Correct and Private Reductions Among Oblivious Transfers

by

Claude Crépeau

B.Sc. Mathématiques-Informatique (Université de Montréal),
M.Sc. Informatique (Université de Montréal)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1990

Signature of Author ....
Department of Electrical Engineering and Computer Science
November 26, 1989

Certified by ....
Silvio Micali
Associate Professor
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Departmental Committee on Graduate Students

Correct and Private Reductions Among Oblivious Transfers

by

Claude Crépeau

Submitted to the Department of Electrical Engineering and Computer Science
on November 26, 1989, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Roughly speaking a protocol is a way of exchanging messages so that each party gains some specified information. Correctness and privacy are fundamental constraints to the notion of a secure protocol. Essentially, correctness guarantees that the information the parties learn in executing a protocol is what was specified. Privacy guarantees that they cannot learn more than that.

We investigate correctness and privacy for the primitive notion of oblivious transfer protocols. As introduced by Rabin, in an oblivious transfer a party $\mathcal{S}$ (the sender) owning a secret message m discloses this message to another party $\mathcal{R}$ (the receiver) with probability $\frac{1}{2}$. $\mathcal{S}$ does not find out whether $\mathcal{R}$ did get the message or not. A different version of this notion (1-out-of-2-oblivious transfer) was introduced by Even, Goldreich and Lempel. Traditionally, correctness and privacy were guaranteed assuming that the parties had bounded computational resources and that some appropriate mathematical problems were intractable.

We formalize these and other notions of oblivious transfer in abstract setting, without assuming any computational restrictions on the participants. We also formalize the notion of a reduction, that is a method by which one form of oblivious transfer can be achieved under the assumption that another form exists abstractly. We then show correct and private reductions between various notions of oblivious transfer. Some of these notions arise naturally in a physical scenario.

Thesis Supervisor: Silvio Micali
Title: Associate Professor

Acknowledgments

First of all I would like to thank Silvio Micali, my supervisor. I met Silvio at a short course on cryptography in Amsterdam, in the fall of 1985. I was impressed both by the content of his lecture and by his lecturing style. His ability to mix intuition and formalism makes him a great lecturer as well as a great researcher. Without him the results of this thesis would have never been formalized in such a crisp way. He taught me how to turn intuitions into proofs, ideas into definitions, results into theorems. I wish to thank him especially for making me a better mathematician, that can deal with formalism as well as intuitive notions. I thank him also for the many comments he did on my work and suggestions to improve it. My stay at M.I.T. has been successful because of Silvio's class where I learned the ongoing research in our field.

My deepest thanks go to Gilles Brassard. Gilles has been the driving force behind my career and the inspiration of my research. I wish to thank him in particular for leading me to the field of cryptography. I also wish to thank him for proofreading chapters of this thesis. His help was very valuable.

I also would like to thank Joe Kilian for the work we did together. Joe is definitely one of the most talented people I have met in my life. I thank him also for his help in writing this thesis, for helping me with my English and for his support. I thank Ray Hirschfeld for his help proofreading my thesis and for refusing to answer questions about UNIX that had nothing to do with my thesis.

I wish to express my gratitude to Shafi Goldwasser and Ron Rivest for their contribution as members of my defense committee.

My gratitude goes as well to David Chaum and the Center for Mathematics and Computer Science (Amsterdam), Ernie Brickell and Bell Communication Research, Cynthia Dwork and I.B.M. Corporation, Ivan Damgård and Peter Landrock from Århus University for inviting me and providing me with their research facilities, personal contributions and advice. I have gained lots of professional experience through the help of these people. My thesis is the result of research lead in part in all of these institutions.

I thank all my friends for their support.

My last thanks go to my sponsors: Conseil de Recherche en Sciences Naturelles et Génie (Canada) and Fonds pour la Formation des Chercheurs et l'Avancement de la Recherche (Québec). Their financial support have been more than appreciated during my studies at M.I.T..

Contents

List of Figures

Chapter 1

Introduction

The history of modern cryptography can be broken in two main parts: information based cryptography and computational complexity based cryptography. These two approaches were the steps that changed cryptography from an art to a science.

1.1 Information Based Cryptography

In the early years of this century, cryptography was only concerned with protecting the privacy of communications. Information theory was the first tool introduced to understand cryptography. The main idea behind information based cryptography is that two parties can share a common secret piece of information and use it later to exchange a secret message. Shannon showed that the only way to securely communicate information of size n involves prior exchange of a common secret key of size n. Any system not respecting this fact would necessarily be insecure. This condition was in fact satisfied by Vernam's one-time pad invented in 1917. Up to that time the existing systems were based on very little theory. With a one-time pad, to communicate a message of n bits, two parties who have exchanged a random key of n bits can XOR the message with the key in order to obtain a cipher text. The result is a perfectly random string that is meaningful only to a party knowing the secret key.

In some sense the result of Shannon was bad news for cryptographers: it meant that considerable amount of information would have to be shared secretly in order to later transfer information securely. This led Diffie and Helmann [DH76] to reconsider the cryptographic model. In Shannon's model the parties involved, including the enemy trying to understand secret conversations, are assumed to have arbitrary computational power. Therefore secure systems in the sense of Shannon do not provide information at all about the messages transmitted. Maybe this condition could be relaxed.

1.2 Computational Complexity Based Cryptography

The right direction to look was computational complexity. Some computational problems have resisted the effort of generations of mathematicians looking for efficient algorithms to solve them. These problems are considered *hard* to solve. Entities with bounded computational ability are asymptotically incapable of solving these problems within their computational bounds.

This is the idea behind computational complexity based cryptography. The security of the crypto-system relies on the fact that some computational problem has a history of being hard. In some sense, this is what cryptographers have done from the beginning but computational complexity is an appropriate way to formalize it. The first application of this idea was a secret key exchange system of Diffie and Helmann [DH76]. Their protocol enables two parties to publicly exchange a secret key that no other party can understand, unless that party can solve a very difficult number theoretic problem.

The rational behind using a model where parties are computationally bounded is that it more closely mirrors reality, and gives more possibilities than the traditional information based model. For instance, if we take the amount of secret information exchanged a priori as a measure of cost for a crypto-system, it is possible to achieve much more efficient systems in the computational model.

Diffie and Helmann's idea led (shortly after it appeared) to the notion of public key cryptography: systems in which anybody can encrypt messages based on public information, but in which only the legitimate receiver can decrypt. Implementations of this abstraction were soon suggested by Rivest, Shamir and Adleman [RSA78] and Merkle and Helmann [MH78]. Since the birth of public key cryptography much of the research in cryptology has drifted in that direction. Public key cryptography seem the avenue of many promises.

In its original conception, public key cryptography, was concerned only with the difficulty of recovering the secret messages as a whole. That is, given the encryption of a message, it must be difficult to recover the complete message before encryption. The implementations originally proposed did not necessarily protect parts of the message from being inferred easily from its encrypted version. The first secure public-key crypto-system that dealt with this deficiency is due to Goldwasser and Micali [GM84].

1.3 Cryptographic Protocols

The mathematical properties of the public key crypto-systems have lead the researchers to investigate a broader field of interest and discover new applications of cryptography. A variety of new problems have come to life: problems where the concern is not the privacy of the information with respect to a third party, but the mutual privacy of the data used by the participants. A key concept in this branch of cryptography is the notion of a *protocol*. Intuitively, a protocol is a way for a number of parties to blend some secret information revealing part of it to the participants, while keeping part of it secret. A formal definition of this notion is provided later on in the thesis.

Influential examples of cryptographic protocols were the ones for secret-key exchange [DH76], oblivious transfer [Rab81] and its applications [Blu81,EGL83], coin flipping by telephone [Blu81], mental poker [SRA81,GM82], voting [CF85], and untraceable communications [Cha88]. Recently, results of a more general nature were obtained. Yao [Yao87] presented a protocol for the general problem of two-party private function evaluation, and Goldreich, Micali and Wigderson in [GMW87], presented a way to obtain a secure multi-party protocol from its abstract specifications.

It should be noticed that for building a protocol one needs more than mere encryption. Encryption, in fact, only provides a way to hide data, but not to "exchange" them. The latter is a much more challenging problem. Indeed, it is interesting to see that some of the protocols mentioned above, though presenting very interesting ideas, could be defeated, while some others are secure only in a restricted sense. For a long time the approach to security of protocols was rather ad hoc. People would write papers about secure protocols, but would merely argue that their protocols were secure. This state of affairs changed dramatically with the introduction of *zero-knowledge* [GMR89]. As will be seen in Chapter 2 this notion is crucial to an adequate treatment of protocol security.

1.4 Reductions

Often protocols are designed using simpler protocols as subroutines. In order to prove security of the global protocol, one must prove that the subroutine remains secure when composed in the specific way it is used. For this reason, we describe a notion of secure *reduction* that enables us to prove general results on composition of protocols. The idea of a reduction has appeared before in the literature [BVV84,BCR86,Cré88], but the notion of security was missing. Unlike very general results that give very general ways of building cryptographic protocols but lead to inefficient solutions, reductions give a number of tools to choose from to built protocols. Using the most appropriate building block one can design rather efficient protocols.

An example of a secure reduction is the construction of pseudo-random functions based on the existence of pseudo-random number generator [GGM86]. However, while they argue that their reduction is secure in a formal way, they do not present a general definition for secure reductions.

A formal definition of reduction among protocols is presented in this thesis. Using such a notion of reduction it becomes possible to describe a set of basic building blocks from which a large set of problems can be solved. Reductions us a trade off between generality and specificity. This thesis covers some of the main aspects of this new direction of research, as well as some practical implementations of these building blocks.

1.5 The Oblivious Transfer

The notion of an oblivious transfer was introduced by Rabin [Rab81]. In essence (a more precise definition will be given later), the notion can be described as follows. There are two parties a sender $\mathcal{S}$, and a receiver $\mathcal{R}$. Assume that party $\mathcal{S}$ owns a message m (which consists in Rabin's example of the factorization of a publicly known number) that $\mathcal{R}$ does not know. In an oblivious transfer, $\mathcal{R}$ will learn m with probability $\frac{1}{2}$. While $\mathcal{S}$ will not know whether m has been learned. Other than contributing this notion, Rabin also suggested a protocol for implementing it that is correct provided that $\mathcal{R}$ follows her [1] prescribed steps.

Fischer, Micali and Rackoff [FMR84] were the first to provide a protocol for the oblivious transfer that is secure no matter how the participants cheat.

Oblivious transfer is a very useful notion, as exemplified by Blum [Blu81] in the context of coin flipping, secret exchange and certified electronic mail. It is also a key ingredient of the mental poker protocol of Goldwasser and Micali [GM82].

An even more useful notion of oblivious transfer was later introduced by Even, Goldreich and Lempel in [EGL83]. In this notion the sender $\mathcal{S}$ has two messages m_0, m_1 and the receiver $\mathcal{R}$ has a selection bit c. In a 1-out-of-2 oblivious transfer $\mathcal{R}$ will earn message m_c while $\mathcal{S}$ will not find out which message $\mathcal{R}$ has learned.

Among the most notable applications of this later notion are the two party oblivious

[1] throughout this thesis we use $\mathcal{S}$ and $\mathcal{R}$ to identify the two parties involved in our protocols. $\mathcal{S}$ usually corresponds to the sender of information and $\mathcal{R}$ to the receiver of the information. We assume $\mathcal{S}$ to be masculine and $\mathcal{R}$ feminine.

circuit evaluation of Yao [Yao87] and the more sophisticated one of Kilian [Kil88]. In fact, while Yao makes use of cryptography on top of oblivious transfer, Kilian's solution solely depends one 1-out-of-2 oblivious transfer. An alternative simpler approach to the result of Kilian can be found in [Cré89].

Oblivious transfer is impossible from a purely information theoretic stand point. In fact, the security of the protocols of [Rab81,FMR84,EGL83] is based on the assumed intractability of computational problems such as factoring. In Chapter 4 we investigate the possibility of basing security on physical laws, such as those governing quantum mechanics.

Other notions of oblivious transfer have been studied and their description can be found in Chapter 5.

Chapter 2

Security, Correctness and Privacy

In this chapter we formalize the notions of a *protocol* and protocol *security.* Protocol security is built around two fundamental notions: correctness and privacy. We would like to thank Silvio Micali for providing the definition of protocol used in this thesis, and Joe Kilian and Silvio for the definition of privacy. The definition of correctness in this thesis, adequate for the protocols we discuss, is not general enough for handling more elaborate protocols. A stronger notion of correctness together with a more general definition of protocols and their security will be the subject of a forthcoming paper by Kilian, Micali and Rogaway [KMR]. The material of the current Chapter together with Chapter 4 will appear as s joint paper of Crépeau and Micali. Those familiar with the notion of zero-knowledge introduced by Goldwasser, Micali and Rackoff [GMR89] will recognize it as the key ingredient of protocol security.

While we need formal definitions to prove our protocols correct and private, we stress that the results we present in this thesis are *quite robust* against changes to the model. Therefore the reader should keep in mind that the results we present can not only be proven correct and private in our model but also in any similar “reasonable” model.

2.1 Other Concerns

We would like also to point out that security is not the only concern in a protocol. Another issue is *fairness* – which is particularly acute for 2-party protocols. Roughly said, a protocol is fair if one party does not get much information *before* the other party also does. (even a single bit may be too much!) The treatment of this subject started with the single-bit secret exchange protocol of Luby, Micali and Rackoff [LMR83]. Other results on fairness can be found in [BOGMR85,Blu84,Ted85,GHY85,Yao87,Cle89]. Finally Beaver and Goldwasser provided a general treatment of the subject in [BG89]. This crucial property will be totally ignored in this thesis as our protocols are not designed to *exchange* information, but only to *transfer* information from one party to the other.

2.2 Notation and Terminology

Let X be a probability space. We use $\mathrm{Prob}_X(u)$ to represent the probability that X associates with element u. Let s be a positive integer. Unless indicated otherwise, we represent s in binary representation. We use 1^s; to represent the unary representation of s. In general putting symbols side by side means concatenation. For instance ab mean a concatenated with b.

An Oracle Turing Machine is a machine with an extra oracle tape, on which it can write an input to be evaluated by an *oracle*. One or more of the states of the machine initiate the evaluation of the content of the oracle tape by the oracle. In one step the oracle will return on that tape the result of an arbitrary computation the oracle is designed for. For any function f, $M^{f(*)}$ is a Turing Machine capable of making **a single** call to an oracle for f. That is, M is able to get in one step the value $f(x)$, for a single x the machine may write on its oracle tape. Notice that the call to the oracle can occur at any point in the machine's computation. Unless specified otherwise, the Turing Machines we are concerned with are always probabilistic machines. It means for instance that for a machine M, $M(x)$ (the output of a machine M on input string x) is not a value in Σ^* but a probability distribution of the string in Σ^*. Notice that this is also true of machines with oracles.

Throughout this thesis, we assume boolean relations to have binary values, i.e. $true = 1$ and $false = 0$. For instance $(x = x)$ is 1, while $(x = 1 - x)$ is 0. The advantage is that expressions like $x \leftarrow (a = b)$ can be written to mean **if** $(a = b)$ **then** $x \leftarrow 1$ **else** $x \leftarrow 0$.

Let U be a set, D be a probability distribution over U, and u be a variable name. The expression $u \in_{\mathrm{R}} U$ means that a random value from U is assigned to u with uniform distribution, i.e. each element of U has equal probability of being assigned to u. The similar expression $u \in_{\mathrm{R}} D$ means that a random value from U is assigned to u, according to distribution U.

2.3 Interacting Turing Machines & Protocols

In describing what a protocol is, we distinguish its *hardware* component from its *software* one. The hardware in our model are Interacting Turing Machines (ITM), a notion explained below. The software to be run on these machines are protocols[1].

Definition 2.1 (Interacting Turing Machine) An *Interacting Turing Machine* (ITM) is a Turing Machine equipped with a finite state control, a read-only private input tape, a read-only common (shared with other such machines) input tape, a write-only private output tape, a read/write private work tape, a private random tape containing an infinite string of uniformly distributed bits and two additional communication tapes (one read-only, one write-only) used for communication with other identical machines.

For variety of discourse we sometimes call an ITM, a *party*. Let $\mathcal{S}$ and $\mathcal{R}$ be two parties. We call a pair $(\mathcal{R}, \mathcal{S})$ a *Connecting Pair of Parties* (CPP) if $\mathcal{S}$'s write-only

[1] some of the earlier mentioned references consider protocols as an amalgam of hardware and software. We view protocols as purely software objects.

Figure 2-1: a typical CPP

communication tape is $\mathcal{R}$'s read-only communication tape and vice versa, and the common input tape is readable by both of them (see figure 2-1).

Let σ and ρ be $\mathcal{S}$'s and $\mathcal{R}$'s private inputs. Let κ be their common input. (We use the greek letters σ, ρ, κ so that the reader can easily associate these inputs with the party they correspond to: σ to $\mathcal{S}$, ρ to $\mathcal{R}$, and κ is common to both of them) A CPP executes its software in rounds. In each round exactly one party is active. $\mathcal{S}$ is active first. When a party is active, it may do one or many of the following. It may read any of its (readable) tapes and perform local computations. It may send a message to the other party by writing a string on the communication tape it shares with it. It may write to some (writable) tapes. It may *switch* or *halt* the computation. Switching will result in stopping the party executing the instruction and activating the other party.

When one party issues the "halt" instruction, his execution is now terminated. If the other party has not yet issued that command, he gets activated so he can perform the end of his program. From that point on any "switch' or "halt" instruction will end the computation of the CPP.

Definition 2.2 (Protocol) Let $(\mathcal{S}, \mathcal{R})$ be a CPP. A *protocol* is a pair of programs $\wp = (\wp_{\mathcal{S}}, \wp_{\mathcal{R}})$ (Where $\wp_{\mathcal{S}}$ is $\mathcal{S}$'s finite state control and $\wp_{\mathcal{R}}$ is $\mathcal{R}$'s).

The output of the execution of a protocol is the pair of strings (ς, ϱ), where ς is the final content of the output tape of $\mathcal{S}$ and ϱ is the content of the output tape of $\mathcal{R}$. We can view a protocol as the description of a process that, for any fixed contents of the random tape, transforms a triplet of inputs σ, ρ, κ into a pair of outputs ς, ϱ. As the contents of the random tapes of $\mathcal{S}$ and $\mathcal{R}$ is randomly selected, for all $\sigma, \rho, \kappa \in \Sigma^*$ a protocol $\wp$ specifies a probability distribution over $\Sigma^* \times \Sigma^*$ (the output domain), namely $\wp(\sigma, \rho, \kappa)$ (the probabilities are taken over all possible contents of the random tapes).

The expression "$\mathcal{S}$ sends x to $\mathcal{R}$" means that $\mathcal{S}$ writes the string x on the communication tape he shares with $\mathcal{R}$. In general, the material of the present thesis will avoid using too much of the Turing Machine terminology. Nevertheless, precise definitions will be given along the way in order to explain thoroughly the formal meaning of words that might sound intuitively simple.

2.4 Protocol Specifications

Let $\mathcal{D} = \{D(\sigma, \rho, \kappa)\}_{\sigma,\rho,\kappa\in\Sigma^*}$ be a family of probability distributions over $\Sigma^* \times \Sigma^*$. We call $\mathcal{D}$ a *protocol specification* (or simply *specification*). Denote $D_{\mathcal{S}}(\sigma, \rho, \kappa)$ and $D_{\mathcal{R}}(\sigma, \rho, \kappa)$ the distributions induced by projecting $D(\sigma, \rho, \kappa)$ on its first or second component. These projections correspond to the specification of what $\mathcal{S}$ and $\mathcal{R}$ get respectively from a protocol. We say that a protocol $\wp$ *implements* specification $\mathcal{D}$ if $\forall \sigma, \rho, \kappa \in \Sigma^*$ $[\wp(\sigma, \rho, \kappa) = D(\sigma, \rho, \kappa)]$. Some specifications may be implemented by protocols, some may not. In general, given a protocol specification, our task is to find a protocol to implement it.

Whenever it is possible to express $D(\sigma, \rho, \kappa)$ in terms of $D_{\mathcal{S}}(\sigma, \rho, \kappa)$ and $D_{\mathcal{R}}(\sigma, \rho, \kappa)$ in a simple enough way, we write a protocol specification in the following table form

$\mathcal{S}$	σ	$\varsigma \in_{\mathrm{R}} D_{\mathcal{S}}(\sigma, \rho, \kappa)$
$\mathcal{R}$	ρ	$\varrho \in_{\mathrm{R}} D_{\mathcal{R}}(\sigma, \rho, \kappa)$
common	κ	

Where σ, ρ, κ are descriptions of the inputs to the protocol and ς, ϱ of the outputs. $D_{\mathcal{S}}(\sigma, \rho, \kappa)$ and $D_{\mathcal{R}}(\sigma, \rho, \kappa)$ are descriptions of the components of the output distribution $D(\sigma, \rho, \kappa)$ from which ς, ϱ are selected. The entries $\sigma, \rho, \kappa, \varsigma, \varrho$ can be composed objects. The reader should understand such object as the result of a binary coding. For instance if σ is a pair (u, v) it means that only inputs of the form $coding(u)00coding(v)$ (where $coding(n)$ is the string obtained from n by replacing each of its 0 by 10 and each of its 1 by 11) are considered.

Frequently, κ is empty; in these cases we just ignore the last line of the table:

$\mathcal{S}$	σ	$\varsigma \in_{\mathrm{R}} D_{\mathcal{S}}(\sigma, \rho)$
$\mathcal{R}$	ρ	$\varrho \in_{\mathrm{R}} D_{\mathcal{R}}(\sigma, \rho)$

As a first example consider a bit exchange protocol. This example does not involve probabilities at all:

$\mathcal{S}$	b_0	b_1
$\mathcal{R}$	b_1	b_0

In general the descriptions ς and ϱ involve exactly two types of variable names: the names of variables from σ, ρ, κ or the name of variables specifying the randomness involved in D. We usually name such variables © or $©_i$ (we use © for its similarity with a coin). We enumerate the probabilities with which the outcomes may occur. For example oblivious transfer would be

$\mathcal{S}$	b	
$\mathcal{R}$		$(\copyright, b') \in_{\mathrm{R}} \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b) & : \frac{1}{2} \end{cases}$

The outputs for both parties can be correlated, as for example

$\mathcal{S}$	b_0	$(\copyright, b_1') \in_{\mathrm{R}} \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b_1) & : \frac{1}{2} \end{cases}$
$\mathcal{R}$	b_1	$(\copyright, b_0') \in_{\mathrm{R}} \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b_0) & : \frac{1}{2} \end{cases}$

will output b_1 to $\mathcal{S}$ and b_0 to $\mathcal{R}$ with probability $\frac{1}{2}$ and $(0,0)$ to both otherwise, while it could also not be correlated as the following with four possible outcomes each occurring with probability $\frac{1}{4}$

$\mathcal{S}$	b_0	$(\copyright_1, b_1') \in_{\mathrm{R}} \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b_1) & : \frac{1}{2} \end{cases}$
$\mathcal{R}$	b_1	$(\copyright_0, b_0') \in_{\mathrm{R}} \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b_0) & : \frac{1}{2} \end{cases}$

In general, it is not possible to describe all protocol specifications in this manner, but fortunately, all the protocols we deal with in this thesis can be so represented.

2.5 Reductions Among Protocols

The main subject of investigation of this thesis is a set of special protocols known as *reductions*. Intuitively, a reduction is a protocol that relies on the existence of another protocol. Indeed, as we will see in Chapters 4 and 5, it seems that implementations of some protocol specifications are readily available in the physical world. In order to exploit this fact, we build a set of reductions that enable us to implement more interesting specifications based on the existence of the physical implementations. In

Figure 2-2: CPP with an abstract protocol

order to formalize how our parties can use existing protocols as "subroutines" we have to modify our notion of CPP to allow the execution of *abstract protocols*.

In addition to the hardware already described, each party of a CPP has an extra read/write protocol tape. These tapes are to be used in connection with a special purpose device known as an *abstract protocol* (represented by a black-box in Figure 2-2). The abstract protocol is a trusted device that implements a specific protocol specification. It may be viewed as a two-party oracle; abstract protocols are to protocol specifications, what oracles are to functions.

Let us now describe how parties can exchange data with an abstract protocol. At each step of its computation, a machine can read or write a single character on his protocol tape. The transformation of the inputs into outputs occurs after both parties issue a special *execute* command. Typically, the parties would first write a number of characters on their protocol tape, they would then request the execution of the abstract protocol and finally extract their outputs from the tape. After a party has issued the "execute" command he can no longer read from or write to his protocol tape until the other party has done the same. Essentially, his side is "frozen" until the other party decides to execute too. Executing the abstract protocol is an oblivious process: after executing, the abstract protocol looses track of its inputs: both protocol tapes contain only the results of the execution, prefixed by a special symbol "\$" to inform them that the computation has been performed.

A protocol $\wp$ involving the use of a set Π of abstract protocols is known as a *reduction* from $\wp$ to Π (or just $\wp'$ if $\Pi = \{\wp'\}$). In the remainder of this paper, we consider a number of reductions among different oblivious transfer protocol specifications.

2.6 Oblivious Transfer Specifications

An interesting subclass of protocol specifications is that of *Oblivious Transfers.* Here "Oblivious Transfer" is used in a very general way: an Oblivious Transfer Specification (OTS) is any protocol specification $\mathcal{D}$ over $\emptyset \times \Sigma^*$, i.e., a protocol that transfers information in only one direction.

All the protocol specifications described in this thesis are OTSs. Therefore, we restrict our focus to this class only. A major feature of this class is that we don't have to be concerned with "fairness" problems. In a general protocol, both parties get outputs. Unavoidably, one will get it before the other. This leads to extra complications due to the possibility of a party stopping after getting part of his output leaving the other party with no information about his. This issue is beyond the scope of the present work.

2.7 Correct and Private Protocols

The oblivious property of an abstract protocol inspires our notion of what an ideal protocol should be. Ideally, the receiving party ($\mathcal{R}$) should be able to compute his output value without learning "too much" of $\mathcal{S}$'s secret input. Similarly, $\mathcal{S}$ should not learn anything at all about $\mathcal{R}$'s input. We would like to come up with protocols that behaves just like abstract protocols. We formalize this intuition in the following subsections.

Here is a way of implementing an abstract protocol: Each party reveals his input to the other party, and then each of them computes his output. Of course this is fine as long as the parties do not care about keeping their input secret and wish to reveal it publicly. The situation we are concerned with is the setting where the parties wish to have the maximum *privacy* of their inputs.

Privacy is just one aspect of *security*. Another aspect is *correctness*. Correctness and privacy comes in two flavours: absolute and statistical. We will define these two variations, but we must go through a short digression on the indistinguishability of probability spaces.

Indistinguishability of Families of Probability Spaces

Let $\mathcal{X} = \{X_s\}_{s\in N}$ and $\mathcal{Y} = \{Y_s\}_{s\in N}$ be two families of probability spaces over Σ^*, parameterized by an integer s. The notion of indistinguishability expresses the fact that $\mathcal{X}$ and $\mathcal{Y}$ are replaceable by one another when s becomes large enough with respect to an observer of a given computing ability. This quite useful notion was introduced by Goldwasser and Micali [GM82], originally they used it in the context of probabilistic encryption schemes, Yao [Yao82] used it in the context of pseudo-random generation of bits and Goldreich, Micali and Wigderson [GMW87] used it in the context of cryptographic protocols. The game is to pick a random sample from either X_s or Y_s and hand it to a judge whose job is to determine where the sample comes from. $\mathcal{X}$

and $\mathcal{Y}$ are *indistinguishable* if for any judge, the judge's answers are independent from the origin of the sample. In other word, if $\text{Prob}_{X_s}(u) = \text{Prob}_{Y_s}(u)$ for all integer s and $u \in \Sigma^*$. We write this as $\mathcal{X} = \mathcal{Y}$.

Similarly, $\mathcal{X} = \{X_s\}_{s \in N}$ and $\mathcal{Y} = \{Y_s\}_{s \in N}$ are *statistically indistinguishable* (we write $\mathcal{X} \approx \mathcal{Y}$) if they are essentially replaceable by one another, for large s, with respect to a judge that may have unbounded computational power, but a limited number of sample points. In technical terms, we require that

$$\sum_{u \in \Sigma^*} |\text{Prob}_{X_s}(u) - \text{Prob}_{Y_s}(u)| < s^{-c},$$

for all $c > 0$ and for all sufficiently large integer s. The rational behind this definition is that no polynomial size sample (in s) will show any significant difference to any judge. Another important notion is computational indistinguishability (where the judge is bounded to polynomial time computations) which is not however used in this thesis.

2.8 Correctness

Let's just focus on correctness for now. A protocol is *absolutely correct* (we usually just say *correct*) if the probability distribution of the outputs of the protocol is the same as the distribution prescribed by the protocol specification.

Consider the probability space of the outputs of the protocol, indexed by a publicly shared parameter s known as the *security parameter* (this parameter is provided to the parties as a unary string on their common input tape).

Definition 2.3 (Correctness) Let $\mathcal{D} = \{D(\sigma, \rho, \kappa)\}_{\sigma,\rho,\kappa \in \Sigma^*}$ be a protocol specification. A protocol $\wp$ is *correct* for $\mathcal{D}$ if

$$\forall \sigma, \rho, \kappa \in \Sigma^* \ [\{\wp_s = \wp(\sigma, \rho, 1^s; \kappa)\}_{s \in N} = \{D_s = D(\sigma, \rho, \kappa)\}_{s \in N}].$$

Notice that D_s does not depend on s, which means that $\wp_s$ is also independent of s.

Definition 2.4 (Statistical Correctness) A protocol $\wp$ is *statistically correct* if

$$\forall \sigma, \rho, \kappa \in \Sigma^* \; [\{\wp_s = \wp(\sigma, \rho, 1^s; \kappa)\}_{s \in N} \approx \{D_s = D(\sigma, \rho, \kappa)\}_{s \in N}].$$

Notice that D_s does not depend on s, which means that $\wp_s$ gets closer and closer to the given D as s grows larger. We call s the security parameter of the protocol $\wp$ because this value determines with which precision the protocol we run will achieve the specification D. The fact that s is provided in unary will become relevant only when we consider the running time of the parties.

2.9 Privacy

Intuitively, we consider a cryptographic protocol to be private if the protocol behaves just like an abstract protocol: it does not give away any *information* about the inputs other than what each output inherently reveals.

During an execution of the protocol with party $\mathcal{S}$, $\mathcal{R}$ has access to some information. This information, which is all that he "sees" during the execution, is totally determined by $\mathcal{R}$'s *coin tosses* and the messages received from $\mathcal{S}$. This is a probability space (as it depends on $\mathcal{R}$'s and $\mathcal{S}$'s random tapes) denoted by $View_{\mathcal{R}}(\wp, (\sigma, \rho, \kappa))$ when $\mathcal{R}$ runs protocol $\wp_{\mathcal{R}}$ with ρ on her private input tape, and with κ on the common input tape, while $\mathcal{S}$ runs $\wp_{\mathcal{S}}$ on secret input σ and public input κ. Define $View_{\mathcal{S}}(\wp, (\sigma, \rho, \kappa))$ symmetrically.

The way to demonstrate that a protocol is private is by simulating the view of the participants. This approach follows the lines of Goldwasser, Micali and Rackoff's definition of *zero-knowledge* [GMR89]. Essentially, in this approach we define a protocol to be private if the behavior of a party can be simulated by the other one by himself as long as he is given the right help. We would like to mention that the formulation of privacy given next has been achieved with the help of Joe Kilian and Silvio Micali. We are infinitely thankful for their help.

Let $\mathcal{P}$ be one of $\mathcal{S}$, $\mathcal{R}$. The intuition behind the definition of $\mathcal{P}$-privacy is that running the actual protocol is computationally equivalent to a run of the corresponding abstract protocol. (If we were dealing with a general abstract protocol both definitions of $\mathcal{S}$-privacy and $\mathcal{R}$-privacy would be symmetrical.) But because of the asymmetrical nature of an oblivious transfer protocol, the notion of privacy is different for $\mathcal{S}$ and $\mathcal{R}$. We say that a protocol is private if it is $\mathcal{S}$-private and $\mathcal{R}$-private.

The definition of $\mathcal{S}$-privacy basically means that no program $\widetilde{\wp_{\mathcal{S}}}$ learns anything from his interaction with $\mathcal{R}$. We obviously need something different for $\mathcal{R}$ since, in general, she does get something from the interaction. (That is the whole point of the interaction!) Therefore this definition must be adapted to this fact.

Intuitively, a protocol is $\mathcal{R}$-private if all the information obtained by any program $\mathcal{R}$ might run is bounded by the information contained in the output of one honest run of the protocol.

When $\mathcal{R}$ is running a cheating program (i.e. a program different than the one she should be running) she can do a number of bad things. In order to remain undetected, $\mathcal{R}$ must however behave according to a legitimate choice of input. On private input ρ and common input κ she can actually behave as if her private input were a different value $\tilde{\rho}$ that she computes from ρ, κ, her coin tosses, and her knowledge of herself. This type of action would be totally unnoticed by $\mathcal{S}$. Actually there is nothing in the world one can do to stop $\mathcal{R}$ from doing such a thing. After all, it does not matter which input she uses since she chooses it anyway.

We say that a protocol $\wp = (\wp_{\mathcal{S}}, \wp_{\mathcal{R}})$ is $\mathcal{R}$-Private if every possible action taken by $\mathcal{R}$ can be interpreted as described above. That is, for every program $\widetilde{\wp_{\mathcal{R}}}$ interacting with the honest program $\wp_{\mathcal{S}}$ on inputs (σ, ρ, κ), an alternative program $M(\widetilde{\wp_{\mathcal{R}}})$ could generate exactly the same conversations with $\wp_{\mathcal{S}}$ while behaving as described in the above paragraph: $M(\widetilde{\wp_{\mathcal{R}}})$ first computes some $\tilde{\rho}$ based on ρ, κ and $\widetilde{\wp_{\mathcal{R}}}$, then given the outcome of running $\wp(\sigma, \tilde{\rho}, \kappa)$ can recreate an instance of the view $\widetilde{\wp_{\mathcal{R}}}$ would have gotten on input ρ.

We can formalize this statement using the notion of Turing Machine with single call to an oracle:

Definition 2.5 ($\mathcal{R}$-Privacy) Consider the function $\wp(\sigma, *, \kappa)$ induced from the specification for a protocol $\wp$. The protocol $\wp = (\wp_{\mathcal{S}}, \wp_{\mathcal{R}})$ is *$\mathcal{R}$-private* if:

$$\exists M \forall \widetilde{\wp_{\mathcal{R}}}, \sigma, \rho, \kappa \, [\{M_s = M^{\wp(\sigma,*,\kappa)}(\widetilde{\wp_{\mathcal{R}}}, \rho, 1^s; \kappa)\}_{s \in N} = \{V_s = View_{\mathcal{R}}((\wp_{\mathcal{S}}, \widetilde{\wp_{\mathcal{R}}}), (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

Definition 2.6 (Statistical $\mathcal{R}$-privacy) The protocol is *statistically $\mathcal{R}$-private* if the simulated view is only statistically indistinguishable from the actual view:

$$\exists M \forall \widetilde{\wp_{\mathcal{R}}}, \sigma, \rho, \kappa \, [\{M_s = M^{\wp(\sigma,*,\kappa)}(\widetilde{\wp_{\mathcal{R}}}, \rho, 1^s; \kappa)\}_{s \in N} \approx \{V_s = View_{\mathcal{R}}((\wp_{\mathcal{S}}, \widetilde{\wp_{\mathcal{R}}}), (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

Recall that, $M^{f(*)}$ is a Turing Machine that can make one oracle call to evaluate f on an input x. A few clarifying remarks are in order.

Remark 1

Since $M^{\wp(\sigma,*,\kappa)}$ is a probabilistic machine, $M^{\wp(\sigma,*,\kappa)}(\widetilde{\wp_{\mathcal{R}}}, \rho, 1^s; \kappa)$ is indeed a probability distribution as is $View_{\mathcal{R}}((\wp_{\mathcal{S}}, \widetilde{\wp_{\mathcal{R}}}), (\sigma, \rho, \kappa))$.

Remark 2

Note that if machine $M^{f(*)}$ is run twice on the same input, each time it will make a **single** oracle query. However, these two queries need not be the same.

Remark 3

The reason why we allow M an oracle call to $\wp(\sigma, *, \kappa)$ is to capture the fact that a bad player on input ρ can choose to change his input to $\tilde{\rho}$ (we can't prevent that) and obtain the result of the evaluation of $\wp(\sigma, \tilde{\rho}, \kappa)$. We should allow the simulator M the same ability to see an output of the protocol on input $\tilde{\rho}$.

For example, a possible behavior of machine M is to first compute a $\tilde{\rho}$, then call the oracle to evaluate $\wp(\sigma, \tilde{\rho}, \kappa)$, and finally compute a simulated view to output. However, there is no restriction on the order of the operations. (For instance in the simulators we describe later, the generation of the view is entangled with the computation of $\tilde{\rho}$)

This definition simplifies to the following for $\mathcal{S}$ since he does not get any output from an OTS.

Definition 2.7 ($\mathcal{S}$-Privacy) The protocol $\wp = (\wp_{\mathcal{S}}, \wp_{\mathcal{R}})$ is $\mathcal{S}$*-private* if $View_{\mathcal{S}}((\widetilde{\wp_{\mathcal{S}}}, \wp_{\mathcal{R}}), (\sigma, \rho, \kappa))$ is independent of ρ and

$$\exists M \forall \widetilde{\wp_{\mathcal{S}}}, \sigma, \rho, \kappa \; [\{M_s = M(\widetilde{\wp_{\mathcal{S}}}, \sigma, 1^s; \kappa)\}_{s \in N} = \{V_s = View_{\mathcal{S}}((\widetilde{\wp_{\mathcal{S}}}, \wp_{\mathcal{R}}), (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

Definition 2.8 (Statistical $\mathcal{S}$-privacy) The protocol is *statistically* $\mathcal{S}$*-private* if the simulated views are only statistically indistinguishable from the actual view:

$$\exists M \forall \widetilde{\wp_{\mathcal{S}}}, \sigma, \rho, \kappa \; [\{M_s = M(\widetilde{\wp_{\mathcal{S}}}, \sigma, 1^s; \kappa)\}_{s \in N} \approx \{V_s = View_{\mathcal{S}}((\widetilde{\wp_{\mathcal{S}}}, \wp_{\mathcal{R}}), (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

We say that a protocol privately implements a protocol specification if it achieves correctness and privacy. Correct protocols may be easy to design. Private protocols may be easy to design, too. Unfortunately, correct **and** private protocols are hard to design. The whole point of this thesis is to present a set of protocols achieving both properties. It would be extremely easy to design correct or private protocols for the problems suggested later.

2.10 Honest Protocols

Intuitively, a protocol is *honest* if it is private for the prescribed programs. Honesty is a weak form of privacy. Instead of being able to simulate every possible programs, the simulator for an honest protocol, need only to simulate the "honest" programs.

Definition 2.9 ($\mathcal{R}$-Honesty) A protocol $\wp$ is $\mathcal{R}$*-honest* if

$$\exists M \forall \sigma, \rho, \kappa \; [\{M_s = M(\wp(\sigma, \rho, \kappa), \rho, 1^s; \kappa)\}_{s \in N} = \{V_s = View_{\mathcal{R}}(\wp, (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

and is *statistically $\mathcal{R}$-honest* if

$$\exists M \forall \sigma, \rho, \kappa \; [\{M_s = M(\wp(\sigma, \rho, \kappa), \rho, 1^s; \kappa)\}_{s \in N} \approx \{V_s = View_{\mathcal{R}}(\wp, (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

Definition 2.10 ($\mathcal{S}$-Honesty) The protocol $\wp$ is *$\mathcal{S}$-honest* if

$$\exists M \forall \sigma, \rho, \kappa \; [\{M_s = M(\sigma, 1^s; \kappa)\}_{s \in N} = \{V_s = View_{\mathcal{S}}(\wp, (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

and is *statistically $\mathcal{S}$-honest* if

$$\exists M \forall \sigma, \rho, \kappa \; [\{M_s = M(\sigma, 1^s; \kappa)\}_{s \in N} \approx \{V_s = View_{\mathcal{S}}(\wp, (\sigma, \rho, 1^s; \kappa))\}_{s \in N}].$$

We will see in Chapter 4 how an honest oblivious bit transfer protocol can be used to build a correct and private oblivious bit transfer protocol.

2.11 Complexity of Protocols

Note that the approach to privacy and honesty described above is oriented toward protocols where the parties have unlimited computing power. Protocols that are private in the previous way are effective against adversaries with unbounded computational resources. The drawback of this approach is that it neglects completely the amount of "knowledge" [GMR89] given away by the protocol. If we wish to consider the meaning of privacy in that context, we must express our conditions in term of **polynomial time** constructibility of the views of the parties.

For matter of efficiency, we focus only on protocols executable in polynomial time (in the total length of the inputs). Therefore, from now on it should be implicit that a correct protocol must run in expected polynomial time. Similarly, in order to get the strongest privacy constraint possible, we request that the simulators run in expected polynomial time too (counting subroutine calls or call to an abstract protocol as 1 step). In that case the level of privacy achieved is known as *zero-knowledge* [GMR89].

Indeed, what we want is an extended version of the standard definition of zero-knowledge that applies to parties of every possible computing power. We use extended notions of zero-knowledge, as suggested in [BC89]. The only key fact we care about is that our simulators should run essentially in an amount of time comparable to the actual party it is simulating. This is what the condition above guarantees.

2.12 Correct and Private Reductions

We treat the correctness and privacy of a reduction almost exactly the same way that we treat them for protocols. The only difference is that the simulator is allowed use of the set of abstract protocols. It is given total control of the inputs and outputs to the "device" and may use it at the cost of a single Turing Machine step per function computation. We assume that the abstract protocol is perfectly secure.

Chapter 3

Reductions Among Oblivious Transfers

As a first example of a reduction among protocols, we reduce the following two well known transfer: 1-out-of-2 Oblivious Bit Transfer ($\binom{2}{1}$–OBT) and Oblivious Bit Transfer (OBT). There is a trivial way to use $\binom{2}{1}$–OBT in order to achieve OBT. Therefore we concentrate only on the reduction of $\binom{2}{1}$–OBT to OBT.

3.1 Two Notions of Oblivious Transfers

First we present the two problems:

Definition 3.1 (OBT)

protocol specification:

$\mathcal{S}$	b	
$\mathcal{R}$		$(©, b') \in_R \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b) & : \frac{1}{2} \end{cases}$

In other words, in this OTS $\mathcal{S}$ has a secret bit b. At the end of the transfer, one of the following two events occurs, each with probability $\frac{1}{2}$.

1. $\mathcal{R}$ learns the value of b.

2. $\mathcal{R}$ gains no further information about the value of b (other than what she knew before the transfer).

At the end of the transfer, $\mathcal{R}$ knows which of these two events actually occurred, and $\mathcal{S}$ learns nothing.

Definition 3.2 ($\binom{2}{1}$–OBT)
protocol specification:

$\mathcal{S}$	(b_0, b_1)	
$\mathcal{R}$	c	b_c

In this OTS, $\mathcal{S}$ has two secret bits, b_0 and b_1. $\mathcal{R}$ has a selection bit c. At the end of the transfer, the following three conditions hold.

1. $\mathcal{R}$ learns the value of b_c.

2. $\mathcal{R}$ gains no information about the value of b_{1-c}.

3. $\mathcal{S}$ learns nothing about the value of c.

3.2 The Basic Idea of the Reduction

The general idea of the reduction is to use the OBT s times, with random inputs $r_i \in \{0, 1\}$. On average, $\frac{s}{2}$ bits are expected to be transferred. The parameter s is chosen large enough so that with high probability $\frac{s}{3}$ bits are transferred, but not $\frac{2s}{3}$. We ask the receiver to build two disjoint sets of r_i's of size $\frac{n}{3}$ and associate each of b_0 with one set, and b_1 with the other. Bit b_c will be transferred only if the corresponding set contains only r_i's that were transferred by the protocol.

The number of successful transfers approximates a binomial distribution, as the number of trials grows. A very useful theorem to describe the asymptotic behavior of binomial distributions is the "Bernsthein's law of large numbers". A complete proof of the two following equivalent propositions may be found in [Kra86].

Theorem 3.3 (Bernsthein) *if* $\mathrm{Prob}(b_i = 1) = p$ *for* $1 \leq i \leq s$ *then for all* $0 < \delta \leq p(p-1)$ *we have*

$$\mathrm{Prob}(|\sum_{i=1}^{s} \frac{b_i}{s} - p| \geq \delta) \leq 2e^{-s\delta^2}.$$

or in an equivalent form:

Theorem 3.4 (Bernsthein) *if* $\mathrm{Prob}(b_i = 1) = p$ *for* $1 \leq i \leq s$ *then for all* $0 < k \leq p(p-1)\sqrt{s}$ *we have*

$$\mathrm{Prob}(|\sum_{i=1}^{s} b_i - sp| \geq k\sqrt{s}) \leq 2e^{-k^2}.$$

We will use this fact a number of times through out this thesis. In some sense it is the main element from the theory of statistics that is relevant to the current work.

3.3 The Reduction

(for simplicity, assume that s is a multiple of 3)

Protocol 3.1 ($\binom{2}{1}$**–OBT**$((b_0, b_1), c, 1^s;)$ **)**

1: $\mathcal{S}$ *chooses at random* s *bits* $r_1, r_2, ..., r_s$

2: $\underset{i=1}{\overset{s}{\mathbf{DO}}}$ *(*$\mathcal{S}$*,*$\mathcal{R}$*) run* $OBT(r_i,)$*,* $\mathcal{R}$ *gets* $(©_i, r'_i)$

(continued on next page)

Protocol 3.1

3: $\mathcal{R}$ *selects at random* $I_0 = \{i_1, i_2, ..., i_{\frac{s}{3}}\}$, *if possible, such that* $©_i = 1$ *for each* $i \in I_0$, *and* $I_1 = \{i_{\frac{s}{3}+1}, i_{\frac{s}{3}+2}, ..., i_{\frac{2s}{3}}\}$, *with* $I_0 \bigcap I_1 = \emptyset$

4: $\mathcal{R}$ *sends* $N_0 = I_c$ *and* $N_1 = I_{1-c}$ *to* $\mathcal{S}$

5: **if** N_0 *and* N_1 *are not disjoint or of the right size* **then** $\mathcal{S}$ *stops*

6: $\mathcal{S}$ *computes* $\hat{b}_x = b_x \oplus \bigoplus_{n \in N_x} r_n$ *for* $x = 0, 1$ *and sends* $\hat{b}_0, \hat{b}_1$ *to* $\mathcal{R}$

7: $\mathcal{R}$ *outputs* $\hat{b}_c \oplus \bigoplus_{i \in I_0} r'_i$

Notice that the "run" operation of Step 2 is an extreme simplification of a more complicated process: first $\mathcal{S}$ would write his input on his protocol tape and issue the "execute" operation. Then $\mathcal{S}$ would switch and give control to $\mathcal{R}$ who would issue the "execute" operation too. Finally, $\mathcal{R}$ gets her output on her protocol tape.

3.4 Properties of the Reduction

Theorem 3.5 *Protocol 3.1 constitutes a statistically correct and private reduction of* $\binom{2}{1}$*–OBT to OBT.*

Proof. follows from Lemma 3.6 and 3.8. □

3.4.1 Correctness

First we show that the above protocol is correct.

Lemma 3.6 *Protocol 3.1 is statistically correct*

Proof.

First notice that the only condition needed for it to be correct is that in an honest run between $(\mathcal{S}, \mathcal{R})$, $\mathcal{R}$ can complete Step 3. Now consider the probability that this step can be completed:

Proposition 3.7 *There exists a positive constant $\alpha < 1$ such that*

$$\text{Prob}(\mathcal{R} \textit{ can complete Step 3 }) \geq 1 - \alpha^s$$

when the protocol 3.1 is run as described.

Proof.

$$\begin{aligned}
\text{Prob}(\mathcal{R} \text{ can complete step 3 }) &= \text{Prob}(\sum_{i=1}^{s} ©_i \geq \frac{s}{3}) \\
&= 1 - \text{Prob}(\sum_{i=1}^{s} ©_i < \frac{s}{3}) \\
&= 1 - \text{Prob}(\frac{s}{6} < \frac{s}{2} - \sum_{i=1}^{s} ©_i) \\
&\geq 1 - \text{Prob}(\frac{s}{6} \leq \frac{s}{2} - \sum_{i=1}^{s} ©_i) \\
&\geq 1 - \text{Prob}(\frac{s}{6} \leq |\frac{s}{2} - \sum_{i=1}^{s} ©_i|) \\
&= 1 - \text{Prob}(|\sum_{i=1}^{s} ©_i - \frac{s}{2}| \geq \frac{s}{6}) \\
&\geq 1 - 2e^{-\frac{s}{36}} \quad \text{by Theorem 3.3} \\
&\geq 1 - \left(\frac{e}{2}\right)^{-\frac{s}{36}} \quad \text{for large enough } s
\end{aligned}$$

the proposition follows by taking $\alpha = \sqrt[36]{\frac{2}{e}}$.

We find that this protocol is statistically correct since $\mathcal{R}$ fails to output b_c only with probability α^s. □

3.4.2 Privacy

Next, we show that this protocol is private. The fact that it is $\mathcal{S}$-private is rather trivial, since the only data $\mathcal{S}$ ever receives through this protocol is the two sets N_0, N_1, which contain random indices. Therefore a simulator would basically run $\widetilde{\wp_{\mathcal{S}}}$ and provide him with two sets of indices of the right size at Step 3. On the contrary proving the $\mathcal{R}$ privacy is not so easy. Consider the following simulator for any program $\widetilde{\wp_{\mathcal{R}}}$, $\mathcal{R}$ might run. Notice that, although it is not explicit in the simulation, whenever M runs $\widetilde{\wp_{\mathcal{R}}}$ it is possible the it will never complete the computation demanded. In that case M just run forever if we consider a definition of privacy where time bounds do not matter, and M stop after a given time otherwise if we consider time bounds.

Protocol 3.2 (simulation$(\widetilde{\wp_{\mathcal{R}}}, c, 1^s;)$ **)**

1: *M chooses at random s bits $r_1, r_2, ..., r_s$*

2: *M executes $\widetilde{\wp_{\mathcal{R}}}$ on inputs $c, 1^s$; until it is ready to perform step 2 of protocol 3.1*

3: $\underset{i=1}{\overset{s}{\textbf{DO}}}$ *M runs $OBT(r_i,)$, M gets $(©_i, r'_i)$ and uses then with $\widetilde{\wp_{\mathcal{R}}}$*

4: *M runs $\widetilde{\wp_{\mathcal{R}}}$ until she sends N_0 and N_1*

5: **if** *N_0 and N_1 are not disjoint or of the right size* **then** *M stops*

6: *M computes c' such that $\forall i \in N_{c'}, [©_i = 1]$*
(**if** *possible for both or none* **then** *picks c' at random)*

7: *M calls the oracle to compute $\binom{2}{1}$–$OBT((b_0, b_1), c')$ and get $b_{c'}$*

(continued on next page)

Protocol 3.2

8: *M computes* $\hat{b}_x = b_{c'} \oplus \bigoplus_{n \in N_x} r_n$ *for* $x = 0, 1$ *and uses* $\hat{b_0}, \hat{b_1}$ *with* $\widetilde{\wp_{\mathcal{R}}}$

9: *M runs* $\widetilde{\wp_{\mathcal{R}}}$ *until she completes her computation*

Remarks

Notice that the use of OBT in step 3 is actually unnecessary. M could have generated the ©$_i$s by flipping coins. We used the call to OBT just to show that such a thing is possible. Indeed, in chapter 4 a similar simulation will need to call its subroutine in order to simulate accurately. The parallel between the two simulations is therefore simplified by our use of OBT.

The Result

Lemma 3.8 *Protocol 3.1 is statistically* $\mathcal{R}$*-private*

Proof.

First notice, by inspection of the protocol and the simulator, that the only condition needed for the simulation to be exact is that $\exists n \in N_0 \bigcup N_1$ such that ©$_n = 0$. This is because this ©$_n$ guarantees that $\mathcal{R}$ cannot compute $\bigoplus_{i \in I_0} r'_i$ and $\bigoplus_{i \in I_1} r'_i$. Now consider the probability that this event does not occur:

Proposition 3.9 *There exists a positive constant* $\alpha < 1$ *such that*

$$\text{Prob}(\forall n \in N_0 \bigcup N_1, \text{©}_n = 1) \leq \alpha^s$$

Proof.

$$\begin{aligned}
\text{Prob}(\forall n \in N_0 \bigcup N_1, ©_n = 1) &= \text{Prob}(\sum_{i=1}^{s} ©_i \geq \frac{2s}{3}) \\
&= \text{Prob}(\sum_{i=1}^{s} ©_i - \frac{s}{2} \geq \frac{s}{6}) \\
&\leq \text{Prob}(|\sum_{i=1}^{s} ©_i - \frac{s}{2}| \geq \frac{s}{6}) \\
&\leq 2e^{-\frac{s}{36}} \quad \text{by theorem 3.3} \\
&\leq \left(\frac{e}{2}\right)^{-\frac{s}{36}}
\end{aligned}$$

the proposition follows by taking $\alpha = \sqrt[36]{\frac{2}{e}}$. □

For all the cases where $\sum_{i=1}^{s} ©_i \leq \frac{2s}{3}$ the simulator will recreate exactly the probability distribution of the actual conversations. Only in the cases where $\sum_{i=1}^{s} ©_i > \frac{2s}{3}$ will the distribution be different since the real $\mathcal{S}$ will provide $b_{1-c'}$ as well as $b_{c'}$ while M will provide $b_{c'}$ twice. Notice that in the case where $b_0 = b_1$ the simulator will always provide the right value since $b_{c'} = b_{1-c'}$.

Define for simplicity $\wp = \binom{2}{1}$–OBT. Let $\mathcal{V}_s = View_{\mathcal{R}}((\wp_{\mathcal{S}}, \widetilde{\wp_{\mathcal{R}}}), ((b_0, b_1), c, 1^s;))$ and $\mathcal{M}_s = M^{\wp((b_0,b_1),*)}(\widetilde{\wp_{\mathcal{R}}}, c, 1^s;)$. We have that $\mathcal{V} = \{\mathcal{V}_s\}_{s\in N}$ and $\mathcal{M} = \{\mathcal{M}_s\}_{s\in N}$ are statistically indistinguishable since

Proposition 3.10

$$\sum_{u\in\Sigma^*} |\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{M}_s}(u)| < \alpha^s$$

for the α given by lemma 3.9 and all sufficiently large s.

Proof. Let u be a possible view (a string such that $\text{Prob}_{\mathcal{V}_s}(u) > 0$). When $b_0 = b_1$ we have exactly that $\text{Prob}_{\mathcal{V}_s}(u) = \text{Prob}_{\mathcal{M}_s}(u)$. Let $\Sigma^*_\alpha = \{u \in \Sigma^* | \sum_{i=1}^{s} ©_i^u \geq \frac{2s}{3}\}$ (the value of the $©_i^u$s are implicitly defined by u: In any view u we can extract a number of bits that correspond to the variables $©_i$ from the protocol 3.1, we call those $©_i^u$.). If $b_0 \neq b_1$ we get that

$$\sum_{u\in\Sigma^*} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| = \sum_{u\in\Sigma^*_\alpha} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| + \sum_{u\in\Sigma^*-\Sigma^*_\alpha} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)|$$

but notice that $u \in \Sigma^* - \Sigma^*_\alpha$ implies that $\mathrm{Prob}_{\mathcal{V}_s}(u) = \mathrm{Prob}_{\mathcal{M}_s}(u)$ therefore

$$\sum_{u\in\Sigma^*} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| = \sum_{u\in\Sigma^*_\alpha} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)|$$

and since

$$\sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u),\ \sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{M}_s}(u) \leq \mathrm{Prob}(u \in \Sigma^*_\alpha) \leq \alpha^s$$

we have that

$$\sum_{u\in\Sigma^*_\alpha} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| \leq \sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u) + \sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{M}_s}(u)$$

which leads to

$$\sum_{u\in\Sigma^*} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| \leq 2\alpha^s$$

□

The proof of the lemma 3.8 and therefore of the theorem 3.5 follows.

3.5 Generalization

Notice that the reduction 3.1 does not rely very much on the fact that OBT transfers bits with probability $\frac{1}{2}$. Indeed the same reduction can be generalized to use an OBT protocol that transfer bits with an arbitrary probability p (see [Cré88]). The only difference necessary to adjust to p is to use sets of size $\frac{2ps}{3}$ in step 3 instead of $\frac{s}{3}$, when $p < \frac{3}{4}$ and of size $\frac{s}{2}$ when $p \geq \frac{3}{4}$.

3.6 Conclusion

The techniques developed in this chapter are very useful. In the next chapter we use similar techniques to reduce oblivious transfers to a natural source of randomness. Thus connecting theory to practice.

Chapter 4

Uncertainty as a Source of Privacy

In this section we describe how to implement the $\binom{2}{1}$–OBT protocol using an abstracted consequence of Heisenberg's uncertainty principle. This work stems from ideas described in [BBBW83], a paper about quantum cryptography.

4.1 History

In the late 1960's a physicist, Steven Wiesner, thought of using the uncertainty principle for cryptography (though he published his result much later [Wie83]). His idea was to transmit two pieces of information using streams of photons in a way that would make only one of piece readable, at the receiver's choice. This notion, which he called "multiplexing", is remarkably similar to the $\binom{2}{1}$–OBTreinvented many years later. A decade after his original invention, Charles Bennett and Gilles Brassard brought Wiesner's idea back to life and used it to implement some basic cryptographic protocols, such as secret key exchange and coin flipping by telephone [BB84]. We shall see, however, that all of these protocols present subtle difficulties.

Figure 4-1: four photons with different polarizations

4.2 Physics Background

Remember that the uncertainty principle states that some quantum systems have *conjugate* variables, meaning that if one of them is read, the other is necessarily lost. Heisenberg first used this principle to explain the impossibility of determining the vectors of both the position and the momentum of an electron. Today many such conjugate variables are known. The one Wiesner was concerned with was the linear polarization of photons.

A photon is a small entity of electromagnetic energy. It is know to us by the electric and magnetic forces it carries, although it is known that these two forces are different manifestations of a single thing. Physics tells us that a photon's magnetic and electric forces are orthogonal to each other and orthogonal to the direction of the photon movement (which forces our two-dimensional figures to be approximate). The angle by which the magnetic axis of a photon must be rotated clockwise in order to superimpose a fixed axis is called its *linear polarization*. (Thus polarization $-45°$ and $+135°$ are the same linear polarization.) In our figures, this fixed axis coincides with the bottom edge of the page, and a photon is represented by the axis of its magnetic force, drawn as a double arrow. For example, Figure 4-1 represents photons of polarizations $\phi = 0°, 45°, 90°, 135°$.

A photon's linear polarization is a quantum state: it cannot be determined accurately. Actually, although the value of the polarization can be anywhere in the (real) range $[0^\circ - 180^\circ)$, only a boolean predicate of this angle can be evaluated. For instance, let ϕ be the polarization of a photon. A question that one may answer by a measurement is: "Is $\phi = 0$?". The result of this measurement is probabilistically determined by the value of ϕ—the probability that it is "yes" equals $cos^2\phi$. This means that our experiment returns an answer *guaranteed* to be correct only if we have (somehow) an a priori *guarantee* that ϕ is either 0° or 90°,

It is not that no one has a good enough device for measuring ϕ; quantum theory states that it is *impossible* to determine this value. In some sense the value does not even really exist: it is just a probability distribution.

Another consequence of the uncertainty principle is that information has a cost. By this we mean that physical systems are disturbed by attempts to determine something about them. So the cost of information is the loss of other information. In our case, the cost of information is that merely evaluating a boolean predicate of a photon's polarization will change this variable's value and therefore all hope of determining any more about it is lost.

The polarization ϕ of a photon can be measured (in the above sense) by many devices. For instance, it can be measured by means of a Nichol's prism positioned at angle $\theta = 0$ and two photon detectors. In fact, a photon that enters a Nichol's prism with polarization ϕ exits the prism on the right side with probability $\cos^2(\phi)$ (and will then have polarization 0) and on the left side with the complement probability $\sin^2(\phi)$ (and will then have polarization 90°). Which of these events has occurred can be determined by a pair of photon detectors. According to quantum mechanics, no device can measure ϕ better.

Using a Nichol's prism, it is also possible to generate and transmit a photon with a given polarization.

Figure 4-2: Photon's behavior when passing through a Nichol's prism

4.3 The BBBW Protocol

Bennett, Breidbart, Brassard, and Wiesner [BBBW83], refining the original idea of [Wie83], devised the following protocol and presented various applications:

Protocol 4.1 (BBBW)

1: *$\mathcal{S}$ selects a polarization angle ϕ and transmits a photon with that polarization to $\mathcal{R}$*

2: *$\mathcal{R}$ chooses an angle θ, positions her Nichol's prism at angle θ, and measures the polarization of $\mathcal{S}$'s photon*

For simplicity we assume that both angles have a rational number of degrees, since we need our parties to be able to express their choice of angle in a finite way and because all of the angles we are concerned with are rational.

This protocol motivates the following protocol specification:

Definition 4.1 (Quantum Transfer)

protocol specification:

$\mathcal{S}$	ϕ	
$\mathcal{R}$	θ	$© \in_R \begin{cases} 0 & : \cos^2(\phi-\theta) \\ 1 & : \sin^2(\phi-\theta) \end{cases}$

In this transfer, $\mathcal{S}$ and $\mathcal{R}$ have rational numbers of degrees, ϕ and θ. At the end of the transfer, $\mathcal{R}$ receives a value which is 0 with probability $\cos^2(\phi-\theta)$, and 1 otherwise. $\mathcal{S}$ learns nothing.

[BBBW83] presented a reduction of $\binom{2}{1}$–OBT to the quantum transfer. This reduction, though, is secure only if $\binom{2}{1}$–OBT is defined in a weaker sense, or if both participants honestly follow their prescribed instructions. By contrast, in section 4.7 we present a reduction of $\binom{2}{1}$–OBT to the quantum transfer that is secure no matter what the participants do.

First, however, we should address a more fundamental question:

4.4 Is the BBBW Protocol a Secure Implementation of the Quantum Transfer?

The answer to this question is "no", because the real world fails to satisfy our model in several different ways.

First, our model is synchronous; inside our mechanism for using abstract protocols is a signal (the "$" on the protocol tape) meant to inform the parties that the protocol has been executed. This enables us to write instructions like "($\mathcal{S}$, $\mathcal{R}$) run ..." because the parties can check for the end of its execution before continuing on to other instructions.

In reality, if we transfer information using photons, there is no way for the sender to check whether the receiver has yet read the photon. We call this problem *delayed evaluation*.

A second problem is that, in our model, each run of an abstract protocol has an independent outcome. In reality, it is possible to create a pair of photons with random but correlated polarizations. This phenomenon is known as the Einstein-Podolsky-Rosen paradox and has been verified experimentally. The result of measurements on two such photons will be correlated no matter where the experiments take place (it is called a paradox because Einstein could not admit that this correlation could take place instantaneously at arbitrary distance). Our model definitely fails to incorporate this possibility.

A third problem is that in principle it is possible to perform computations on photons. The amount of information available about the polarization of a photon is limited to one bit. The trouble is that the two bits of information available from two photons are not necessarily those available from each photon separately. For instance, it might be possible to determine accurately the bit of information "is $\phi = \theta$?", which is impossible to determine from the separate information available about ϕ and θ. Physicists call this a *coherent measurement*. This provides the receiver with extra tools to determine the sender's secret information, which is not covered by our model.

There are a number of other ways in which our model can differ from reality. We do not claim to have an exhaustive list of these differences. We analyze in section 4.8 how the problems mentioned above can affect the practical version of the basic quantum protocol.

In no way do we imply that our protocol specifications can be securely implemented using the technology discussed above. As we will see in section 4.8, one can only approximate these specifications given the disparity between our model of computation and reality. We describe the real world primarily to motivate our choice of primitive.

4.5 An Honest OBT Based on the Quantum Transfer

Assume that the quantum transfer is given as an abstract protocol. In order to later understand more complex protocols, we first consider an honest protocol for the simpler problem of OBT that is not necessarily private. We will later use this protocol in our construction of a private protocol for $\binom{2}{1}$–OBT.

Assume $\mathcal{S}$ wants to obliviously transfer a bit b. A useful thing to notice about the quantum transfer is that $\sin^2(\phi + b \times 90° - \theta) = \sin^2(b \times 90°) = b$ when $\phi = \theta$, while $\sin^2(\phi + b \times 90° - \theta) = \sin^2(b \times 90° \pm 45°) = \frac{1}{2}$ when $|\phi - \theta| = 45°$. This establishes a correspondence between $\phi = \theta$ and $\mathcal{R}$ getting the bit b, and between $|\phi - \theta| = 45°$ and $\mathcal{R}$ getting nothing.

From these facts we can deduce the following very simple OBT protocol:

Protocol 4.2 (Honest-OBT$(b,)$ **)**

1: *$\mathcal{S}$ chooses a random ϕ in $\{0°, 45°\}$*

2: *$\mathcal{R}$ chooses a random θ in $\{0°, 45°\}$*

3: *($\mathcal{S}$,$\mathcal{R}$) run $QT(\phi + b \times 90°, \theta)$. $\mathcal{R}$ gets ©*

4: *$\mathcal{S}$ sends ϕ to $\mathcal{R}$*

5: *$\mathcal{R}$ outputs $((\phi = \theta), (\phi = \theta) \wedge$ ©$)$*

4.6 Honesty of the Reduction

Theorem 4.2 *Protocol 4.2 constitutes a statistically correct and honest reduction of*

OBT to QT.

4.6.1 Correctness

First we show that the protocol is correct.

Lemma 4.3 *Protocol 4.2 is statistically correct*

Proof.

The fundamental property exploited by this protocol is that for every angle ϕ and every bit b we have

$$\begin{aligned}\mathrm{Prob}(\copyright = b|b = 1) &= \mathrm{Prob}(\copyright = 1)\\ &= \sin^2(\phi + 90^\circ - \theta)\\ &= \cos^2(\phi - \theta)\end{aligned}$$

and

$$\begin{aligned}\mathrm{Prob}(\copyright = b|b = 0) &= \mathrm{Prob}(\copyright = 0)\\ &= \cos^2(\phi - \theta).\end{aligned}$$

Therefore

$$\mathrm{Prob}(\copyright = b) = \cos^2(\phi - \theta.)$$

Note that $\mathrm{Prob}(\phi = \theta) = \frac{1}{2}$, so with probability $\frac{1}{2}$ $\mathcal{R}$ will get $\copyright = b$ and know that she did because she learned ϕ in the last step. On the other hand, $\mathrm{Prob}(|\phi - \theta| = 45^\circ) = \frac{1}{2}$, and in this case, $\mathcal{R}$ learns nothing about b because $\cos^2 45^\circ = \frac{1}{2}$. This very simple protocol achieves exactly the OBT as long as the parties are honest and stick to their instructions. □

4.6.2 Honesty

This protocol is trivially $\mathcal{S}$-honest, since $\mathcal{S}$ never gets anything from $\mathcal{R}$. It is also easy to see that it is $\mathcal{R}$-honest, since for any run of OBT it is easy to choose a set of angles ϕ, θ consistent with that run.

4.6.3 Is this reduction Private?

Now the trouble is, we don't really know what $\mathcal{R}$ is doing. She can use any angle θ to get the bit ©, and can ensure that she gets something about b in each case. By using $\theta = 22\frac{1}{2}^\circ$, for example, she gets that Prob(© $= b) = \cos^2(22\frac{1}{2}^\circ) \approx 0.8536$, whether $\phi = 0^\circ$ or $\phi = 45^\circ$.

4.7 Reducing $\binom{2}{1}$–OBT to Quantum Transfer

The general idea of the reduction is to use the Quantum Transfer to implement the non-private OBT. Then, by strengthening our previous results, we show that reduction 3.1 is still private even when this non private OBT is used instead of a private one.

4.7.1 The Reduction

Replace Step 2 of protocol 3.1 by the following (for simplicity, assume that s is a multiple of 3):

Protocol 4.3 ($\binom{2}{1}$–**OBT**$((b_0, b_1), c, 1^s;)$ **)**

1: ...

2: $\underset{i=1}{\overset{s}{\mathbf{DO}}}$

2.1 *$\mathcal{S}$ chooses a random ϕ_i in $\{0°, 45°\}$*

2.2 *$\mathcal{R}$ chooses a random θ_i in $\{0°, 45°\}$*

2.3 *($\mathcal{S}$,$\mathcal{R}$) run $QT(\phi_i + r_i \times 90°, \theta_i)$, $\mathcal{R}$ gets ©*

2.4 *$\mathcal{S}$ sends ϕ_i to $\mathcal{R}$*

2.5 *$\mathcal{R}$ sets $©_i \leftarrow (\phi_i = \theta_i), r'_i \leftarrow ©_i \wedge ©$*

3: ...

4.7.2 Correctness and Privacy of the Reduction

Theorem 4.4 *Protocol 4.3 constitutes a statistically correct and private reduction of $\binom{2}{1}$–OBT to QT.*

Proof. Follows from Lemma 4.5 and 4.7. □

Correctness

First we show that the above protocol is correct.

Lemma 4.5 *Protocol 4.3 is statistically correct*

Proof. As long as one accept that protocol 4.2 is a correct implementation of OBT, then the reduction 4.3 is correct just as reduction 3.1. □

Privacy

Now, we show that this protocol is private. Just like the reduction of Chapter 3, its $\mathcal{S}$-private is trivial, since the only data $\mathcal{S}$ ever receives through this protocol are the two sets N_0 and N_1 that contain random indices. Proving $\mathcal{R}$-privacy is trickier. Consider the following simulator for any program $\widetilde{\wp_{\mathcal{R}}}$:

Protocol 4.4 (simulation$(\widetilde{\wp_{\mathcal{R}}}, c, 1^s;)$ **)**

1: *M chooses at random s bits* $r_1, r_2, ..., r_s$

2: *M executes* $\widetilde{\wp_{\mathcal{R}}}$ *on inputs* $c, 1^s$; *until she is ready to perform Step 2 of protocol 4.3*

3: $\mathbf{DO}_{i=1}^{s}$

3.1 *M chooses a random* ϕ_i *in* $\{0°, 45°\}$

3.2 *M runs* $\widetilde{\wp_{\mathcal{R}}}$ *until she sends* θ_i

3.3 *M runs* $QT(\phi_i + r_i \times 90°, \theta_i)$. *M gets* © *and uses it with* $\widetilde{\wp_{\mathcal{R}}}$

3.4 *M sends* ϕ_i *to* $\widetilde{\wp_{\mathcal{R}}}$

3.5 *M sets* $\Psi_i \leftarrow 2\cos^2(\phi_i + r_i \times 90° - \theta_i) - 1$, *i.e.*, $2\ \mathrm{Prob}(© = r_i) - 1$

4: *M runs* $\widetilde{\wp_{\mathcal{R}}}$ *until she sends* N_0 *and* N_1

5: **if** N_0 *and* N_1 *are not disjoint or of the right size* **then** *M stops*

6: *M computes* c' *such that* $\left|\prod_{i \in N_{c'}} \Psi_i\right| \geq \left|\prod_{i \in N_{1-c'}} \Psi_i\right|$
(**if** *equal* **then** *picks* c' *at random)*

(continued on next page)

Protocol 4.4

7: *M calls the oracle to compute* $\binom{2}{1}$*–*$OBT((b_0, b_1), c')$ *and get* $b_{c'}$

8: *M computes* $\hat{b}_x = b_{c'} \oplus \bigoplus_{n \in N_x} r_n$ *for* $x = 0, 1$ *and uses* $\hat{b_0}, \hat{b_1}$ *with* $\widetilde{\wp_{\mathcal{R}}}$

9: *M runs* $\widetilde{\wp_{\mathcal{R}}}$ *until she completes her computation*

A Bit of Intuition

Basically, this simulator tries to determine as well as possible which of N_0 or N_1 is most correlated with the input bits. In other words, t tries to determine which of b_0 or b_1 would be more disclosed in an actual conversation. What we show about this simulation is that one of N_0 and N_1 (indeed $N_{1-c'}$) is not highly correlated.

The value computed at Step 3.5 is a measure of the correlation between the bit transferred and the bit sent. When this value is 1, it means that the bit received is necessarily the bit sent; when it is -1, it means that the bit received is necessarily the inverse of the bit sent; when it is 0, it means that the bit received is uncorrelated to the bit sent.

The product of the values computed at Step 2.5 is a measure of the correlation between the XOR of the bits received and the XOR of the bits sent. The value furthest from 0 is the more correlated (or anti-correlated) of the two.

The following proposition formalizes this idea:

Proposition 4.6 *Let* $b_1, b_2, ..., b_s$*,* $b'_1, b'_2, ..., b'_s$ *be* $2s$ *random bits. If for* $1 \leq i \leq s$*,* $\text{Prob}(b_i = b'_i) = p_i$*, then*

$$\mathrm{Prob}(\bigoplus_{i=1}^{s} b_i = \bigoplus_{i=1}^{s} b'_i) = \frac{1}{2} + \frac{\prod_{i=1}^{s}(2p_i - 1)}{2}$$

Proof. by induction on s. Let $P_s = \mathrm{Prob}(\bigoplus_{i=1}^{s} b_i = \bigoplus_{i=1}^{s} b'_i)$. For $s = 1$, we have that $P_s = \mathrm{Prob}(b_1 = b'_1) = p_1 = \frac{1}{2} + \frac{(2p_1-1)}{2}$. Now if $s > 1$, assume for an induction hypothesis that $P_{s-1} = \frac{1}{2} + \frac{\prod_{i=1}^{s-1}(2p_i - 1)}{2}$. Consider P_s:

$$\begin{aligned}
P_s &= \mathrm{Prob}(\bigoplus_{i=1}^{s} b_i = \bigoplus_{i=1}^{s} b'_i) \\
&= \mathrm{Prob}(\bigoplus_{i=1}^{s-1} b_i = \bigoplus_{i=1}^{s-1} b'_i \wedge b_s = b'_s \vee \bigoplus_{i=1}^{s-1} b_i \neq \bigoplus_{i=1}^{s-1} b'_i \wedge b_s \neq b'_s) \\
&= P_{s-1}p_s + (1 - P_{s-1})(1 - p_s) \\
&= (\frac{1}{2} + \frac{\prod_{i=1}^{s-1}(2p_i - 1)}{2})p_s + (\frac{1}{2} - \frac{\prod_{i=1}^{s-1}(2p_i - 1)}{2})(1 - p_s) \\
&= \frac{1}{2} + \frac{\prod_{i=1}^{s}(2p_i - 1)}{2}
\end{aligned}$$

□

This basically says that the product $\prod_{i=1}^{s}(2p_i - 1)$ determines exactly how correlated $\bigoplus_{i=1}^{s} b_i$ and $\bigoplus_{i=1}^{s} b'_i$ are. The closer $\prod_{i=1}^{s}(2p_i - 1)$ is to 0, the less correlated (or anticorrelated) $\bigoplus_{i=1}^{s} b_i$ and $\bigoplus_{i=1}^{s} b'_i$ are. Now we prove our main lemma:

Lemma 4.7 *Protocol 4.3 is statistically $\mathcal{R}$-private*

Proof.

First, notice by inspection of the protocol and the simulator that the only way they differ is that whereas $\hat{b}_0$ and $\hat{b}_1$ are correlated to $b_{c'}$ in the simulation, they are correlated to b_0 and b_1 in the protocol. But if we show that the less correlated of the two (the one associated with $N_{1-c'}$) has correlation no greater than an exponentially small bias, then the output of the simulator will be statistically indistinguishable from the actual conversations. Now consider the probability that this is not so:

Proposition 4.8 *There exists a positive constant $\alpha < 1$ such that*

$$\mathrm{Prob}(\left|\prod_{i \in N_{1-c'}} \Psi_i\right| \geq \alpha^s) \leq \alpha^s$$

Proof. Notice that since $\left|\prod_{i \in N_0 \bigcup N_1} \Psi_i\right| = \left|\prod_{i \in N_{c'} \bigcup N_{1-c'}} \Psi_i\right| = \left|\prod_{i \in N_{c'}} \Psi_i\right| \times \left|\prod_{i \in N_{1-c'}} \Psi_i\right|$ and that $\left|\prod_{i \in N_{c'}} \Psi_i\right| \leq \left|\prod_{i \in N_{1-c'}} \Psi_i\right|$, it suffice to show proposition 4.9 since

$$\mathrm{Prob}(\left|\prod_{i \in N_{1-c'}} \Psi_i\right| \geq \sqrt{\alpha}^s) \leq \mathrm{Prob}(\left|\prod_{i \in N_0 \bigcup N_1} \Psi_i\right| \geq \alpha^s) \leq \alpha^s \leq \sqrt{\alpha}^s$$

Proposition 4.9 *There exists a positive constant $\alpha < 1$ such that*

$$\mathrm{Prob}(\left|\prod_{i \in N_0 \bigcup N_1} \Psi_i\right| \geq \alpha^s) \leq \alpha^s$$

Proof.

The key insight is that for whatever angle θ $\mathcal{R}$ uses, the following holds:

$$\mathrm{Prob}\left(1 - \cos^2(22\frac{1}{2}^\circ) \leq \cos^2(\phi + b \times 90^\circ - \theta) \leq \cos^2(22\frac{1}{2}^\circ)\right) = \frac{1}{2}$$

This is because ϕ is chosen at random among $\{0^\circ, 45^\circ\}$ and because for all θ, one of $\cos^2\theta$ and $\cos^2(\theta + 45^\circ)$ satisfies the above inequality. From this we deduce that there is a constant $\beta = 2\cos^2(22\frac{1}{2}^\circ) - 1 = \sqrt{\frac{1}{2}}$ such that $0 < \beta < 1$ and such that

$$\mathrm{Prob}(-\beta < 2\cos^2(\phi_i + r_i \times 90^\circ - \theta_i) - 1 < \beta) = \frac{1}{2},$$

which implies that, for all $i \leq s$,

$$\mathrm{Prob}(-\beta < \Psi_i < \beta) = \frac{1}{2}$$

Now we use this to prove our proposition. Suppose that $\#\{i \in N_0 \bigcup N_1 | -\beta < \Psi_i < \beta\} \geq \frac{s}{12}$. Then

$$\left| \prod_{i \in N_0 \bigcup N_1} \Psi_i \right| \leq \beta^{\frac{s}{12}}$$

Therefore

$$\mathrm{Prob}(\left| \prod_{i \in N_0 \bigcup N_1} \Psi_i \right| \geq \beta^{\frac{s}{12}}) \leq \mathrm{Prob}(\#\{i \in N_0 \bigcup N_1 | -\beta < \Psi_i < \beta\} \leq \frac{s}{12})$$

But what is the probability of this?

Consider the following random variable:

$$\copyright_i = \begin{cases} 0 & \text{if } -\beta < \Psi_i < \beta \\ 1 & \text{if } |\Psi_i| \geq \beta \end{cases}$$

Then

$$\begin{aligned} \mathrm{Prob}(\#\{n \in N_0 \bigcup N_1 | -\beta < \Psi_n < \beta\} \leq \frac{s}{12}) &\leq \mathrm{Prob}(\sum_{i=1}^{s} \copyright_i \geq \frac{7s}{12}) \\ &= \mathrm{Prob}(\sum_{i=1}^{s} \copyright_i - \frac{s}{2} \geq \frac{s}{12}) \\ &\leq \mathrm{Prob}(|\sum_{i=1}^{s} \copyright_i - \frac{s}{2}| \geq \frac{s}{12}) \\ &\leq 2e^{-\frac{s}{144}} \quad \text{by theorem 3.3} \\ &\leq \left(\frac{e}{2}\right)^{-\frac{s}{144}} \quad \text{for large enough } s \end{aligned}$$

The proposition follows by taking $\alpha = \max\{ \sqrt[12]{\beta}, \sqrt[144]{\frac{2}{e}}\} = \sqrt[144]{\frac{2}{e}}$.

□

Proposition 4.8 leads to the following result, which is what we actually need to show statistical privacy:

Proposition 4.10 *There is a positive constant $\alpha < 1$ such that*

$$\left(|\text{Prob}(\bigoplus_{n \in N_{1-c'}} r'_n = \bigoplus_{n \in N_{1-c'}} r_n) - \frac{1}{2}| \geq \alpha^s \right) \leq \alpha^s$$

For all the cases where $\sum_{i=1}^{s} ©_i \leq \dfrac{7s}{12}$ the simulator will produce conversations that will differ from the actual ones with probability at most α^s. Only in the cases where $\sum_{i=1}^{s} ©_i > \dfrac{7s}{12}$ will the probabilities be substantially different, since the real $\mathcal{S}$ will provide $b_{1-c'}$ as well as $b_{c'}$, whereas M will provide $b_{c'}$ twice. Notice that if $b_0 = b_1$, the simulator will always provide the correct value since $b_{c'} = b_{1-c'}$.

For simplicity, define $\wp = \binom{2}{1}$–OBT. Let $\mathcal{V}_s = View_{\mathcal{R}}((\wp_{\mathcal{S}}, \widetilde{\wp_{\mathcal{R}}}), ((b_0, b_1), c, 1^s;))$ and $\mathcal{M}_s = M^{\wp((b_0,b_1),*)}(\widetilde{\wp_{\mathcal{R}}}, c, 1^s;)$. Then $\mathcal{V} = \{\mathcal{V}_s\}_{s \in N}$ and $\mathcal{M} = \{\mathcal{M}_s\}_{s \in N}$ are statistically indistinguishable since

Proposition 4.11

$$\sum_{u \in \Sigma^*} |\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{M}_s}(u)| < 4\alpha^s$$

for the α given by lemma 4.10 and all sufficiently large s.

Proof. Let u be a possible conversation. When $b_0 = b_1$ we have exactly that $\text{Prob}_{\mathcal{V}_s}(u) = \text{Prob}_{\mathcal{M}_s}(u)$. Let $\Sigma^*_\alpha = \{u \in \Sigma^* | \sum_{i=1}^{s} ©^u_i \geq \dfrac{7s}{12}\}$ (the value of the $©^u_i$s are implicitly defined by u). If $b_0 \neq b_1$ we get that $\text{Prob}_{\mathcal{M}_s}(u) = \text{Prob}_{\mathcal{V}_s}(1-u)$ where $1-u$ is the string identical to u except for the value of $\hat{b}_{c'}$, which is complemented. In this case we have

$$\sum_{u \in \Sigma^*} |\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{M}_s}(u)| =$$

$$\sum_{u \in \Sigma^*_\alpha} |\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{M}_s}(u)| + \sum_{u \in \Sigma^* - \Sigma^*_\alpha} |\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{V}_s}(1-u)|$$

but notice that $u \in \Sigma^* - \Sigma^*_\alpha$ implies that $|\text{Prob}_{\mathcal{V}_s}(u) - \text{Prob}_{\mathcal{V}_s}(1-u)| \leq \text{Prob}_{\mathcal{V}_s}(u) 2\alpha^s$ by proposition 4.10. Therefore

$$\sum_{u\in\Sigma^*} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| \leq \sum_{u\in\Sigma^*_\alpha} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| + \sum_{u\in\Sigma^*-\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u) 2\alpha^s$$

and since

$$\sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u), \sum_{u\in\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u) \leq -\mathrm{Prob}(u \in \Sigma^*_\alpha) \leq \alpha^s$$

and

$$\sum_{u\in\Sigma^*-\Sigma^*_\alpha} \mathrm{Prob}_{\mathcal{V}_s}(u) \leq 1 - \alpha^s$$

we have

$$\sum_{u\in\Sigma^*} |\mathrm{Prob}_{\mathcal{V}_s}(u) - \mathrm{Prob}_{\mathcal{M}_s}(u)| \leq 2\alpha^s + (1-\alpha^s)2\alpha^s \leq 4\alpha^s$$

□

The proof of Lemma 4.7 and therefore of Theorem 4.4 follows immediately from this.

4.8 Practical Considerations

Now that we have proved our main theorems, let us digress from our main topic (protocol reductions) for a small journey into the practical world. From now on, we abandon the rigor we have used so far, in order to present more results with fewer proofs. In the next subsections we will argue the correctness of our results but not prove them formally. We investigate what can be achieved using real devices (as long as quantum mechanics is right about reality). We analyze how we can adjust our protocol to address the problems, mentioned in section 4.4, of compatibility between our model and reality.

4.8.1 Delayed Evaluation

In the our model of computation, the receiver must evaluate the predicate as soon as the data from the sender is available, since the sender can check whether or not the receiver has executed the abstract protocol. In a physical implementation, the receiver could (in principle) "save" photons and not read them until later. This possibility reveals a problem with protocol 4.3: if, by some trick, $\mathcal{R}$ manages to delay the output of the transfer from Step 3 until after Step 4 is performed, she would have a way to get $\mathcal{S}$'s bit all the time. Unfortunately this is clearly possible.

Quantum mechanics does not rule out the possibility of saving a photon in a given polarization state. In the physical setting, $\mathcal{R}$ gets the output of Step 3 when she runs an experiment to determine the polarization of a given photon carrying the information of ϕ. To cheat, she could let the photon travel through a vacuum long enough, or just bounce it between mirrors (although this is not technically known to be possible) until Step 4 is performed.

In order to prevent her from doing this, we would like to force her to "prove" that she has performed Step 3 correctly before going on to Step 4. Actually, we do not force her to prove it but, rather we ask her to commit to what she read in Step 3 and with probability $\frac{1}{2}$ we ask her to reveal what she read and not go through with Step 4 at all. If what she reveals is consistent with what was sent, $\mathcal{S}$ will be willing to continue the interaction, otherwise $\mathcal{R}$ will have been caught cheating and $\mathcal{S}$ will stop talking to her.

To accomplish this we need a way for $\mathcal{R}$ to commit to the bits read in Step 3. We need a *commitment scheme*. We use the properties of the quantum channel to build this primitive too.

4.8.2 Two Bit Commitment Schemes

Bennett and Brassard presented in [BB84] a coin-flipping protocol based on the quantum transfer that can be easily converted to a bit commitment scheme.

E-P-R

As mentioned in [BB84] their bit commitment scheme can be defeated in theory by the consequences of the Einstein-Podolsky-Rosen paradox. The interested reader can consult their paper for a more detailed explanation. Nevertheless, in practice, this kind of attack is improbable because the technology necessary to perform it is far beyond the current technology available, and because any small failure in the cheater's technology will result in the loss of his of cheating ability.

In a forthcoming paper, the author describes a commitment scheme which does not suffer from E-P-R.

Coherent Measurements

Another possibility not ruled out by quantum mechanics is evaluating predicates of more than one photon (coherent measurements).

This would obviate the commitment scheme mentioned above if the $\oplus$ of the values carried by many photons could be evaluated at once. This also applies to the reduction 4.3 for reducing $\binom{2}{1}$–OBT to the quantum transfer.

As far as the author knows, this is the only possible way to cheat our protocol. Fortunately, not only is there no technology available for coherent measurements, but physicists do not even know how to get photons to interact in ways that could lead to such measurements. Therefore, although such a possibility exists in principle, there should be very little concern that the protocol be broken this way in practice.

4.8.3 Conclusion

The reader interested in the practical implementation of the quantum transfer should consult Bennett and Brassard's article on the topic [BB89]. As this thesis is being completed, experiments leading to physical implementation of the transfer are being run by Bennett and Brassard. Their results are extremely promising. At this time, their device can do a quantum transfer over a distance of 30 centimeters with a crossover frequency of about 6.6%. Using subsequent protocols they can remove all errors if the parties are honest. Consult section 6.1 for open problems about errors in the quantum transfer.

The limiting technological factor at this time is the transmission of polarized photons over long distances without changing their polarization. Bennett and Brassard are optimistic that such technology will become available.

Chapter 5

Further Results on Oblivious Transfer Technology

In this chapter we mention some related results. Some of which are extensions of those presented so far.

5.1 A Weaker Form of Oblivious Transfer

Crépeau and Kilian show that even imperfect OBT can be used to implement $\binom{2}{1}$–OBT. By not imperfect we mean that the outcome of the transfer can be wrong with some fixed probability $p < \frac{1}{2}$.

Definition 5.1 (error(p)-OBT)

protocol specification:

$\mathcal{S}$	b	
$\mathcal{R}$		$(\copyright, b') \in_R \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b) & : \frac{1-p}{2} \\ (1,1-b) & : \frac{p}{2} \end{cases}$
common	p	

Techniques similar to but more complicated than those of chapter 3, yield the following result:

Theorem 5.2 ([CK88]) *For any fixed $p < \frac{1}{2}$ there exists a statistically correct and private reduction of $\binom{2}{1}$–OBT to error(p)-OBT.*

The very powerful tools of probability theory used by this reduction can be applied to obtain the following much more surprising result.

5.2 Correct and Private Oblivious Transfer from Random Noise

The result of chapter 4 is only one of the implementations that we have achieved from a physical primitive. We obtain a similar result based on the existence of a noisy channel.

Consider the following transfer:

Definition 5.3 (p–NT)

protocol specification:

$\mathcal{S}$	b	
$\mathcal{R}$		$b' \in_R \begin{cases} b & : 1-p \\ 1-b & : p \end{cases}$
common	p	

This transfer models to a symmetric noisy communication channel, in which a bit is flipped with probability $p < \frac{1}{2}$. It is a very natural primitive that has been extensively studied. The entire field of coding theory has been developed around this model.

This definition leads to a result similar to that of chapter 4.

Theorem 5.4 ([CK88]) *For any fixed $p < \frac{1}{2}$ there exists a statistically correct and private reduction of $\binom{2}{1}$–OBT to p–NT.*

It seems paradoxical that noise can be made so useful. Surprisingly, it can be domesticated and be used in a very constructive way as long as it is stable enough.

Something looking like OBT can be pretty easily achieved using p–NT. Suppose that $\mathcal{S}$ want to obliviously transfer a bit b to $\mathcal{R}$. He can use p–NT on the string bb, in which case $\mathcal{R}$ receives either bb or $\overline{bb}$ with total probability $p^2 + (1-p)^2$, which she interprets as receiving b, and she receives either $\overline{b}b$ or $b\overline{b}$ with total probability $2p(1-p)$, which she interprets as not receiving b (since her two readings are contradictory).

$\mathcal{S} \setminus \mathcal{R}$	$(1,0)$	$(1,1)$	$(0,0)$
00	$(1-p)^2$	p^2	$2p(1-p)$
11	p^2	$(1-p)^2$	$2p(1-p)$
01, 10	$p(1-p)$	$p(1-p)$	$p^2+(1-p)^2$

Figure 5-1: Input-Output Relation Through a Noisy Transfer

This is just a OBT in which $\mathcal{R}$ receives $\mathcal{S}$'s bit with probability $p^2 + (1-p)^2$ and receives nothing with complement probability $2p(1-p)$. The only difference is that the bits received have non-zero probability of being wrong. To summarize, the protocol is:

Protocol 5.1 (Honest-Weak-OBT$(b,,p)$ **)**

1: *($\mathcal{S}$,$\mathcal{R}$) run p–NT$(b,,p)$ twice. $\mathcal{R}$ gets b'_0 on the first run and b'_1 on the second run.*

2: *$\mathcal{R}$ outputs* $((b'_0 = b'_1), (b'_0 = b'_1) \wedge b'_0)$

Starting from this basic idea, there are two problems to deal with. One is the possibility that $\mathcal{S}$ uses a string $b\overline{b}$ as input, and the other is to the wrong bits transmitted to $\mathcal{R}$. In the first case, $\mathcal{S}$ just increases the probability of the result insignificance from

$2p(1-p)$ to $p^2+(1-p)^2$. Refer to [CK88] for further details. The complete version of this protocol will appear shortly in a separate paper of Crépeau and Kilian.

The hypothesis behind this primitive, although simpler than that of chapter 4, is actually harder to satisfy. The most difficult condition to fulfill is that the channel not be alterable by any of the parties. If, for instance, $\mathcal{S}$ could reduce the probability p to $p' < p$, he could artificially bring it back to p by feeding in some "bad" bits (sent as 01 or 10). This change would be transparent to $\mathcal{R}$ and would let him get data about c.

Because of this sensitivity and because of the high exponent in the running time of the proposed reduction (about $O(s^{20})$), this implementation is more of an intellectual exercise. The implementation of chapter 4 is far more practical.

5.3 An Extensions of Oblivious Transfer: $\binom{n}{1}$–OST

In order to motivate the broad interest in oblivious transfer protocols, we mention two more complicated protocols known to reduce to $\binom{2}{1}$–OBT.

Definition 5.5 ($\binom{n}{1}$–OST)

protocol specification:

$\mathcal{S}$	$(S_1, S_2, ..., S_n)$	
$\mathcal{R}$	c	S_c

In this transfer, $\mathcal{S}$ has n secret strings $S_0, S_1, \ldots, S_n$. $\mathcal{R}$ has a selection number, $c \leq n$. At the end of the transfer, the following three conditions hold.

1. $\mathcal{R}$ learns the value of S_c.

2. $\mathcal{R}$ gains no information about the value of $S_{c'}, c' \neq c$.

3. $\mathcal{S}$ learns nothing about the value of c.

This transfer is also known as ANDOS (for "all or nothing disclosure of secrets") [BCR86].

Theorem 5.6 ([BCR86]) *There exists a correct and private reduction of* $\binom{n}{1}$*–OST to* $\binom{2}{1}$*–OBT.*

Remark: One remarkable property of this reduction is that it does not involve an exponentially small (in s) probability of failure. It is an interesting open problem to accomplish the other reductions without this chance of failure as well (make the reduction absolutely correct and private instead of statistically correct and private).

This protocol is an important primitive in the design of more complex protocols. It is a key step in the following result.

5.4 Application of Oblivious Transfer: Oblivious Circuit Evaluation

Definition 5.7 (Oblivious Circuit Evaluation)
protocol specification:

$\mathcal{S}$	x	
$\mathcal{R}$	y	$f(x,y)$
common	f	

In this transfer, $\mathcal{S}$ and $\mathcal{R}$ agree on a circuit that computes a function f of their choice. $\mathcal{S}$ has a secret string x, and $\mathcal{R}$ has a secret string y. At the end of the transfer, the following three conditions hold.

1. $\mathcal{R}$ learns the value of $f(x,y)$.

2. $\mathcal{R}$ gains no further information about the value of x.

3. $\mathcal{S}$ learns nothing about the value of y.

Theorem 5.8 ([Kil88],[Cré89]) *There exists a statistically correct and private reduction of Oblivious Circuit Evaluation to* $\binom{2}{1}$*–OBT.*

Oblivious Circuit Evaluation is the most general natural extension of oblivious transfer. For instance it includes the class of zero-knowledge proofs of language membership [GMR89]. Given that two parties can do OCE it is very easy for the sender to prove that a string is in a language by letting f be a verifying circuit and x a string in the language.

In some ways, the theorem above is the most powerful theorem that can be established about oblivious transfers. It does not mean that research in this direction is complete. For one thing, the current set of reductions from p–NT to OCE requires $O(s^{25})$ operations. More research must be done before these results become practical.

Chapter 6

Open Problems and Conclusion

6.1 Open Problems

Here are a few open problems related to the material in this thesis. The following variations are more likely to be implementable in practice. It is still an open problem to achieve $\binom{2}{1}$–OBT from one of the following primitives:

Definition 6.1 (error($< p$)-OBT)
protocol specification:

$\mathcal{S}$	b	
$\mathcal{R}$		$(\copyright, b') \in_R \begin{cases} (0,0) & : \frac{1}{2} \\ (1,b) & : > \frac{1-p}{2} \\ (1,1-b) & : < \frac{p}{2} \end{cases}$
common	p	

This transfer is the same as OBT except that an external source of errors will flip the output in an arbitrary way with probability at most p. $\mathcal{S}$ and $\mathcal{R}$ know the upper bound p, but do not know the real value of the probability of error exactly. It could even vary at times but never exceed p. The same modification can be made to Quantum Transfer. This primitive is actually more realistic because there always is a small probability of error in the quantum transfer which cannot be guaranteed to be stable. Based on this protocol Bennett and Brassard have proposed protocols for the secret key exchange problem and coin tossing by telephone in [BB84].

Another open problem is how to handle a source of non-uniform noise that has probability of error at most p. The probability of error may vary with time and cannot be predicted. The only assumption is that this probability never exceeds p.

Definition 6.2 ($<p$–**NT**)
protocol specification:

$\mathcal{S}$	b	
$\mathcal{R}$		$b' \in_R \begin{cases} b & : \; > 1-p \\ 1-b & : \; < p \end{cases}$

Solving this kind of abstract problem in a very general way will lead to a much larger set of possible implementation. Relaxing the constraints of a protocol makes it easier to find a natural primitive that fulfills the specification.

6.2 Conclusion

The study of two party cryptographic protocols is fascinating. We hope that the formalism in this thesis will be a useful tool for further research in the field.

The main research directions we envision on the practical side of the field are the search for other natural primitives, the elaboration of more efficient reductions, and the elimination of the error probabilities (even if exponentially small) from the existing reductions.

From the theoretical point of view it is important to consider synchronization issues and integrate them with the notion of secure cryptographic protocols. Similar formalization should be achieved for more complex models like two party protocols in presence of eavesdroppers, or multi-party protocols.

Bibliography

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179. IEEE, 1984.

[BB89] C. H. Bennett and G. Brassard. The dawn of a new era for quantum cryptography: the experimental prototype works! *SIGACT News*, 20(4), 1989.

[BBBW83] C. H. Bennett, G. Brassard, S. Breidbard, and S. Wiesner. Quantum cryptography. In *Proceedings CRYPTO 82*, pages 267–275. Plenum Press, 1983.

[BC89] G. Brassard and C. Crépeau. Everything in np can be *argued* in *perfect* zero-knowledge. In *13th International Colloquium on Automata, Languages and Programming*, Lecture Notes on Computer Science. Springer-Verlag, 1989.

[BCR86] G. Brassard, C. Crépeau, and J.–M. Robert. Information theoretic reductions among disclosure problems. In 27^{th} *Symp. of Found. of Computer Sci.*, pages 168–173. IEEE, 1986.

[BG89] D. Beaver and S. Goldwasser. In *Advances in Cryptology: Proceedings of Crypto '89*, Lecture Notes on Computer Science. Springer-Verlag, 1989.

[Blu81] M. Blum. Three applications of the oblivious transfer: Part i: Coin flipping by telephone; part ii: How to exchange secrets; part iii: How to send certified electronic mail. Technical report, Department of EECS, University of California, Berkeley, CA, 1981.

[Blu84] M. Blum. Independent unbiased coin flips from a correlated biased source: A finite state Markov chain. In *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science*, pages 425–433, Singer Island, 1984. IEEE.

[BOGMR85] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A fair protocol for signing contracts. In *ICALP*, pages 43–52, 1985.

[BVV84] M. Blum, U. Vazirani, and V.Vazirani. Reducibility among protocols. In *Advances in Cryptology: Proceedings of Crypto '83*, pages 137–146. Plenum Press, 1984.

[CF85] J. D. Cohen and M. J. Fischer. A robust and verifiable cryptographically secure election scheme. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 372–382, Portland, 1985. IEEE.

[Cha88] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[CK88] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In 28^{th} *Symp. on Found. of Computer Sci.* IEEE, 1988. submitted to FOCS '88.

[Cle89] R. Cleve. In *Advances in Cryptology: Proceedings of Crypto '89*, Lecture Notes on Computer Science. Springer-Verlag, 1989.

[Cré88] C. Crépeau. Equivalence between two flavours of oblivious transfers (abstract). In C. Pomerance, editor, *Advances in Cryptology: Proceedings* *of Crypto '87*, volume 293 of *Lecture Notes on Computer Science*, pages 350–354. Springer-Verlag, 1988.

[Cré89] C. Crépeau. Verifiable disclosure of secrets and application. In *Advances in Cryptology: Proceedings of Eurocrypt '89*, Lecture Notes on Computer Science. Springer-Verlag, 1989.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, November 1976.

[EGL83] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. In R. L. Rivest, A. Sherman, and D. Chaum, editors, *Proceedings CRYPTO 82*, pages 205–210, New York, 1983. Plenum Press.

[FMR84] M. Fischer, S. Micali, and C. Rackoff. A secure protocol for the oblivious transfer, 1984. presented at EuroCrypt 84, the only manuscript available is the extended abstract submitted to the conference.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 133:792–807, 1986.

[GHY85] Z. Galil, S. Haber, and M. Yung. A private interactive test of a boolean predicate and minimum-knowledge public-key cryptosystems. In *Proceedings of the* 26*th IEEE Symposium on Foundations of Computer Science*, pages 360–371, Portland, 1985. IEEE.

[GM82] S. Goldwasser and S. Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proc.* 14*th ACM Symposium on Theory of Computing*, pages 365–377, San Francisco, 1982. ACM.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28(2):270–299, April 1984.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. *SIAM. J. Computing*, 18(1):186–208, February 1989.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or: A completeness theorem for protocols with honest majority. In *Proc. 19th ACM Symposium on Theory of Computing*, pages 218–229, New York City, 1987. ACM.

[Kil88] J. Kilian. Founding cryptography on oblivious transfer. In *Proc. 20th ACM Symposium on Theory of Computing*, pages 20–31, Chicago, 1988. ACM.

[KMR] J. Kilian, S. Micali, and P. Rogaway. personal communication, to appear shortly.

[Kra86] E. Kranakis. Primality and cryptography. John Wiley and Sons, 1986.

[LMR83] M. Luby, S. Micali, and C. Rackoff. How to simultaneously exchange a secret bit by flipping a symmetrically biased coin. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science*, pages 11–22, Tucson, 1983. IEEE.

[MH78] R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525–530, September 1978.

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Harvard Aiken Computation Laboratory, 1981.

[RSA78] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communcations of the ACM*, 21:120–126, February 1978.

[SRA81] A. Shamir, R. L. Rivest, and L. M. Adleman. Mental poker. In D. Klarner, editor, *The Mathematical Gardner*, pages 37–43. Wadsworth, Belmont, California, 1981.

[Ted85] T. Tedrick. Fair exchange of secrets. In *Advances in Cryptology: Proceedings of Crypto '84*, Lecture Notes on Computer Science, pages 434–438. Springer-Verlag, 1985.

[Wie83] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. manuscript written *circa* 1970, unpublished until it appeared in SIGACT News.

[Yao82] A. C. Yao. Theory and application of trapdoor functions. In *Proceedings of the* 23*rd IEEE Symposium on Foundations of Computer Science*, pages 80–91, Chicago, 1982. IEEE.

[Yao87] A. C. Yao. Toronto, 1987. IEEE. lecture presented at the Mathematical Sciences Research Institute, Berkeley California, May 1987.