back disruptors. If one of our participants is determined to disrupt the channel by, for example, always applying a phase flip himself, we are never able to find and exclude him from the network.

As indicated in our model description above, we assume that the participants use at least a detectable broadcast channel. This is to ensure all honest participants obtain the same value for an announcement. We have already described why this is necessary to defeat active attackers in the context of a DC-net in Section 6.3.1. The same argument also holds for our protocol.

## 7.4 Sending Qubits

We now extend the above protocol, to allow anonymous transmissions of qubits.

### 7.4.1 Anonymous Entanglement

#### Definition

To achieve this, we first allow two parties to create entanglement between them without learning each other's identity. We define

**Definition 7.4.1** *If two anonymous parties A and B share entanglement, we speak of* anonymous entanglement (AE)*.*

**Definition 7.4.2** *If two parties A and B share entanglement, where one of them is anonymous, we speak of* one-sided anonymous entanglement (AE)*.*

As we saw in Section 1.4.4, we can use shared entanglement together with classical communication to transmit information. An important property of quantum teleportation using entangled states is that value of the classical bits which are transmitted is independent of the state to be teleported. Anonymous entanglement together with broadcast thus forms a virtual channel between two parties who do not know who is sitting at the other end. This allows for easy sender and recipient untraceability. We can create such channels directly by exchanging quantum information instead of, for example, changing the configuration of a hardware switch as in the classical scenario. Classically, such a virtual channel would have to be emulated by exchanging a key anonymously. Below, we give one example of using anonymous entanglement. We think that the resource of anonymous entanglement, however, could play a role in a variety of other protocols as well.

### Protocol

Since in our scenario all participants already share states $|\Psi\rangle$, we use the same resource to establish anonymous entanglement for transmitting information. More general protocols are certainly possible. Here the sending and receiving party can establish entanglement between them anonymously using an idea presented in the context of quantum broadcast [5]. We now assume that there are exactly two parties, sender (Alice) and receiver (Bob), among the $n$ participants interested in sharing an EPR pair. If more parties are interested, they can use a form of collision detection described later.

**Protocol 4: AE**
Prerequisite: Shared states $(|0^n\rangle + |1^n\rangle)/\sqrt{2}$

**1:** Alice and Bob don't do anything to their qubit.
All participants run ANON twice, where Alice's and Bob's input is always 0.

**2:** Participant $j$ (of the $n-2$ remaining participants):

- Applies a Hadamard transform to his qubit.
- Measures his qubit in the computational basis with outcome $m$.
- Runs ANON($m$):
  All participants run ANON, where the $j$'s input is $m$.

**3:** Bob applies a phase flip $Z$ for every measurement result of 1 he receives.

At the end of the protocol Alice and Bob share the state $(|00\rangle+|11\rangle)/\sqrt{2}$. Because ANON($b$) from Section 7.3, does not reveal the sender of a bit $b$, no one obtains any information about the identity of Alice. Likewise, all participants learn the outcomes of all measurement results, so any of them could act as Bob does. Therefore the "real" Bob also remains hidden.

### 7.4.2 Sending Qubits using Anonymous Entanglement

Alice and Bob can now use their shared EPR pair to send a qubit $|\phi\rangle$ via quantum teleportation [65], which we described in Section 1.4.4. We can thus construct the following simple protocol to anonymously transmit a qubit, which uses this EPR pair along with the above protocol ANON.

**Protocol 5: ANONQ($|\phi\rangle$)**
Prerequisite: Shared states $(|0^n\rangle + |1^n\rangle)/\sqrt{2}$

**1:** The participants run AE:
Alice and Bob then share $|\Psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ anonymously.

**2:** Alice uses the quantum teleportation circuit with input $|\phi\rangle$ and her half of $|\Psi\rangle$:

- Obtains measurement outcomes $m_0, m_1$.
- Runs ANON($m_0$) and ANON($m_1$):
  All participants perform ANON twice, where Alice's inputs are $m_0$ and $m_1$.

**3:** Bob applies the transformation described by $m_0, m_1$ on his part of the EPR pair to retrieve $|\phi\rangle$.

As discussed earlier, AE and ANON($b$) do not leak any information about Alice or Bob. Since no additional information is revealed during the teleportation step, it follows that ANONQ($|\phi\rangle$) does not leak any information either.

## 7.5 Dealing with multiple senders

So far we have assumed that only a single person is transmitting in any one round. In reality, many users may wish to transmit simultaneously, leading to collisions. A user can easily detect a collision, if it changes the classical outcome of the broadcast. Depending on the application this may be sufficient. However, it may be desirable to detect all types of collisions, even those that lead to the same broadcast outcome. This could, for example, be important if we are also interested in the number of outcomes of a certain type.

When transmitting quantum states, collisions are not so easy to detect. If more than two parties wish to create entanglement between them using AE, they will disrupt the transmission of a qubit using ANONQ later on. To solve this problem, the participants can employ the same collision detection mechanism that was used to regulate the transmission of classical bits.

### 7.5.1 Collision Detection

There is a simple quantum protocol to detect all kinds of collisions, provided that no user tries to actively disrupt the protocol. We use the same resource, namely shared entangled states $|\Psi\rangle$. The important part of this protocol is that it preserves the property of our original protocol, namely that the sender cannot be reconstructed at a later point in time.

### Protocol

Before each round of communication, the $n$ participants run a $\lceil \log n \rceil$-round test to check, whether a collision would occur. For this they require $\lceil \log n \rceil$ additional states of the form $|\Psi\rangle = (|0^n\rangle + |1^n\rangle)/\sqrt{2}$. Each such state is rotated before the start of the collision detection protocol. Define

$$U_j = R_z(-\pi/2^j) \otimes I^{\otimes n-1} = e^{i\frac{\pi}{2^{j+1}}} \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/2^j} \end{pmatrix} \otimes I^{\otimes n-1}$$

and map the $j$th state to $|t_j\rangle = U_j|\Psi\rangle$. This could for example be done by a dedicated participant or be determined upon distribution of the entangled states $|\Psi\rangle$.

```
Protocol 6: Collision Detection
Prerequisite: $\lceil \log n \rceil$ states $|\Psi\rangle = (|0^n\rangle + |1^n\rangle)/\sqrt{2}$

1: A designated player prepares $\lceil \log n \rceil$ states by rotations:
   For $1 \leq j \leq \lceil \log n \rceil$, he applies $R_z(-\pi/2^j)$ to his part of one $|\Psi\rangle$ to create $|t_j\rangle$.
2: In round $1 \leq j \leq \lceil \log n \rceil$ each of the $n$ players
      - Applies $R_z(\pi/2^j)$ to his part of the state $|t_j\rangle$, if he wants to transmit.
      - Applies a Hadamard transform to his part of the state.
      - Measures in the computational basis.
      - Announces his measurement result to all other players.
      - Counts the total number of 1's, $k_j$, in the measurement results.
      - If $k_j$ is odd, concludes a collision has occurred and the protocol ends.
3: If all $k_j$ are even, exactly 1 player wants to send.
```

### Analysis

**Correctness** Let's first take an informal look, why this works. In round $j$ with $0 \leq j \leq \lceil \log n \rceil$, each user who wishes to send applies a rotation described by $R_z(\pi/2^j)$ to his part of the state. Note that if exactly one user tries to send, this simply rotates the global state back to the original state $|\Psi\rangle = (|0^n\rangle + |1^n\rangle)/\sqrt{2}$. If $k > 1$ users try to send, we can detect the collision in round $j$ such that $k = 2^j m + 1$ where $m \in \mathbb{N}$ is odd: First $|t_j\rangle$ is rotated back to $|\Psi\rangle$ by the first of the $k$ senders. The state is then rotated further by an angle of $(\pi/2^j) \cdot 2^j m = m\pi$. But

$$R_z(m\pi) = e^{-i\frac{m\pi}{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{m\pi} \end{pmatrix} = \pm i \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

applied to $|\Psi\rangle$ gives $|\Psi'\rangle = \pm i(|0^n\rangle - |1^n\rangle)/\sqrt{2}$, where we can ignore the global phase. The users now all apply a Hadamard transform to their part of the state again, measure and broadcast their measurement results to all participants. As before, they can distinguish between $|\Psi\rangle$ and $|\Psi'\rangle$, by counting the number of 1's in the outcome. If the number of users who want to transmit in round $j$ is not of the form $2^j m + 1$, the players may observe an even or odd number of 1's, thus they only detect a collision with a certain probability.

The important observation is that in $\lceil \log n \rceil$ rounds, the players will obtain $|\Psi'\rangle$ at least once, if more than one user wants to send, which they can detect. If no phase flip has been observed in all $\lceil \log n \rceil$ rounds of the collision detection protocol, the players can be sure there is exactly one sender. The key to understanding this part of the protocol is the following observation:

**Theorem 7.5.1** *For any integer $1 < k \leq n$, there exist unique integers $m$ and $j$, with $m$ odd and $0 \leq j \leq \lceil \log n \rceil$, such that $k = 2^j m + 1$.*

**Proof.** By the fundamental theorem of arithmetic we can write $k - 1 = 2^j m$ for unique $j, m \in \mathbb{N}$ where $m$ is odd. We have $j \leq \lceil \log n \rceil$, since $1 < k \leq n$. Thus $k = 2^j m + 1$. □

**Corollary 7.5.2** *$\lceil \log n \rceil$ rounds, using one state $(|0^n\rangle + |1^n\rangle)/\sqrt{2}$ each, are sufficient to detect $2 \leq k \leq n$ senders within a group of $n$ participants.*

**Proof.** Using Theorem 7.5.1 we can write $k = 2^j m + 1$ with $0 \leq j \leq \lceil \log n \rceil$. In round $j$ the final state will be $R_z((2^j m) \cdot (\pi/2^j))|\Psi\rangle = R_z(m\pi)|\Psi\rangle = \pm i(|0^n\rangle - |1^n\rangle)/\sqrt{2}$, which the participants can detect. □

The player who wanted to send then does so in a single round of anonymous broadcast following this protocol. The protocol has the side effect that we can also recognize the case of no senders. In the classical DC-net protocol, we cannot distinguish between the sending of a 0 and no transmission. Here we will observe a collision in the first round, if no senders are present. In the following we will speak of one application of the collision detection protocol to denote all $\lceil \log n \rceil$ rounds necessary to perform collision detection.

**Security** Furthermore, this protocol preserves anonymity. We cannot determine which player applied a certain rotation, since rotations by any player would result in the same outcome. Thus this method does not leak any information concerning the identity of a sender. All it reveals is a rough indication of the number of senders present, which is what we were looking for. As before, this protocol is secure against collusions against participants. Most importantly, it also preserves the repudiation property of our original protocol: A participant is always able to deny sending later on.

### 7.5.2 Slot Reservation

The simplest way to deal with collisions is for the user to wait a random number of rounds, before attempting to resend the bit. This method was suggested by Chaum [28] and is generally known as ALOHA [79]. Unfortunately this approach is rather wasteful, if many parties try to send simultaneously. Instead, we can use a reservation map technique similar to what was suggested by Pfitzmann et al. [69]. For this we use $n$ applications of collision detection (of $\lceil \log n \rceil$ rounds each) to reserve the following $n$ slots: Each participant picks a round $1 \leq r \leq n$ to send in, which he announces during collision detection. If no one else wanted to send, the player then sends in round $r$ following the reservation period. Otherwise he waits until the next execution of slot reservation. Figure 7.1 illustrates this concept. Since all participants are aware of the collision, the players can restrict the sending period to only those rounds where no collision occurred earlier on. This restriction will ensure that no sending rounds are performed unnecessarily.

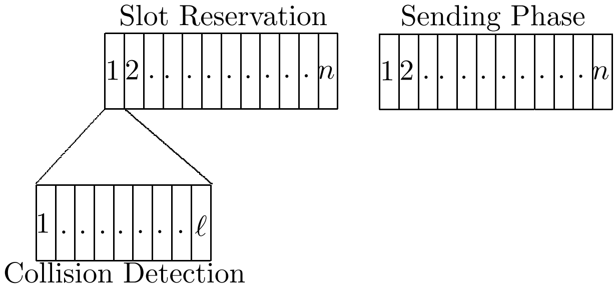

Figure 7.1: Slot Reservation with Collision Detection ($\ell = \log n$)

Note that this does not protect us from disruptors who try to send in the rounds following the reservation period. This allows them to successfully disrupt the protocol. However, the same is possible by measuring all qubits in the computational basis, which destroys the resource required for transmissions. We are therefore not concerned with this problem.

### 7.5.3 Using Collision Detection for Establishing Anonymous Entanglement

The protocol to establish anonymous entanglement relies on the fact that only two parties refrain from measuring. We therefore require some coordination between the two parties. In our scenario, we can make use of the same collision detection protocol as we used to send classical bits. A simple approach would be to first run collision detection protocol to determine the sender. The sender again expresses his interest in indicating that he wants to send by employing rotations. We then perform another application of collision detection for the receiver.

### 7.5.4 Classical Collision Detection

Classically, we can perform a similar kind of collision detection using $n^2 \log n$ bits of private shared randomness. Instead of one bit of shared randomness, we now let each two nodes $i$ and $j$ pool $\log n$ bits of shared randomness to obtain a $\log n$-bit key $r_{i,j}$. Computations are now performed modulo $n$. We furthermore assume that all parties have a unique (non private) id $i \in [n]$. As was also suggested by Pfitzmann et al. [83] player $i$ now computes

$$a_j = b_j + \sum_{k=1}^{n} \operatorname{sign}(k-i) r_{i,k}$$

and announces the result. During collision detection, player $j$ just sets $b_j = 1$ if he intends to send. By calculating the total sum of all announcements, all players learn how many participants intend to send. If there is only one sender, the total sum will be 1. He then proceeds with sending in a single round of the original DC-net protocol following collision detection. The same slot reservation techniques can be employed.

Since we are again dealing with classical information, we can trace back the sender at any later point in time, once we obtain all keys $r_{i,j}$.

## 7.6 Resource Tradeoffs

By Theorem 6.3.5, one round in a DC-net requires at least $\Omega(n^2)$ bits of private shared randomness to be secure against collusions of any $n-2$ participants. Our protocol for transmitting classical bits, however, makes use of an $n$-qubit shared entangled state $|\Psi\rangle$ to achieve the same. Interestingly, the entangled state is shared equally by all participants, whereas each private key is known to only two parties. The global entangled state cannot be used itself to create these private keys: By measuring the state in the computational basis, the $n$ players could distill one bit of globally shared randomness. This common randomness cannot be used to establish the private keys required by the DC-net protocol, since we allow all transmissions to be monitored.

Our protocol for anonymous classical broadcast, requires each player to hold exactly 1 qubit of a shared entangled state and to transmit 1 bit of information using a broadcast channel. Without any form of shared resource the participants cannot carry out a protocol for unconditionally secure anonymous transmissions. Likewise, each player needs to communicate at least 1 bit of classical or quantum information in each round. Since faster than light communication is impossible, he cannot apply any modifications to his local quantum state which would transmit information to the other players without additional transmissions. If, however, only the sending player would transmit a bit, the other players could immediately be excluded from the set of possible senders and the real sender cannot remain anonymous. Thus each sender needs to send at least 1 bit. We therefore believe that our protocol is close to optimal.

## 7.7 Conclusions and Future Work

We have presented a protocol for achieving anonymous transmissions using shared quantum states together with a classical broadcast channel. The main feature of this protocol is that, unlike all known classical protocols, it prevents later reconstruction of the sender. This indicates that shared entangled states are very well suited to achieve anonymity. Perhaps similar techniques could also play an important role in other protocols where later reconstruction should be prevented.

Our protocol is a first attempt at providing anonymous transmissions with this particular property. More efficient protocols may be possible. Perhaps a different form of quantum resource gives an additional advantage. However, we believe that our protocol is close to optimal for the given resources. We have also not considered the possibility of allowing quantum communication between the participants, which could be required by more efficient protocols. It is also open whether a better form of collision detection and protection against malicious disruptors is possible. Using shared entangled states, it is always possible for a malicious user to measure his qubit in the computational basis to make further transmissions impossible.

Another issue arises in terms of state distribution. So far, we have simply assumed that the participants share a certain quantum resource. In reality, however, this resource would need to be established before it can be used. This would almost certainly require quantum communication between the participants in order to distribute and test the necessary states. The original DC-net protocol suffers from a similar problem: Unless the participants are supplied with private shared randomness initially or let a trusted third party distribute it, they have to establish these keys later on. This, however, is impossible to do from scratch using only classical channels [20]. Quantum states on the other hand have the interesting property that the participants can create and test the states among themselves, instead of relying on a trusted third party.

# Appendix A

# Linear Algebra

Quantum mechanics expressed using the Dirac notation makes heavy use of linear algebra. We therefore provide a quick overview over the most important concepts necessary for the understanding of this text. We assume the reader is familiar with basic concepts, such as matrix multiplication and addition. A more detailed overview of linear algebra can be found in any text book such as [7].

## A.1 Vector Space

A $d$-dimensional *vector space* $V$ is a set of all $d$-dimensional vectors closed under vector addition and scalar multiplication. We use $V = \mathbb{C}^d$ to denote the $d$-dimensional complex vector space. In general $V$ is a vector space over a field $F$, if the following conditions hold for all elements $X, Y, Z \in V$ and any scalars $r, s \in F$:

- Commutativity: $X + Y = Y + X$.
- Associativity of vector addition: $(X + Y) + Z = X + (Y + Z)$.
- There exists an additive identity: $X + 0 = 0 + X = X$.
- There exists an additive inverse: $X + (-X) = 0$.
- Associativity of scalar multiplication: $(rs)X = r(sX)$.
- Distributivity of scalar sums: $(r + s)X = rX + sX$.
- Distributivity of vector sums: $r(X + Y) = rX + sY$.
- Scalar multiplication identity: $1X = X$.

## A.2 Basic Notions

A set of vectors $v_1, \ldots, v_d \in V$ is *linearly independent* if $\sum_{i=1}^{d} a_i v_i = 0$ has only the trivial solution in scalars $a_i$. A *basis* of a $d$-dimensional vector space $V$ is a set of linearly independent vectors $v_1, \ldots, v_d \in V$, the *basis vectors*, such that any vector $u \in V$ can be written as a linear combination of basis vectors.

The *transpose* of a matrix $A$ is written as $A^T$ and given by $A^T_{ij} = A_{ji}$, where $A_ij$ denotes the entry of the matrix $A$ at column $i$ and row $j$. Similarly, the conjugate transpose $A^\dagger$ of $A$ is of the form $A^\dagger_{ij} = A^*_{ji}$. The inverse of a square matrix is denoted as $A^{-1}$. We have that $AA^{-1} = A^{-1}A = I$, where $I$ is the identity matrix with $I_{ij} = \delta_{ij}$.

If there exists a vector $v \in V$ with $v \neq 0$ such that $Av = \lambda v$, we say that $v$ is an *eigenvector* of $A$ and the scalar $\lambda$ the corresponding *eigenvalue.* The set of all eigenvectors corresponding to an eigvenvalue together with the 0 vector forms a subspace of the underlying vector space and is referred to as *eigenspace.*

The *inner product* of two vectors $u, v \in V$ is given by $\langle u|v\rangle = u^* v = \sum_i u_i^* v_i$. The vector *norm* is given by $\parallel v \parallel = \sqrt{\langle v|v\rangle}$. Two vectors $u, v \in V$ such that $\langle u|v\rangle = 0$ are *orthogonal.* If, in addition, $\parallel u \parallel = \parallel v \parallel = 1$ then they are also called *orthonormal.* A *Hilbert space* is defined as a vector space $V$ with an inner product inducing a distance metric.

## A.3 Unitary Matrix

A square matrix $U$ is *unitary* if $U^{-1} = U^\dagger$. The following expressions are equivalent

- $U$ is unitary
- $U$ preserves inner product: $\forall v \forall w, \langle Uv|Uw\rangle = \langle v|w\rangle$
- $U$ preserves norm: $\forall v, \parallel Uv \parallel = \parallel v \parallel$

## A.4 Hermitian Matrix

A square matrix $H$ is *hermitian* if is it *self-adjoint*, that is $H = H^\dagger$. In terms of matrix elements of $H$ that means that $H_{ij} = H^*_{ji}$.

## A.5 Tensor Product

The *tensor product* of an $m \times n$-matrix $A$ and an $m' \times n'$ matrix $B$ is given by the $mm' \times nn'$-matrix

$$A \otimes B = \begin{pmatrix} A_{11}B & \ldots & A_{1n}B \\ A_{21}B & \ldots & A_{2n}B \\ & \ddots & \\ A_{n1}B & \ldots & A_{nn}B \end{pmatrix}.$$

The following properties hold for all $A, B \in V$ and scalars $c \in F$:

- $c(A \otimes B) = (cA) \otimes B = A \otimes (cB)$
- $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$
- $A \otimes (B + C) = (A \otimes B) + (A \otimes C)$
- $A \otimes (B \otimes C) = (A \otimes B) \otimes C$
- $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

The tensor product is also defined for two vector spaces $V$ and $V'$. In particular, if the basis of the $d$-dimensional vector space $V$ is given by $\{v_1, \ldots, v_d\}$ and the basis of the $d'$-dimensional vector space $V'$ is given by $\{v'_1, \ldots, v'_{d'}\}$, then $V \otimes V'$ denotes the $d \cdot d'$-dimensional vector space $W$ with basis $\{v_i \otimes v'_j | 1 \leq i \leq d, 1 \leq j \leq d'\}$. Applying linear operators $A$ to $V$ and $B$ to $V'$ corresponds to applying $A \otimes B$ to $W$.

## A.6 Trace

The trace of a matrix $A$ is given by the sum of its diagonal entries $tr(A) = \sum_i A_{ii}$. Note that

- $tr(A + B) = tr(A) + tr(B)$
- $tr(AB) = tr(BA)$
- $tr(A)$ is the sum of the eigenvalues of $A$

In particular, when calculating the trace of a density matrix from Section 1.3.5, note that

- $tr(\rho) = tr(\sum_i p_i |\psi_i\rangle\langle\psi_i|) = \sum_i p_i tr(|\psi_i\rangle\langle\psi_i|) = \sum_i p_i \langle\psi_i|\psi_i\rangle$

# List of Figures

# List of Tables

# List of Symbols

| | | |
|---|---|---|
| $A^{-1}$ | Inverse of the matrix $A$ | 7 |
| $A^\dagger$ | Conjugate transpose of the matrix $A$ | 5 |
| $A^T$ | Transpose of the matrix $A$ | 7 |
| $[n]$ | The set of numbers up to $n$: $\{1, \ldots, n\}$ | 14 |
| $[n]^d$ | The cartesian product of $d$ instances of $[n]$ | 26 |
| $a_{\vert S}$ | The string $a$ restricted to the indices in set $S \subseteq [n]$ | 48 |
| $S \subseteq [n]$ | The $n$-bit string $S = S_1, \ldots, S_n$, where $i \in S$ iff the $i$th bit $S_i = 1$ | 27 |
| $e_i$ | The $n$-bit string corresponding to the singleton set $S = \{i\}$ | 26 |
| $a \vert b$ | $a$ divides $b$ | 30 |
| $\delta_{ij}$ | Delta function: $\delta_ij = 1$ if $i = j$ and $\delta_ij = 0$ otherwise | 80 |

# Bibliography

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39(1):195–203, 1989.

[2] A.Beimel and Y. Stahl. Robust information-theoretic private information retrieval. In *Proceedings of the 3rd conference on security in Communications networks*, pages 326–341, 2002.

[3] B. Alpern and F.B. Schneider. Key exchange using 'keyless cryptography'. *Information Processing Letters*, 16:79–1, 1983.

[4] A. Ambainis. Upper bound on communication complexity of private information retrieval. In *Proceedings of the 24th ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 401–407, 1997.

[5] A. Ambainis, H. Buhrman, H. Roehrig, and Y. Dodis. Multiparty quantum coin flipping. quant-ph/0304112, 16 Apr 2003.

[6] Anonymizer. Anonymizing proxy. http://www.anonymizer.com.

[7] G. Arfken and H. Weber. *Mathematical Methods for Physicists.* Harcourt Academic Press, 5, international edition edition, 2001.

[8] D. Asonov. Private information retrieval. an overview and current trends. In *GI Jahrestagung (2)*, pages 889–894, 2001.

[9] D. Asonov and J. Freytag. Almost optimal private information retrieval. In *Proceedings of Privacy Enhancing Technologies, Second International Workshop (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 209–223, 2002. Earlier version as Technical Report HUB-IB-156, Humboldt University Berlin.

[10] D. Asonov and J. Freytag. Repudiative information retrieval. In *Proceedings of ACM Workshop on Privacy in the Electronic Society (WPES2002)*, pages 32–40. ACM Press, 2002.

[11] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of 23rd ACM STOC*, pages 21–31, 1991.

[12] L. Babai, P. G. Kimmel, and S.V. Lokam. Simultaneous messages vs. communication. In *Proceedings of 12th Annual Symposium on Theoretical Aspects of Computer Science (STACS'2000)*, volume 900 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 1995.

[13] D. Beaver. Commodity-based cryptography (extended abstract). In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 446–455. ACM Press, 1997.

[14] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *Proceedings of 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS'90)*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 1990.

[15] R. Beigel, L. Fortnow, and W. Gasarch. Nearly tight bounds for private information retrieval systems. Technical Report 2002-L001N, NEC Laboratories America, October 2002.

[16] A. Beimel and Y. Ishai. Information-theoretic private information retrieval: A unified construction. In *Proceedings of 28th ICALP*, pages 912–926, 2001. Longer version on ECCC.

[17] A. Beimel, Y. Ishai, and E. Kushilevitz. General constructions for information-theoretic private information retrieval. Submitted for publication, 2003.

[18] A. Beimel, Y. Ishai, E. Kushilevitz, and J. Raymond. Breaking the $O(n^{1/(2k-1)})$ barrier for information-theoretic Private Information Retrieval. In *Proceedings of 43rd IEEE FOCS*, pages 261–270, 2002.

[19] A. Beimel, Y. Ishai, and T. Malkin. Reducing the servers computation in private information retrieval: PIR with preprocessing. *Lecture Notes in Computer Science*, 1880:56–74, 2000.

[20] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, 1984.

[21] O. Berthold, H. Federrath, and S. Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. *Lecture Notes in Computer Science*, 2009:115–129, 2001.

[22] C. Blundo, P. D'Arco, and A. De Santis. A t-private k-database private information retrieval scheme. *International Journal of Information Security*, 1(1):64–68, 2001.

[23] P. Boykin. *Information Security and Quantum Mechanics: Security of Quantum Protocols.* PhD thesis, University of California, Los Angeles, 2002.

[24] G. Brassard, C. Crépeau, and J.-M. Robert. All-or-nothing disclosure of secrets. In *Proceeding of Advances in Cryptology - Crypto'86*, number 263 in Lecture Notes in Computer Science, pages 234–238. Springer-Verlag, 1986.

[25] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Proceedings of Eurocrypt'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414. Springer, 1999.

[26] Y. Chang. Single database private information retrieval with logarithmic communication. Cryptology ePrint Archive, Report 2004/036, 2004. http://eprint.iacr.org/2004/036/.

[27] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[28] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[29] D. Chaum, C. Crépeau, and I. Damgard. Multiparty unconditionally secure protocols. In *Proceedings of 20th ACM STOC*, pages 11–19, 1988.

[30] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proceedings of the 32th ACM Sym. on Theory of Computing*, 2000.

[31] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords. http://www.cs.technion.ac.il/ gilboa/perk0405.ps, 1998.

[32] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. Later version in J. ACM 45, without the referenced item, 1995.

[33] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998. Earlier version in FOCS'95.

[34] I. Chuang and L. Vandersypen. NMR techniques for quantum control and computation, 2004. quant-ph/0404064.

[35] C. Crépeau, D. Gottesman, and A. Smith. Secure multiparty quantum computation. In *Proceedings of 34th ACM STOC*, 2002.

[36] W. Dei. Pipenet. http://www.eskimo.com/w̃eidai/pipenet.txt.

[37] D. Deutsch. Quantum theory, the Church-Turing principle, and the universal quantum Turing machine. In *Proceedings of the Royal Society of London*, volume A400, pages 97–117, 1985.

[38] G. Di-Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for database private information retrieval (extended abstract). In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 91–100. ACM Press, 1998.

[39] R. Dingledine. The free haven project: Design and deployment of an anonymous secure data haven. Master's thesis, Massachusetts Institute for Technology, 2000.

[40] M. Fitzi, N. Gisin, U. Maurer, and O. von Rotz. Unconditional byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In *Proceedings of Eurocrypt '02*, volume 2332 of *Lecture Notes in Computer Science*, pages 482–501, 2002.

[41] M. Fitzi and U. Maurer. From partial consistency to global broadcast. In *Proceedings of 32th STOC*, pages 494–503, 2000.

[42] W. Gasarch. A survey on private information retrieval. *The Bulletin of the EATCS*, 82:72–107, 2004.

[43] Y. Gertner, S. Goldwasser, and T. Malkin. A random server model for private information retrieval or how to achieve information theoretic PIR avoiding database replication. *Lecture Notes in Computer Science*, 1518:200–217, 1998.

[44] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes, 1998.

[45] I. Goldberg. Privacy-enhancing technologies for the internet, ii: Five years later. In *Proceedings of Privacy Enhancing Technologies, Second International Workshop (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 1–12, 2002.

[46] I. Goldberg, D. Wagner, and E. Brewer. Privacy-enhancing technologies for the internet. In *Proceedings of 42nd IEEE Spring COMPCON*, 1997. http://now.cs.berkeley.edu/ daw/papers/privacy-compcon97.ps.

[47] O. Goldreich, H. Karloff, L. Schulman, and L. Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *Proceedings of 17th IEEE Conference on Computational Complexity*, pages 175–183, 2002. Also on ECCC.

[48] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[49] J. Helsingius. Email anonymizing server: anon.penet.fi, 1996.

[50] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[51] T. Itoh. Efficient private information retrieval. *IEICE Trans. Fundamentals*, ES82-A(1):11–20, 1999.

[52] J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of 32nd ACM STOC*, pages 80–86, 2000.

[53] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *Journal of Computer and Systems Sciences*, 2004. Earlier version in STOC'03. quant-ph/0208062.

[54] I. Kerenidis and R. de Wolf. Quantum symmetrically-private information retrieval. *Information Processing Letters*, 90(3):109–114, 2004. quant-ph/0307076.

[55] E. Kushilevitz and R. Ostrovsky. Single-database computationally private information retrieval. In *Proceedings of 38th IEEE FOCS*, pages 364–373, 1997.

[56] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *Advances in Cryptology (EUROCRYPT-2000)*, volume 1807 of *Lecture Notes in Computer Science*, pages 104–121. Springer, 2000.

[57] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4:382–401, 1982.

[58] H. Lipmaa. Computationally private information retrieval with quasilogarithmic total communication. Cryptology ePrint Archive, Report 2004/063, 2004. http://eprint.iacr.org/2004/063/.

[59] T. Malkin. *A Study of Secure Database Access and General Two-Party Computation.* PhD thesis, Massachusetts Institute for Technology, 2000.

[60] E. Mann. Private access to distributed information. Master's thesis, Technion - Israel Institute of Technology, Haifa, 1998.

[61] D. Martin. *Local Anonymity in the Internet.* PhD thesis, Boston University, 1999.

[62] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[63] MixMaster. Implementation of a remailer. http://mixmaster.sourceforge.net/.

[64] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proceedings of 40th IEEE FOCS*, pages 369–376, 1999. quant-ph/9904093.

[65] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[66] K. Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *Proceedings of 6th RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 39–50, 2002.

[67] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[68] A. Pfitzmann. How to implement isdns without user observability - some remarks. Technical report, Universitaet Karlsruhe, 1985.

[69] A. Pfitzmann. *Dienstintegrierende Kommunikationsnetze mit teilnehmerueberpruefbarem Datenschutz.* PhD thesis, Fakultaet fuer Informatik, Universitaet Karlsruhe, 1989.

[70] B. Pfitzmann and M. Waidner. Unconditional byzantine agreement for any number of faulty processors. In *Symposium on Theoretical Aspects of Computer Science*, pages 339–350, 1992.

[71] M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[72] K.H. Rosen. *Elementary Number Theory and Its Applications.* Addison-Wesley, 2000.

[73] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[74] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027.

[75] S. W. Smith and D. Safford. Practical private information retrieval with secure coprocessors, 2000. Technical report, IBM T.J. Watson Research Center.

[76] F. Stajano and R. J. Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding*, pages 434–447, 1999.

[77] J. P. Stern. A new and efficient all or nothing disclosure of secrets protocol. In *Advances in Cryptology–ASIACRYPT'98*, number 1514 in Lecture Notes in Computer Science, pages 357–371. Springer-Verlag, 1998.

[78] P F Syverson, D M Goldschlag, and M G Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, 1997.

[79] A. S. Tanenbaum. *Computer Networks, 3rd edition.* Prentice-Hall, 1996.

[80] MagicQ Technologies. http://www.magicqtech.com.

[81] L. von Ahn, A. Bortz, and N. J. Hopper. k-anonymous message transmission. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 122–130. ACM Press, 2003.

[82] M. Waidner. Unconditional sender and recipient untraceability in spite of active attacks. *Lecture Notes in Computer Science*, 434:302–319, 1990.

[83] M. Waidner and B. Pfitzmann. Unconditional sender and recipient untraceability in spite of active attacks - some remarks. Technical report, Universitaet Karlsruhe, 1989.

[84] M. Waidner and B. Pfitzmann. The dining cryptographers in the disco: unconditional sender and recipient untraceability with computationally secure serviceability. In *Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*, page 690. Springer-Verlag New York, Inc., 1990.

[85] S. Wiesner. Conjugate coding. *Sigact News*, 15(1), 1983.

[86] M. Wright, M. Adler, B. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. Technical report, University of Massachusetts, Amherst., April 2001.

[87] A. Yamamura and T. Saito. Private information retrieval based on subgroup membership problem. In *Proceedings of the 6th Australasian Conf., ACISP 2001*, 2001.

# Abstract

Quantum Computation not only allows for more efficient local computations, but also has far reaching effects on multi-party protocols. In this thesis, we investigate two cryptographic primitives for privacy protection: private information retrieval and anonymous transmissions.

The goal of private information retrieval (PIR) is to allow a user to retrieve any item from a database while preventing the database from learning the identity of the requested entry. We prove new lower bounds for the communication complexity of unconditionally secure classical private information retrieval using a novel quantum trick. In particular we show that for a 2 server PIR with an $n$-bit database and $t$-bit queries, where the user needs $b$ bits from each of the two $\ell$-bit answers satisfies

$$t = \Omega\left(\frac{n}{2^b \sum_{i=0}^{b} \binom{a}{i}}\right).$$

Our result implies that several known PIR schemes are close to optimal. Closely related to the problem of private information retrieval are locally decodable codes (LDC). These are error-correcting codes that allow efficient decoding of individual bits from the codeword, without having to read all of it. This is particularly useful in applications where we wish to encode a large chunk of data, but are only interested in extracting small pieces at a time. Imagine for example we want to encode an entire book, but want to retrieve only a single page. We show that a 2 query LDC encoding $n$-bit strings over an $\ell$-bit alphabet, where the decoder uses only $b$ bits of each queried position, needs code length

$$m = \exp\left(\Omega\left(\frac{n}{2^b \sum_{i=0}^{b} \binom{a}{i}}\right)\right).$$

These results generalize those of Goldreich et al. [47], who proved roughly the same bounds for only *linear* LDCs and PIRs. Like earlier work by Kerenidis and de Wolf [53], our classical lower bounds are proved using quantum computational techniques. The new trick used here is a tight analysis of how well a 2-input function can be computed from a quantum superposition of both inputs.

We also study the problem of anonymous transmissions. In this setting members of a group of participants want to send and receive data, without revealing their

identity to any other participant or an outside observer. We present a quantum protocol for sending and receiving classical bits anonymously, which is resistant to collusions of participants and, unlike all known classical protocols, prevents later reconstruction of the sender. It appears that entangled quantum states are uniquely suited for anonymous transmissions. We then extend this protocol to provide sender and recipient untraceability for qubits as well. In the process we also introduce a new primitive called anonymous entanglement, which may be useful for many other protocols. Our protocol furthermore provides an example where $O(k^2)$ pairwise private shared random bits can be replaced by a $k$-qubit shared entangled state. This is an interesting tradeoff, as the $k$-qubit entangled state is equally shared by everyone, whereas the classical random bits are shared only by each pair of participants and are unknown to the rest.

# Index