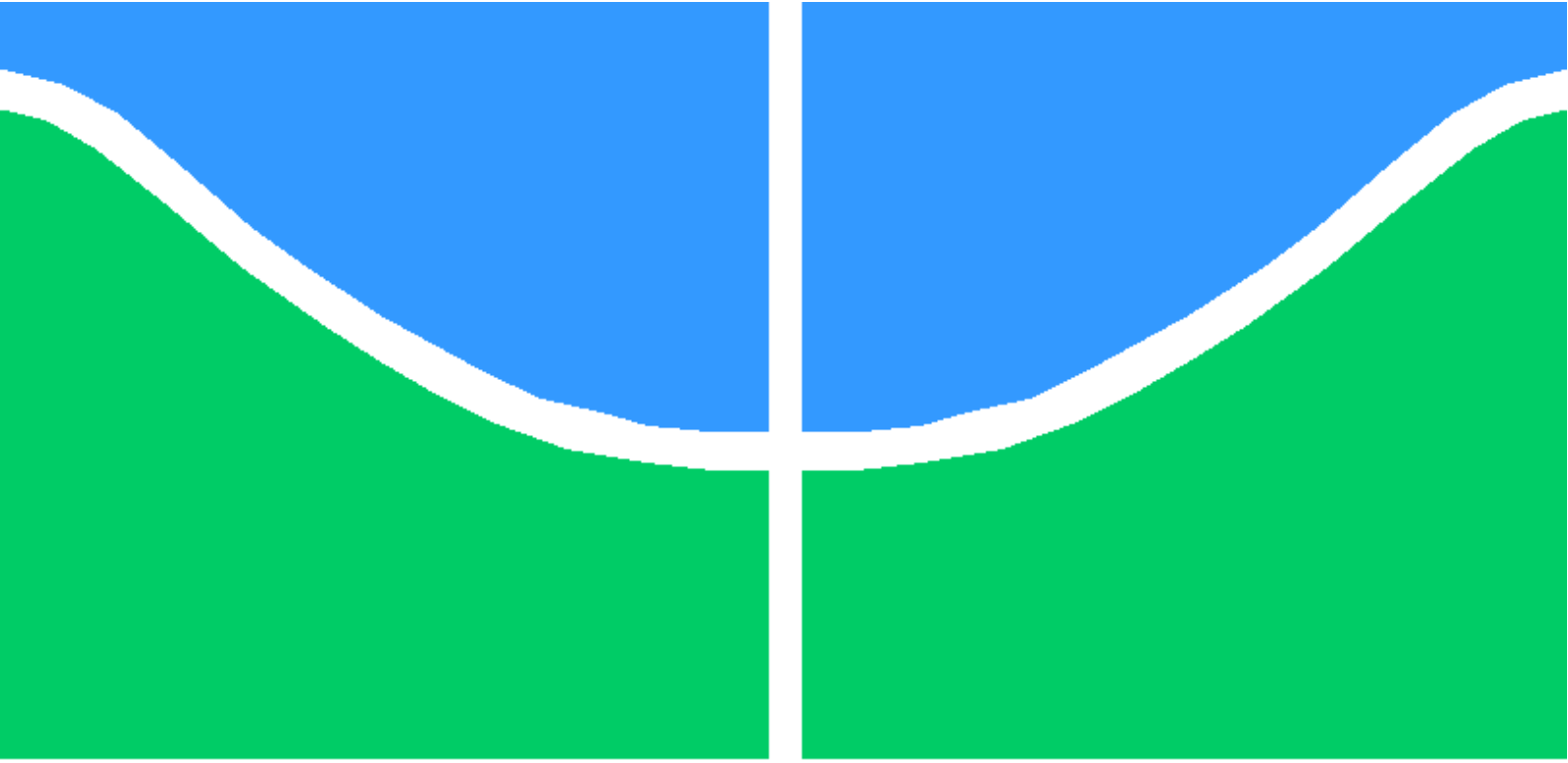

TRABALHO DE GRADUAÇÃO

ACHIEVING OBLIVIOUS TRANSFER CAPACITY OF GENERALIZED ERASURE CHANNELS IN THE MALICIOUS MODEL

Adriana Cristina Bastos Pinto

Brasília, dezembro de 2009

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

ACHIEVING OBLIVIOUS TRANSFER CAPACITY OF GENERALIZED ERASURE CHANNELS IN THE MALICIOUS MODEL

Adriana Cristina Bastos Pinto

Relatório submetido ao Departamento de Engenharia Elétrica
como requisito parcial para obtenção do grau de
Engenheiro de Redes de Comunicação

Banca Examinadora

Anderson C. A. Nascimento - Ph.D., UnB/ENE (Orientador) _____

Jeroen van de Graaf - Ph.D., UFOP (Membro Externo) _____

FICHA CATALOGRÁFICA

PINTO, ADRIANA CRISTINA BASTOS. Achieving Oblivious Transfer Capacity of Generalized Erasure Channels in The Malicious Model [Distrito Federal] 2009. iii, 46p. (ENE/FT/UnB, Engenheiro de Redes de Comunicação, 2009) Monografia de Graduação - Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Criptografia
2. Information-theoretic security
3. Oblivious Transfer Capacity
4. Generalized Erasure Channel

I. ENE/FT/UnB
II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

PINTO, ADRIANA CRISTINA BASTOS (2009). Achieving Oblivious Transfer Capacityof Generalized Erasure Channels in The Malicious Model Monografia de Graduação, Publicação ENE 01/2009, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 46p.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: Adriana Cristina Bastos Pinto

TÍTULO: Achieving Oblivious Transfer Capacityof Generalized Erasure Channels in The Malicious Model

GRAU / ANO: Engenheiro de Redes de Comunicação / 2009.

Dedication

I dedicate this work to my family for supporting me in all these years of study.

Adriana Cristina Bastos Pinto

Acknowledgements

I want to express my gratitude to my advisor Anderson Clayton Alves Nascimento, my collegue Rafael Baião Dowsley and the professor Kiril Morozov who provided me assistance in numerous ways and made this work possible.

Adriana Cristina Bastos Pinto

RESUMO

Oblivious Transfer (OT) é uma das primitivas criptográficas principais, uma vez que pode-se implementar computação de duas partes (e múltiplas partes) a partir dele. Neste trabalho, será mostrado que oblivious transfer de strings, seguro no senso da teoria da informação, pode ser construído baseado em canal "com apagamentos" generalizado (GEC) – que é uma combinação de um canal discreto sem memória (DMC) com um canal "com apagamentos". Em seguida, será apresentado um protocolo que atinge a capacidade de OT frente à adversários maliciosos para um canal GEC com probabilidade de ocorrer "apagamentos" de ao menos 1/2, i.e. o protocolo explora o canal com a máxima eficiência para o oblivious transfer de strings na presença de adversários maliciosos.

ABSTRACT

Oblivious Transfer (OT) is one of the central cryptographic primitives since one can implement secure two-party (and multi-party) computation from it. In this work we show that information-theoretically secure string oblivious transfer can be constructed based on the generalized erasure channel (GEC), which is a combination of a discrete memoryless channel (DMC) with the erasure channel. Then, we presents a protocol that achieves the OT capacity against malicious adversaries for the GEC with erasure probability at least 1/2, i.e. the protocol exploits the channel with maximum efficiency for secure oblivious transfer of strings against malicious adversaries.

CONTENTS

1 INTRODUCTION

Oblivious Transfer (OT) is a cryptographic primitive by which the sender transmits some information to the receiver, without knowing precisely which information was received. It was initially proposed in different flavors [1, 2] which later turned out to be equivalent [3]. OT is a very powerful tool since it implies secure two-party (and multi-party) computation [4, 5, 6]. Namely, a secure multi-party computation [7] allows two or more distrustful players to jointly compute a function of their inputs without learning more than what they can extract from their own input and output. So, using OT as a subprotocol, one can implement any secure multi-party computation like Oblivious Circuit Evaluation, Mental Games and so on. Therefore, improvements in Oblivious Transfer are of great importance for cryptography. In this work we will consider the *one-out-of-two string oblivious transfer*, string-OT, in which Alice transmits two input strings $r_0, r_1 \in \{0,1\}^k$ and Bobs uses a choice bit c to choose the string r_c that he will receive. This protocol ensures that a dishonest Alice cannot learn c, while a dishonest Bob cannot learn both r_0 and r_1.

The potential of noisy channels for implementing information-theoretically secure cryptographic protocols was first noted by the pioneering work of Wyner [8], with respect to secret key agreement. Crépeau and Kilian [9] proved that noisy channels can be used to implement oblivious transfer. Those results were later improved in [10, 11, 12, 13, 14]. We will implement oblivious transfer based in a Generalized Erasure Channel(GEC). This channel can be seen as a combination of a discrete memoryless channel and a erasure channel since the output and the input of the channel is uncorrelated with certain probability (this behavior resembles an erasure in a erasure channel) and otherwise the output will behavior in accordance with a discrete memoryless channel associated. We call the reader's attention to the fact that a generalized erasure channel, in fact, is a general channel, since the bits transmitted can be erasure and flipped according to the channel. Then, we will work with a general primitive in a general channel.

The question of determining the optimal rate at which oblivious transfer can be implemented using a noisy channel (i.e., the oblivious transfer capacity of the channel) was raised by Nascimento and Winter in [14]. Strictly speaking, oblivious transfer capacity is a measure of how efficiently one can implements oblivious transfer from a noisy resource which can be a noisy channel or a noisy distribution. Nascimento and Winter also characterized the noise distributions that provide strictly positive oblivious transfer capacity. Imai et al [15] obtained the oblivious transfer capacity of the Erasure Channels. Ahlswede and Csiszár [16] proved additional bounds to those capacities and also obtained the oblivious transfer capacity

of GECs in the passive adversary model (where the players always follow the protocol). The related notion of commitment capacity was proposed by Imai et al in [17].

The aim of this work is to show that the rates achieved in [16] against passive players can actually be achieved even against the malicious ones (i.e. those that can arbitrarily deviate from the rules specified by the protocol). As the upper bounds proved in [16] for the case of passive players still hold against active ones, we thus establish the oblivious transfer capacity of the Generalized Erasure Channels [16] in the malicious adversary model. Moreover, we prove security of our protocols using definitions by Crépeau and Wullschleger [18], which are known to imply sequential composability.

The main tool for obtaining our results is Interactive Hashing (IH), originally introduced by Ostrovsky et al [19]. Our solution is based on the protocol proposed by Savvides [20] (building on the results of [21]) for oblivious transfer from erasure channels that employs information-theoretic Interactive Hashing [22] as a sub-protocol. However, instead of directly adapting Savvides' solution to our scenario, we show that it is possible to use the interactive hashing protocol by Ding et al [23] and obtain a constant number of rounds, thus answering an open question posed by [20].

ORGANIZATION OF THIS WORK In the Chapter 2, we present the notation, some basic concepts and auxiliary results. Chapter 3 presents the concepts and definitions of two-party protocols. Chapter 4 presents Interactive Hashing, its security and the protocol by Ding et al [23]. Chapter 5 presents Oblivious Transfer, the security definition of String OT, OT capacity and asserts our main theorem. In Chapter 6 we present a protocol that achieves the OT capacity of GEC in the malicious model and prove that this protocol follows the definition of security. In Chapter 7 we prove the main theorem and in Chapter 8, we discuss the conclusions of this work.

2 PRELIMINARIES

2.1 NOTATION

We will denote by calligraphic letters the domains of random variables and other sets; by $|\mathcal{X}|$ the cardinality of a set $\mathcal{X}$; by upper case letters the random variables and algorithms; and by lower case letters one realization of the random variable. For a random variable X over $\mathcal{X}$, we denote its probability distribution by $P_X : \mathcal{X} \to [0,1]$ with $\sum_{x\in\mathcal{X}} P_X(x) = 1$. For a joint probability distribution $P_{XY} : \mathcal{X} \times \mathcal{Y} \to [0,1]$, let $P_X(x) := \sum_{y\in\mathcal{Y}} P_{XY}(x,y)$ denote the marginal probability distribution and let $P_{X|Y}(x|y) := \frac{P_{XY}(x,y)}{P_Y(y)}$ denote the conditional probability distribution if $P_Y(y) \neq 0$. We write $X \in_R \mathcal{X}$ for a random variable uniformly distributed over $\mathcal{X}$.

If a and b are two bit strings of the same lenght; we denote by $a \oplus b$ their bitwise XOR. The logarithms used in this work are in base 2. The entropy X is denoted by $H(X)$ and the mutual information between X and Y by $I(X;Y)$. We write $[n]$ for $\{1,...,n\}$ and $\binom{[n]}{l}$ for the set of all subsets $\mathcal{K} \subseteq [n]$, where $|\mathcal{K}| = l$. For $X^n = (X_1, X_2, \ldots, X_n)$ and $\mathcal{S} \subset [n]$, we write $X^{\mathcal{S}}$ for the restriction of X^n to the positions in the subset $\mathcal{S}$. Similarly for a set $\mathcal{R}$, $\mathcal{R}^{\mathcal{S}}$ is the subset of $\mathcal{R}$ consisted of the elements determined by $\mathcal{S}$.

2.2 BASIC CONCEPTS

In this section we will introduce some basic concepts and auxiliary results used in subsequent chapters.

2.2.1 Generalized Erasure Channel

Aiming to define a generalized erasure channel we define a discrete memoryless channel and an erasure channel.

Definition 2.2.1 *A* discrete memoryless channel *is a system consisting of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$ and a probability transition matrix $W = [p(y|x)]$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The output does not depend on the initial state of the channel and $p(y|x)$ is conditionally independent of previous channel inputs or outputs, i.e., $p(y_1,...,y_n|x_1,...,x_n) = p(y_1|x_1)p(y_2|x_2)...p(y_n|x_n)$, $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.*

Definition 2.2.2 *An* erasure channel *is a channel $\{W : \mathcal{X} \to \mathcal{Y}\}$ such that the input is erased (lost) with a*

probability α *or the output is equal to the input with a probability* $1-\alpha$.

With these definitions, we can define a Generalized Erasure Channel(GEC).

Definition 2.2.3 ([16]) *A discrete memoryless channel* $\{W : \mathcal{X} \to \mathcal{Y}\}$ *will be called the* Generalized Erasure Channel (GEC) *if the output alphabet* $\mathcal{Y}$ *can be decomposed as* $\mathcal{Y}_0 \cup \mathcal{Y}^*$ *such that* $W(y|x)$ *does not depend on* $x \in \mathcal{X}$*, if* $y \in \mathcal{Y}^*$*. For a GEC, we denote* $W_0(y|x) = \frac{1}{1-p^*}W(y|x)$*,* $x \in \mathcal{X}$*,* $y \in \mathcal{Y}_0$*, where* p^* *is the sum of* $W(y|x)$ *for* $y \in \mathcal{Y}^*$ *(not depending on* x*).*

In other words, a GEC is a combination of a discrete memoryless channel and an erasure channel: with probability p^* the output will be such that its probability is independent of the input $x \in \mathcal{X}$, it is as if the input was erased, and the output was changed for one $y \in \mathcal{Y}^*$; and with probability $1-p^*$ the input will be transmitted in an discrete memoryless channel with channel matrix W_0. Notice that, in this definition, we do not know *a priori* the error probability.

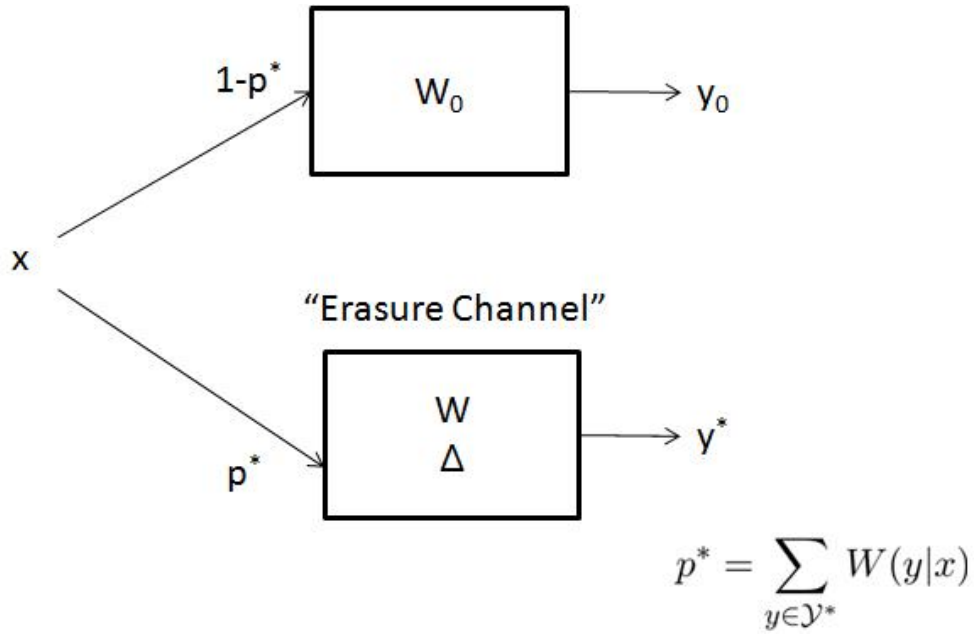

Figure 2.1: Generalized Erasure Channel

2.2.2 Entropy, Mutual Information and Statistical Information

We will use the following tools in our proofs. The *Shannon entropy* of a discrete random variable X is

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

and the *conditional Shannon entropy* of a discrete random variable Y given a discrete random variable Y is

$$H(Y|X) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x).$$

Some fruitful results of entropy are

$$H(X,Y) = H(X) + H(Y|X) \quad \textit{Chain rule for entropy},$$

its conditional version

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z),$$

and

$$H(X|Y) \leq H(X) \quad \textit{conditioning reduces entropy}.$$

Similarly, the *mutual information* of a discrete random variable X given Y is

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = H(X) - H(X|Y),$$

the *conditional mutual information* of discrete random variables X, Y given Z is

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$$

and

$$I(X,Y,Z;V) = I(X;V) + I(Y;V|X) + I(Z;V|XY) \quad \textit{Chain rule for mutual information}.$$

In order to build a security definition, Crépeau and Wullschleger [18] introduced the definition of Statistical Information. This measure was introduced in their paper with the intent to obtain the probability that an adversary gets any information at all.

Definition 2.2.4 *The* statistical distance *between two probability distributions P_X and P_Y over the same domain $\mathcal{V}$ is*

$$\mathsf{SD}(P_X, P_Y) := \frac{1}{2}\sum_{v\in\mathcal{V}} |P_X(v) - P_Y(v)|.$$

We say that a random variable X over $\mathcal{V}$ is ϵ-close to uniform with respect to Y if $\mathsf{SD}(P_{XY}, P_U P_Y) \leq$

ϵ, where P_U is the uniform distribution over $\mathcal{V}$.

Definition 2.2.5 *The* statistical information *of X and Y given Z is defined as*

$$I_S(X;Y|Z) = \mathsf{SD}(P_{XYZ}, P_Z P_{X|Z} P_{Y|Z}).$$

2.2.3 Typical Sequences and Jointly Typical Sequences

Here we will follow chapter 3 and 7 of [24]. The asymptotic equipartition property (AEP) states that for i.i.d. random variables X_i such that $-\frac{1}{n}\log p(X_1, X_2, ..., X_n)$ converges in probability to the entropy H(X). It means that

$$\forall \epsilon > 0, \Pr\left\{\left|-\frac{1}{\mathrm{n}}\log \mathrm{p}(\mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_\mathrm{n}) - \mathrm{H}(\mathrm{X})\right| > \epsilon\right\} \longrightarrow 0.$$

This result enables us to divide the set of all sequences X^n into two sets: the set of *typical sequences*, where the empirical entropy is close to the true entropy and the set of non-typical sequences. The power of AEP lies in the fact that the set of typical sequences is a highly representative set. In other words, the probability that a sequence belongs to the set of typical sequences is close to one.[1] In fact, the typical set is defined below:

Definition 2.2.6 *The* typical set a_ϵ^n *with respect to the probability mass function $p(x)$ is the set of sequences $(x_1, x_2, ..., x_n)$ with the property*

$$2^{-n(H(x)+\epsilon)} \leq p(x_1, x_2, ..., x_n) \leq 2^{-n(H(x)-\epsilon)}$$

Armed with this definition, we can state the follow theorem that show us some properties of typical sequences:

Theorem 2.2.7

1. $(x_1, x_2, ..., x_n) \in a_\epsilon^n \Rightarrow H(X) - \epsilon \leq -\frac{1}{n}\log p(x_1, x_2, ..., x_n) \leq H(X) + \epsilon$
2. $\Pr\{a_\epsilon^n\} > 1 - \epsilon$ *for n sufficiently large.*
3. $|a_\epsilon^n| \leq 2^{n(H(X)+\epsilon)}$

[1] Note that the typical set is representative relative to the probability. The number of typical sequences is fairly small set. See the theorem 2.2.7.

4. $|a_\epsilon^n| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$ *for* n *sufficiently large.*

Now, we can extend the concept of typical sequences to jointly typical sequences. The set A_ϵ^n of jointly typical sequences is the set of n-sequences x^n and y^n such that the empirical entropies are ϵ-close to the true entropies and true jointly entropy.

Definition 2.2.8 *The set* A_ϵ^n *of jointly typical sequences* $\{(x^n, y^n)\}$ *with respect to the distribution* $p(x, y)$ *is:*

$$A_\epsilon^n = \{(x^n, y^n)\} \in \mathcal{X}^n \mathrm{x} \mathcal{Y}^n : \left|-\frac{1}{n} logp(x^n) - H(X)\right| < \epsilon,$$

$$\left|-\frac{1}{n} logp(y^n) - H(Y)\right| < \epsilon,$$

$$\left|-\frac{1}{n} logp(x^n, y^n) - H(X,Y)\right| < \epsilon.$$

where

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i)$$

Similarly to the case of typical sequences, we can state the follow theorems:

Theorem 2.2.9 *Let* (X^n, Y^n) *be sequences of length* n *drawn i.i.d. according to*

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i).$$

Then:

1. $\mathrm{Pr}((\mathrm{X^n}, \mathrm{Y^n}) \in \mathrm{A_\epsilon^n}) \to 1$ *as* $n \to \infty$
2. *The number of the sequences* x^n *jointly typical with* y^n *is* $|A_\epsilon^n| \leq 2^{n(H(X,Y)+\epsilon)}$
3. *If* $(\tilde{X}^n, \tilde{Y}^n)$ *are independent of the same marginal probabilities as* X, Y*, i.e.* $p(\tilde{x}) = p(x)$ *and* $p(\tilde{y}) = p(y)$*, then:*

$$Pr\big((\tilde{X}^n, \tilde{Y}^n) \in A_\epsilon^n\big) \leq 2^{-nI(X;Y)-3\epsilon}.$$

Theorem 2.2.10 *Let* (X^n, Y^n) *be sequences of length* n *drawn i.i.d. according to*

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i).$$

Then the number of the sequences x^n *jointly typical with* y^n *is* $|A_\epsilon^n| \leq 2^{n(H(Y|X)+\epsilon)}$

Proof The proof follows that of item 2 of the last theorem.

$$\begin{aligned} 1 &= \sum p(x^n, y^n) \\ &\geq \sum_{A_\epsilon^n} p(x^n, y^n) \\ &\geq \sum_{A_\epsilon^n} p(y^n|x^n) \\ &\geq |A_\epsilon^n| 2^{-n(H(Y|X)+\epsilon)} \end{aligned}$$

Then, $|A_\epsilon^n| \leq 2^{n(H(Y|X)+\epsilon)}$ □

2.2.4 Strong extractors

Extractors were introduced in [25]. They are tools that act extracting randomness from a source with an arbitrary distribution using a small number of truly random bits. Aiming to define extractors, we need the concepts of statistical distance (definition 2.2.4), δ-source and min-entropy.

Definition 2.2.11 *A distribution D on $\{0,1\}^n$ is called a δ-source if $D(x) \leq 2^{-\delta n}$, $\forall x \in \{0,1\}^n$.*

The follow definition, by [25], states that from an input of n bits and using r truly random bits, the function Ext gives us an output of m bits with distribution ϵ-close to the uniform distribution.

Definition 2.2.12 *Let $Ext : \{0,1\}^n \times \{0,1\}^r \rightarrow \{0,1\}^m$. Ext is called a* (δ, ϵ)-extractor *if for every δ-source D, the distribution of $Ext(x,y) \circ y$ induced by choosing x from D and y uniformly in $\{0,1\}^r$ has a statistical distance similar than ϵ from the uniform distribution on $\{0,1\}^m \times \{0,1\}^r$.*

However, in this work we will use a version a little different of extractors [26]. This definition does not use Shannon entropy, it uses the min-entropy. Intuitively, if a distribution has min-entropy k, it is *"at least as random"* as the uniform distribution on k bit strings. [2]

Definition 2.2.13 *For a finite alphabet $\mathcal{X}$, the* min-entropy *of a random variable $X \in \mathcal{X}$ is defined as*

$$H_\infty(X) = \min_x \log(1/P_X(x)).$$

[2] In fact when we work with min-entropy, we are considering the worst case. Moreover, it is impossible to extract randomness from distributions which are not close to having high min-entropy (see [27])

Its conditional version, defined over $\mathcal{Y}$ with finite alphabet is

$$H_\infty(X|Y) = \min_y H_\infty(X|Y=y).$$

Definition 2.2.14 (Strong Randomness Extractors) *Let* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^l$ *be a probabilistic polynomial time function which uses r bits of randomness. We say that* Ext *is an efficient* $(n, m, l, \epsilon)-$strong extractor *if for all probability distributions P_W with $\mathcal{W} = \{0,1\}^n$ and $H_\infty(W) \geq m$, we have that* $\mathsf{SD}(P_{\mathsf{Ext}(W;U_r),U_r}, P_{U_l}P_{U_r}) \leq \epsilon$.

Note that a δ-source has min-entropy $\delta n \geq m$.

In [28] we have the result that strong extractors can extract at most $l = m - 2\log(\epsilon^{-1}) + O(1)$ bits of nearly random bits, and this optimal bound is achieved by a Universal Hash Function [29] which we define below.

Definition 2.2.15 (Universal Hash Function) *A class $\mathcal{G}$ of functions $\mathcal{A} \to \mathcal{B}$ is* 2-universal *if, for any distinct $x_1, x_2 \in \mathcal{A}$, the probability that $g(x_1) = g(x_2)$ is at most $|\mathcal{B}|^{-1}$ when g is chosen uniformly at random from $\mathcal{G}$.*

The Leftover-Hash Lemma (similarly the Privacy-Amplification Lemma) [30, 31, 32, 33, 26] guarantees that the Universal Hash Functions allow us to extract $l = m - 2\log(\epsilon^{-1}) + 2$ bits.

Lemma 2.2.16 (Leftover-Hash Lemma) *Assume that a class $\mathcal{G}$ of functions $G : \{0,1\}^n \to \{0,1\}^l$ is 2-universal. Then for G selected uniformly at random from $\mathcal{G}$ we have that*

$$\mathsf{SD}(P_{G(W),G}, P_{U_l}P_G) \leq \frac{1}{2}\sqrt{2^{-H_\infty(W)}2^l}.$$

In particular, Universal Hash Functions are $(n, m, l, \epsilon)-$strong extractors *whenever*

$$l \leq m - 2\log(\epsilon^{-1}) + 2$$

.

3 SECURE TWO-PARTY PROTOCOLS

In this chapter we present secure two-party protocols and related definitions of security. Specifically, we present security definitions of protocols in the real/ideal paradigm and the corresponding information-theoretic conditions for security on the malicious and semi-honest models. We closely follow the definitions presented by Crépeau and Wullschleger [18]. An excellent exposition of this theme can be found in [34].

3.1 ADVERSARIES AND PROTOCOLS

A two-party protocol consists of a pair of interactive, probabilistic algorithms. Such algorithms are formally defined as Interactive Turing machines. See more in [35].

Another way to see a two-party protocol is through a set formed by a specification of a random process and a pair of inputs and one of outputs. The random process describes a desired *functionality*, denoted by $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$, which maps the pair of inputs to the pair of outputs. Then, for every pair of inputs (x, y) and an associated probability distribution, the desired output pair is also a random variable $f(x, y)$. We stress that the output $f(x, y)$ is bipartite, i.e. it is composed of two parts, one in possession of the first player and another one in possession of a second player.

In order to achieve security in the case where protocols run one after another, we use hybrid-protocols. An $\mathcal{F}$-hybrid protocol consists of a pair of algorithms $P = (A_1, A_2)$, representing the parties which strategies can interact by means of two-way message exchange and have access to some functionality $\mathcal{F}$. We say that a pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ is admissible for protocol P if at least one of the parties is honest (follows exactly the protocol), that is, if $\widetilde{A_1} = A_1$ or $\widetilde{A_2} = A_2$.

Regarding the players, they can be either *honest* or *corrupted*. The honest ones are those that follow exactly the protocol and refuse to obtain any knowledge about information which they are not supposed to obtain. Corrupted players can further be classified as *active*, *passive*, *adaptive*, and *non-adaptive*. Active players (a.k.a malicious player) are those that can arbitrarily deviate from the rules specified by the protocol, i.e., they sends messages that differ from those specified by the protocol. Passive players (also called honest-but-curious or semi-honest player) are those that follow the protocol but gather information.

Adaptive players are those that may corrupt parties to the protocol, according to the information they have gathered so far, while the execution goes on. Conversely, non-adaptive players form a arbitrarily fixed set before the execution starts.

We call the set of all corrupted players of *adversary*. Formally, an adversary is defined as a probabilistic Turing machine. [1] In this work we consider non-adaptive and active adversaries. As we work with two-party protocols, we provide security when at most one of the players is corrupted.

3.2 DEFINITION OF SECURITY IN REAL/IDEAL PARADIGM

One way to define secure protocols is using the real/ideal paradigm which postulates that any effect of adversaries that participate in the execution of the real protocol can be achieved by simulation of a corresponding ideal protocol with corresponding adversaries. In this paradigm, a protocol is considered secure if running it is equivalent to all parties sending their inputs to a trusted party, who locally computes the outcome of the desired functionality and sends the appropriate output to each party.

More precisely, an ideal model describes the two parties send their inputs to a trusted party who locally computes the outcome. A real model describes the execution of a real two-party protocol where there exists no trusted party. Then, a protocol in the real model is called secure with respect to a certain adversary if the outcome of execution of the real protocol with such adversary and the outcome of execution on corresponding ideal model are indistinguishable.

To prove that a protocol in the real model is close enough to or indistinguishable from the corresponding ideal model, we can use some measures of similarity: “equal distribution", “statistical closeness" or “computational indistinguishability". We stress that each measure of similarity will produce a different notion of security. Whenever working with computationality unbounded adversaries, it is natural to use perfect indistinguishability or statistical indistinguishability. As perfect indistinguishability is a stronger requirement we will use the notion of statistical indistinguishability. It means that the probability distribution of the outcome of the protocol in the real model is ϵ-close to the probability distribution of the outcome in the corresponding ideal model.

We will use the construction of the real/ideal model from [34].

[1] A probabilistic Turing machine is a Turing machine in which some transitions are randomly chooses between the available ones according to some probability distribution. Then, for the same input and instruction a probabilistic TM may have different reactions, yielding different outputs.

3.2.1 Security in the Honest-But-Curious Model

The ideal model The protocol in the ideal model that both parties send their inputs to the trusted party, who computes the corresponding output pair and sends each output to the corresponding party. The honest-but-curious adversary tries to learn something from his view of the execution. It means that he can only determine his own output based on his input and the output he has received from the trusted party. The output of honest party is the one received from the trusted party.

In order to formalize the output in the ideal model, let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a desired functionality, and let $(u, v) = f(x, y)$ be the outcome of the functionality. Given an f-hybrid protocol $B = (B_1, B_2)$ and pair of algorithms $\widetilde{B} = (\widetilde{B_1}, \widetilde{B_2})$ which is admissible in the ideal model for the protocol B, the joint execution of B under $\widetilde{B}$ on input pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and auxiliary input $z \in \{0,1\}^*$ (see section 3.4 to more explanations of the necessity of this auxiliary input) in the ideal model is defined as

$$IDEAL_{f,\widetilde{B}(z)}(x,y) \triangleq \begin{cases} (f(x,y), B_1(x,u,z), B_2(y,v)) & \text{if the corrupted player is Player 1} \\ (f(x,y), B_1(x,u), B_2(y,v,z)) & \text{if the corrupted player is Player 2} \end{cases}$$

The real model The two-party protocol in this model is executed properly, but the honest-but-curious adversary may produce his own output based on his view of the execution.

Similarly to the previous case, let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a desired functionality and let Π be a two-party real protocol for computing f. The view of the first player during an execution of Π on (x, y), denoted by $VIEW_1^{\Pi}(x, y)$, is $(x, m_1, \ldots, m_2)$, where m_i represents the i-th message it has received. The view of the second player is defined similarly. However, the output of execution of Π on (x, y) in this model will be denoted as $OUTPUT^{\Pi}(x, y) = (OUTPUT_1^{\Pi}(x, y), OUTPUT_2^{\Pi}(x, y))$.

More precisely, given a g-hybrid protocol $\Pi = (A_1, A_2)$ and pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ which is admissible in the real model for the protocol Π, the joint execution of Π under $\widetilde{A}$ on input pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and auxiliary input $z \in \{0,1\}^*$ in the real model is defined as

$$REAL_{\Pi,\widetilde{A}(z)}(x,y) \triangleq \begin{cases} (OUTPUT^{\Pi}(x,y), A_1(VIEW_1^{\Pi}(x,y),z), B_2(VIEW_2^{\Pi}(x,y))) \\ \qquad\qquad \text{if the corrupted player is Player 1} \\ (OUTPUT^{\Pi}(x,y), A_1(VIEW_1^{\Pi}(x,y)), A_2(VIEW_2^{\Pi}(x,y),z)) \\ \qquad\qquad \text{if the corrupted player is Player 2} \end{cases}$$

Then, a protocol is secure if, considered whenever one of the parties is honest-but-curious, we can simulate the joint outcome by the execution in a ideal model with one party is also honest-but-curious.

Definition 3.2.1 (Statistical Security in Honest-But-Curious Model) *Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a desired functionality; let Π be a two-party protocol for computing f with access to the functionality g; and let B be an ideal protocol which has access to the ideal functionality f. A g-hybrid protocol Π securely computes f with an error of at most ϵ if for every pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ that is admissible in the real model for the protocol Π, there exists a pair algorithms $\widetilde{B} = (\widetilde{B_1}, \widetilde{B_2})$ that is admissible in the ideal model for protocol B, such that for all $x \in \mathcal{X}, y \in \mathcal{Y}$, and $z \in \{0,1\}^*$, we have*

$$\mathrm{SD}(\mathrm{P}_{\mathrm{REAL}^{\mathrm{g}}_{\Pi,\widetilde{\mathrm{A}}(\mathrm{z})}(\mathrm{x},\mathrm{y})}, \mathrm{P}_{\mathrm{IDEAL}_{\mathrm{f},\widetilde{\mathrm{B}}(\mathrm{z})}(\mathrm{x},\mathrm{y})}) \leq \epsilon$$

3.2.2 Security in the Malicious Model

In [18] the ideal and real model is different from what is presented here. However, the proof of the theorem 1 of that paper does not suffer any change.

The ideal model A malicious adversary can refuse to participate in the protocol when the protocol is first invoked or he can modify his input on different from what the protocol determines. As we work with oblivious transfer, we define a functionality f in which only one party obtains an output (single-output functionality).

The ideal protocol works as follows. An honest party always sends his own input w to the trusted party and a malicious party may either send some $w' \in \{0,1\}^{|w|}$ or nothing to the trusteds party. In the case of the malicious party, nothing is sent to the trusted party and the trusted party assumes to that his input is a default value. Next, the trusted party computes the desired functionality and only replies the second player with v, where $f(x,y) = (\perp, v)$. An honest party always outputs the message that it receives from the trusted party but a malicious player may change the output received from the trusted party according to

his initial input and the message that he obtained from the trusted party (this message can be $\perp$).

Formally, let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a desired single-output functionality and let $(\perp, v) = f(x,y)$, $v = f_2(x,y)$ be the outcome of the functionality. Given an f-hybrid protocol $B = (B_1, B_2)$ and the pair of algorithms $\widetilde{B} = (\widetilde{B_1}, \widetilde{B_2})$ which is admissible in the ideal model for the protocol B, the joint execution of B under $\widetilde{B}$ on input pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$, where $|\mathcal{X}| = |\mathcal{Y}|$, and auxiliary input $z \in \{0,1\}^*$ in the ideal model is defined as

$$IDEAL_{f,\widetilde{B}(z)}(x,y) \triangleq \begin{cases} (B_1(x,z,\perp), B_2(y, f_2(B_1(x,z), B_2(y)))) & \text{if the corrupted player is Player 1} \\ (B_1(x,\perp), B_2(y,z, f_2(B_1(x), B_2(y,z)))) & \text{if the corrupted player is Player 2} \end{cases}$$

The real model In a two-party protocol in the real model there is no trusted party, and the malicious player may follow an arbitrary strategy.

Formally, let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be a desired single-output functionality and $(\perp, v) = f(x,y)$ be the outcome of functionality. Given a g-hybrid protocol $\Pi = (A_1, A_2)$ and a pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ which is admissible in the ideal model for the protocol B, the joint execution of Π under $\widetilde{A}$ on input pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$ and auxiliary input $z \in \{0,1\}^*$ in the ideal model, denoted $REAL^g_{\Pi,\widetilde{A}(z)}(x,y)$, is defined as the output pair resulting from the interaction between $\widetilde{A_1}(x,z)$ and $\widetilde{A_2}(y,z)$. An honest party ignores the auxiliary input z.

Definition 3.2.2 (Statistical Security in Malicious Model, [18], Definition 3) *A g-hybrid protocol Π securely computes f with an error of at most ϵ if, for every pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ admissible in the real model for the protocol Π, there exists a pair of algorithms $\widetilde{B} = (\widetilde{B_1}, \widetilde{B_2})$ is admissible in the ideal model for protocol B , such that for all $x \in \mathcal{X}, y \in \mathcal{Y}$, and $z \in \{0,1\}^*$, we have*

$$\mathsf{SD}(P_{REAL^g_{\Pi,\widetilde{A}(z)}(x,y)}, P_{IDEAL_{f,\widetilde{B}(z)}(x,y)}) \leq \epsilon$$

3.3 INFORMATION-THEORETIC CONDITIONS FOR SECURITY

In order to simplify the security proofs, the real/ideal paradigm can be replaced by information-theoretic conditions to define secure protocols. As our intention is work in the malicious model, we show here the security in the malicious model. We stress that in an honest-but-curious model, the adversary is not allowed to modify its input. So the definition in that case is similar to the next definition.

3.3.1 Security on Malicious Model

Let X and Y be random variables denoting players inputs distributed according to a distribution P_{XY} unknown to the players, and let U and V be random variables denoting the outputs of the parties. We assume that the ideal functionality is deterministic.

Definition 3.3.1 ([18], Theorem 1) *A protocol Π securely compute the deterministic functionality f with an error of at most 3ϵ if, for every admissible pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ admissible in the real model for protocol Π and for any inputs (X, Y) distributed according to P_{XY} over $\mathcal{X} \times \mathcal{Y}$, $\widetilde{A}$ produces outputs (U, V) distributed according to $P_{UV|XY}$ such that the following conditions are satisfied:*

- *(Correctness) If both parties are honest, we have*

$$\Pr[(U, V) = f(X, Y)] \geq 1 - \epsilon$$

- *(Security for Player 1) If player 1 is honest, then there exist random variables Y' and V' distributed according to $P_{Y'V'|X,Y,U,V}$, such that*

$$\Pr[(U, V') = f(X, V')] \geq 1 - \epsilon$$

$$I_S(X; Y'|Y) \leq \epsilon$$

and

$$I_S(UX; V|YY'V') \leq \epsilon.$$

- *(Security for Player 2) If player 2 is honest, then there exist random variables X' and U' distributed according to $P_{X'U'|X,Y,U,V}$, such that*

$$\Pr[(U', V) = f(X', V)] \geq 1 - \epsilon$$

$$I_S(Y; X'|X) \leq \epsilon$$

and

$$I_S(VY; U|XX'U') \leq \epsilon.$$

Both $P_{y'V'|X,Y,U,V}$ and $P_{X'U'|X,Y,U,V}$ should have explict constructions.

Crépeau and Wullschleger proved that this definition is equivalent to the definition 3.2.2.

3.4 COMPOSITION OF PROTOCOLS

Since a protocol can be implemented using other protocols as subroutines, a definition of secure protocols must guarantee that a protocol composed of secure protocols is still secure. There are different types of composition, such as *sequential composition* and *universal composition*. Sequential composition means run several secure protocols one after the other, where the inputs for each execution are the local outputs and all intermediate results from the previous one. Universal composition means running protocols in parallel at the same time (concurrently).

According to Canneti [36], a composite protocol is secure if it has the follow property: even adversaries that have already gathered some information on other executions of the protocol will not be able to gather additional information about the current execution, or otherwise gain some undesired advantage. In the case of non-adaptative adversaries, this property is guaranteed by letting the adversary have some arbitrary auxiliary input, representing the information gathered by the adversary during the other protocol executions, at the beginning of the current execution. For this reason, in definition 3.2.2 we have an auxiliary input $z \in \{0,1\}^*$.

A result of [36] is that Definition 3.2.2 implies sequential composition and by [18] we have that the Definition 3.3.1 is equivalent to Definition 3.2.2.

4 INTERACTIVE HASHING

Interactive Hashing (IH) was introduced in [37] and has been a very valuable tool in cryptography. It is a cryptographic primitive between two players, the sender (Bob) and the receiver (Alice) that takes as input a string $w \in \{0,1\}^m$ from Bob, and produces as output two m-bit strings: one of them is w and the other is $w' \neq w$. Let the two output strings be w_0 and w_1, according to lexicographic order. Then exists a $d \in \{0,1\}$ such that $w_d = w$. The output strings are available to both Bob and Alice. Interactive Hashing has the following properties: the cheating Alice cannot tell which of (w, w') was the Bob's input, and at least one of w, w' is effectively beyond the control of the cheating Bob.

Interactive Hashing is named after its initial implement which used random hash functions. In that scenario, Bob wants to send w to Alice. He transfers a random two-to-one hash $y = h(w)$ to Alice. Thus, Alice does not learn which of the two pre-images $\{w, w'\} = h^{-1}(y)$ was Bob input. Furthermore, Bob is guaranteed that Alice does not learn any additional information on w. Note that interactive hashing can be implemented without hash functions. For example, in [20] we have an IH protocol where Bob wants to send w with m bits to Alice. In order to, Alice chooses uniformly on $(m-1) \times m$ matrix of rank $m-1$. For each row of the matrix, Alice sends it to Bob and he responds with the inner product between the row and w. After $m-1$ rows have been transmitted, Alice and Bob compute the two values of w consistent with the linear system $Q.w = c$, where Q is the matrix chosen by Alice and c is the vector of Bob's response. However, in this work a interactive hashing protocol will be used that is closer to the former, i.e., it uses hash functions to implement a secure interactive hashing. We will focus on the information-theoretic variant of IH. This line of research was initiated by [22].

4.1 SECURITY OF INTERACTIVE HASHING

In this section we present the definition for a secure interactive hashing protocol. We say that the IH protocol is secure for Bob if, even Alice cheats, she does not know which of the two outputs of the protocol the Bob' s input was. On the other hand, we say that the IH protocol is secure for Alice if, for any strategy of Bob, he cannot force a particular property on both outputs. It means that w and w' cannot simultaneously belong to a specific set S. Formally, we have the following definition:

Definition 4.1.1 (Security of Interactive Hashing [23]) *An interactive hashing protocol is* secure for Bob

if, for every unbounded strategy of Alice (A'), and every W, if W_0, W_1 are the outputs of the protocol between an honest Bob with input W and A', then the distributions $\{View_{A'}^{\langle A',B\rangle}(W)|W = W_0\}$ *and* $\{View_{A'}^{\langle A',B\rangle}(W)|W = W_1\}$ *are identical, where* $View_{A'}^{\langle A',B\rangle}(W)$ *is Alice's view of the protocol when Bob's input is W. An interactive hashing protocol is (s, ρ)-*secure for Alice *if, for every $S \subseteq \{0,1\}^m$ of size at most 2^s and every unbounded strategy of Bob (B'), if W_0, W_1 are the outputs of the protocol, then*

$$\Pr[W_0, W_1 \in S] < \rho.$$

*Here the probability is taken over the coin tosses of Alice and Bob. An interactive hashing protocol is (s, ρ)-*secure *if it is secure for Bob and (s, ρ)-*secure for Alice.

In the next section we present the constant round interactive hashing protocol of [23, Section 5.4].

4.2 THE CONSTANT ROUND INTERACTIVE HASHING PROTOCOL

One of the principal tools used in the constant round interactive hashing protocol is an η-almost t-wise independent permutation. A t-wise independent permutation π is such that when it is applied in any t points in $\{0,1\}^m$, the permutation π behaves as a truly random permutation. [1] In an η-almost t-wise independent permutations π' [2], the distribution on any t points has statistical distance at most η to the distribution induced on these points by a truly random permutation. A 2-wise independent permutation can be seen as a 0-almost 2-wise independent permutation and we can build it, by choosing $a, b \in_R GF(2^m)$, $a \neq 0$ (the strings $\{0,1\}^m$ are identified with the field $GF(2^m)$) and defining the permutation by $g(x) = ax + b$.

Another tool used in this protocol is the 2-1 hash function. Let h: $\{0,1\}^m \rightarrow \{0,1\}^{m-1}$ be a 2-1 hash function. Then, for each output of h there are exactly 2 pre-images. Notice that to construct a 2-wise independent 2-1 hash function, one can take a 2-wise independent permutation and omit the last bit of its output. Thus, we have a way of computing 2-wise independent 2-1 hash function by permutation polynomials over finite fields.

In order to implement the constant round interactive hashing, let the set S have size $|S| = 2^s$. Remember that S is the subset of $\{0,1\}^m$ whose strings have some particular property and the sender's input is $w \in \{0,1\}^m$. Then, the parameters and tools of the protocol are

[1] The reason we do not use a truly random permutation is because your description would be exponential on m.
[2] See [38, 39] for a survey.

– The parameters are m and s, such that m is the length of string w and s is the parameter which defines the size of set S,

– $t = m$ and $\eta = (\frac{1}{2^v})^t$, where $v = s - \log m$,

– A family Π of η-almost t-wise independent permutations π: $\{0,1\}^m \rightarrow \{0,1\}^m$,

– A family G of 2-wise independent 2-1 hash functions g: $\{0,1\}^{m-v} \rightarrow \{0,1\}^{m-v-1}$,

– A family H (induced by Π and G) of 2-1 hash functions h: $\{0,1\}^m \rightarrow \{0,1\}^{m-1}$ defined as:

$$h(x) \triangleq \pi(x)_1, ..., \pi(x)_v, g(\pi(x)_{v+1}, ..., \pi(x)_m),$$

where $\pi(x)_i$ denotes the i^{th} bit of $\pi(x)$.

Now we will present the protocol that implements interactive hashing with only four message exchanges, regardless of the size of the string that the sender wants to send to the receiver. Let Bob be the sender and Alice be the receiver. Bob, in this case, has the input $w \in \{0,1\}^m$.

Protocol 4.2.1

1. Alice chooses $\pi \in_R \Pi$ and sends the description of π to Bob.
2. Bob computes $\pi(w) = z_1...z_m$, where z_i is the i^{th} bit of $\pi(w)$, and sends the bits $z_1...z_v$ to Alice.
3. Alice chooses $g \in_R G$ and sends the description of g to Bob.
4. Bob computes and sends $g(z_{v+1}...z_m)$ to Alice.
5. Both compute and output (w_0, w_1) such that $h(w) = h(w_0) = h(w_1)$.

In [23, Section 5.4] has the following theorem which determines the bounds of security parameters for to achieve a (s, ρ)-*secure* interactive hashing.

Theorem 4.2.2 *For all s, m such that $s \geq \log m + 2$, the above protocol is a $(s, 2^{-(m-s)+O(\log m)})$-secure η'-uniform interactive hashing protocol for $\eta' = (2^{s-\log m-1})^{-m} < 2^{-m}$.*

Remark. Note that [39, 23] has a result that confirms that there exists an η-almost t-wise independent permutation space Π which runs in polynomial time and space with the security parameters of the protocol above. Furthermore, this is an η'-uniform interactive hashing protocol, where $\eta' = \eta 2^m$, since h is η-close to be pairwise independent.

5 OBLIVIOUS TRANSFER

Oblivious Transfer was introduced by Rabin in [2]. In Rabin OT the sender transmits a bit b to the receiver who was output either the bit b with probability $\frac{1}{2}$ or an erasure with probability $\frac{1}{2}$. The sender does not know which output receiver received. In addition, if b is chosen independently and uniformly at random when the receiver receives an erasure, he cannot guess what bit was sending with probability greater than $1/2$. Thus, the receiver remains ignorant about the sender's input. Note that Rabin OT can be seen as an erasure channel [Chapter 2, Definition 2.2.2]. Another flavor of OT introduced by [40] is the one-out-of-two oblivious transfer. In this protocol, the sender inputs two bits b_0, b_1 and the receiver inputs a bit c called the *choice bit*. The receiver receives b_c and remains ignorant about $b_{\bar{c}}$, while the sender remains ignorant about receiver's choice. In this protocol, if the sender's input are two strings $b_0, b_1 \in \{0,1\}^k$, one has one-out-of-two string oblivious transfer (String OT). We call the reader's attention to the fact that has been showed by [3] that both kinds of OT are equivalent.

In this work, we will use one-out-of-two string oblivious transfer. So, unless explicitly indicate, otherwise when we say Oblivious Transfer, we refer to one-out-of-two string oblivious transfer. We assume that the inputs are uniformly random to make our analysis easier. It can be done without loss of generality due to the (very efficient) randomized self-reduction of OT [41, Section 3.2].

We will present a definition of security OT in the next section, then we present the definition of OT capacity and, finally, we present our main theorem: the capacity of oblivious transfer on a generalized erasure channel in a malicious model, in other words, it give us an optimal way in which a GEC can be used for obtaining secure implementations of OT with fully malicious players.

5.1 SECURITY OF OT

We will use the security definition of String OT from Crépeau and Wullschleger [18]. We choose this definition because it is an information-theoretic security definition with a particular property: this definition implies that the String OT protocol that satisfies it is sequentially composable.

We remember the reader that a $\mathcal{F}$-hybrid protocol consists of a pair of algorithms $P = (A_1, A_2)$ that can interact by means of two-way message exchange and have access to some functionality $\mathcal{F}$. A pair of

algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$ is admissible for protocol P if at least one of the parties is honest, that is, if $\widetilde{A_1} = A_1$ or $\widetilde{A_2} = A_2$ or both.

Let B and C be the random variables denoting the sender's input (B_0, B_1) and the receiver's input, respectively, distributed according to a distribution P_{BC} unknown to the players, and left U and V be the random variables denoting the outputs of sender and receiver, respectively.

It was shown in [18] that Definition 3.3.1 applied to the case of the Oblivious Transfer functionality is the follow definition:

Definition 5.1.1 *A protocol P securely realizes String OT (for strings of length k) with an error of at most 6ϵ if, for every admissible pair of algorithms $\widetilde{A} = (\widetilde{A_1}, \widetilde{A_2})$, for protocol P, and for all inputs (B, C), $\widetilde{A}$ produce outputs (U, V) such that the following conditions are satisfied:*

- *(Correctness) If both parties are honest, then $U = \perp$ and*

$$\Pr[V = B_C] \geq 1 - \epsilon$$

- *(Security for Sender) If the sender is honest, then we have $U = \perp$ and there exists a random variable C' distributed according to $P_{C'|B,C,V}$, such that*

$$I_S(B; C'|C) \leq \epsilon$$

  and

$$I_S(B; V|C, C', B_{C'}) \leq \epsilon.$$

- *(Security for Receiver) If the receiver is honest, we have $V \in \{0,1\}^k$ and*

$$I_S(C; U|B) \leq \epsilon.$$

The protocol is secure if ϵ is negligible in the security parameter n.

5.2 OT CAPACITY

The *oblivious transfer capacity* [14] measures the optimal way of implementing secure OT from a noisy resource, i.e., it measures how efficiently an OT protocol uses the noisy resource. [1]

We present the definition of the oblivious transfer capacity.

Definition 5.2.1 *If the noisy channel is used n times and k is the length of the string being obliviously transferred, the* oblivious transfer rate *of the protocol is given by*

$$R_{OT} = \frac{k}{n}.$$

Definition 5.2.2 *If there exists a protocol implementing oblivious transfer securely and with a OT rate R, we call R as an* achievable rate*.*

Definition 5.2.3 *The* oblivious transfer capacity *of a protocol is given by*

$$C_{OT} = \sup \frac{k}{n}.$$

where the supremum is taken over all achievable rates.

Note that the oblivious transfer capacity, by definition, depends on the type of the channel. In this work, we consider the oblivious transfer capacity of the Generalized Erasure Channel when the adversaries are malicious. We calculate the optimal capacity and a protocol that achieves it. Our main result is given in the next theorem. We would like to remember that [16] has found OT capacity of the GEC when the adversaries are passive. Actually, our result asserts that the OT capacity of the GEC in the passive and in the malicious adversary model, is the same. The reader can obtain bounds for OT capacity in other noisy resources in [14, 16, 15].

Theorem 5.2.4 *For a Generalized Erasure Channel with $p^* \geq \frac{1}{2}$, the oblivious transfer capacity in the case of malicious adversaries is $(1 - p^*)C(W_0)$ where $C(W_0)$ is the Shannon capacity of the discrete memoryless channel $\{W_0 : \mathcal{X} \to \mathcal{Y}_0\}$.*

Remark The Shannon capacity of the discrete memoryless channel is defined as $C = \max_{p(x)} I(x; y)$, where the maximum is taken over all possible input distributions $p(x)$ and $I(.;.)$ is the Shannon mutual

[1] Here, we will work with Generalized Erasure Channel, but one can implement OT from noisy distribution. See [14] to know what kind of noisy distribution one can implement OT.

information. Remember that the channel capacity is the logarithm of the maximum number of distinguishable signals for n uses of a communication channel and operationally the channel capacity is the highest rate in bits per channel use at which information can be sent with arbitrarily low probability of error.

The protocol that achieves this capacity is presented in next chapter and the proof of Theorem 5.2.4 is in chapter 7.

6 THE PROTOCOL

In this chapter we show first the encoding scheme of subsets used in the protocol that achieves OT capacity based on GEC. Subsequently we present the own protocol and, finally, we prove that it is a secure implementation of OT against malicious adversaries.

6.1 ENCODING SCHEME OF SUBSETS

An encoding scheme of subsets the following problem: given a string of length n, we choose l positions it and we wish represent the set k of l positions as a binary string.

Cover showed [42] that there exist an efficiently computable, one to one mapping $F : \binom{[n]}{l} \rightarrow [\binom{n}{l}]$ for pair of every integers $l \leq n$. The encoding, as described, associates an integer in the set $\{0, ..., \binom{n}{l} - 1\}$ with the subset k of $[n]$, $|k| = l$. The encoding of the subset k of $[n]$ is given by:

$$\sigma(k) = \sum_{i=1}^{l} \sum_{j=e_{i-1}+1}^{e_1-1} \binom{n-i}{l-i}$$

where $k = \{e_1, e_2, \ldots, e_v, \ldots, e_l\}$. Hence, we can encode the set $\binom{[n]}{l}$ in bit strings of length $m = \lceil log\binom{n}{l} \rceil$(see [22, Section 3.1] for more details). The decoding of a binary string m is done by calculating each e_v like $\max_r\{\sum_{e_1+1}^{r-1} \binom{n-j}{l-i}\} \leq m$, taking $m = m - \sum_{e_1+1}^{j=e_i-1} \binom{n-j}{l-i}$ and repeating from $i = 1$ to $i = l$. Nevertheless, a disadvantage of this scheme is that the strings that correspond to valid encodings can constitute only slightly more than half of all strings.

We use here the solution of Savvides [20, Section 4.2.1] in which each string $w \in \{0,1\}^m$ encodes the same subset as $w \pmod{log\binom{n}{l}}$, which is always a valid encoding in the original scheme. Since each subset correspond to either 1 or 2 strings in $\{0,1\}^m$, this scheme can at most double the fraction of the strings that maps to subsets with a desired property.

6.2 OUR PROTOCOL

The next protocol is based on the protocol for String OT from the erasure channel [20, Protocol 5.1]. We assume a malicious (or active) adversary that can have an arbitrary behavior. The players are connected by a noiseless channel and by a Generalized Erasure Channel (Chapter 2, Definition 2.2.3).

Let Alice be the sender and Bob be the receiver. Alice wishes to send the random strings r_0 and r_1 to Bob and he will learn information about at least one of them.

Alice's inputs are a binary string x^n, chosen randomly, and four 2-universal hash functions g_0, g_1, h_0 and h_1 chosen randomly too. Bob's inputs are the choice bit c and $w \in_R \{0,1\}^m$, where w will decoded into a set S whith $|S| = \alpha n$. In the beginning of the protocol both players agree on the security parameters α, ϵ and γ, where αn is the proportion of bits sacrificed to test. Both know the probability p^*.

The protocol works in the following way: Alice sends x^n through the GEC, whose output is y^n. Bob creates two disjoint sets, a set $\mathcal{B}$ containing the positions in which the bits were transmitted through the erasure channel, and a set $\mathcal{G}$ containing the positions where the bits were transmitted through the discrete memoryless channel. Bob aborts if he does not have the sufficient amount of positions to continue the protocol. Bob creates also two other sets, $\mathcal{R}_c$ with only positions from $\mathcal{G}$ and $\mathcal{R}_{\overline{c}}$ with the αn positions to test from $\mathcal{G}$ and the other $(\beta - \alpha)n$ positions, out of the positions have already chosen, from $\mathcal{G} \cup \mathcal{B}$. Note that both positions αn and from $\mathcal{G}$ are transmitted through discrete memoryless channel, then Bob knows all bits in this positions at least some bits that can be flipped by the channel according to the matrix channel W_0. Then Bob sends that description of $\mathcal{R}_0$ and $\mathcal{R}_1$ to Alice so that she can confirm that all positions have chosen once and Bob is following the protocol at this point. Of course, Alice does not know which set (either $\mathcal{R}_0$ or $\mathcal{R}_1$) is formed only by positions from $\mathcal{G}$ and which of them has positions to test.

Then, Bob sends to Alice which positions are to test from w through interactive hashing. For interactive hashing properties Alice will receive the positions chosen by Bob and will receive other set of αn positions to test that is not under control of Bob. We will show later that this property of IH will help us to guarantee to Alice that Bob is not deviating from the protocol. Note also that Bob knows the bit b s.t. $w = w_b$ and knows $w_{\overline{b}}$.

Until this point, Bob just sent positions to Alice. Now, Bob wishes to send to Alice the bits received in the positions to test, $y^{\mathcal{R}_1^{\mathcal{S}_a}} = y^{\mathcal{R}_{\overline{c}}^{\mathcal{S}_b}}$, and the bits received in positions determined by the other output of interactive hashing, $y^{\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}} = y^{\mathcal{R}_c^{\mathcal{S}_{\overline{b}}}}$.

In order to known if the possible errors introduced by discrete memoryless channel in the string y^n are so many as to invalidate the string received by Bob, Alice tests if $y^{\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}}$ and $y^{\mathcal{R}_1^{\mathcal{S}_a}}$ are jointly typical with her input on these positions. If they are not typical, y^n contains too many errors and the protocol is aborted.

Now, Alice wants to discard the αn bits used in the test and extract the random strings r_0 and r_1. However, Alice has to correct the possible errors in the string received by Bob y^n. To this end, Alice computes $g_0(x^{\mathcal{R}_0})$, $g_1(x^{\mathcal{R}_1})$ and sends the descriptions of g_0, g_1 to Bob. Thus, Bob can computes all possible $\widetilde{x}^{\mathcal{R}_c}$ that are jointly typical with $y^{\mathcal{R}_c}$ and satisfy $g_c(\widetilde{x}^{\mathcal{R}_c}) = g_c(x^{\mathcal{R}_c})$.[1] By doing this, Bob is able to "recover" the string sent in the beginning of protocol by Alice $\widetilde{x}^{\mathcal{R}_c}$.

At this point, Alice can extract the random strings r_0 and r_1 (discarding the αn bits used in the positions test; the $\beta n[H(X|Y \in \mathcal{Y}_0) + \epsilon]$ bits used to jointly typical sequences test; and other $\beta n + H(X)[\beta n - 5\alpha n] + \gamma n$ bits such that the security of protocol is achieved) doing $r_0 = h_0(x^{\mathcal{R}_0})$ and $r_1 = h_1(x^{\mathcal{R}_1})$ and sending the descriptions of h_0, h_1 to Bob. In turn, Bob can finally receives the desired random string $r_c = h_c(\widetilde{x}^{\mathcal{R}_c})$.

So, we can assert the following theorem.

Theorem 6.2.1 *The protocol 6.2.2 is secure under the definition 5.1.1.*

[1] Since Bob uses the output of universal hash functions to correct errors, the above protocol is not computationally efficient. However, this suffices for our result as we only claim possibility of achieving the OT capacity.

Protocol 6.2.2

1. Alice and Bob select a (typically very small) positive constant $\alpha < \frac{1-p^*}{3}$ and set $\beta = 1 - p^* - 2\alpha$.

2. Alice randomly chooses x^n according to the probability distribution that achieves the Shannon capacity of W_0 and sends x^n through the GEC.

3. Bob receives the string y^n and collects the *good* (those corresponding to $y \in \mathcal{Y}_0$) and the *bad* (those corresponding to $y \in \mathcal{Y}^*$) positions in sets $\mathcal{G}$ and $\mathcal{B}$, respectively. He aborts if $|\mathcal{G}| < (1 - p^* - \alpha)n = \beta n + \alpha n$.

4. Bob chooses $c \in_R \{0,1\}$ and $w \in_R \{0,1\}^m$, where $m = \lceil \log \binom{\beta n}{\alpha n} \rceil$. He decodes w into a subset $\mathcal{S}$ of cardinality αn (out of βn) using the encoding scheme of Section 6.1. Bob then defines two disjoint sets $\mathcal{R}_c$ and $\mathcal{R}_{\overline{c}}$ of cardinality βn. $\mathcal{R}_c$ consists only of positions from $\mathcal{G}$, chosen randomly and without repetition. $\mathcal{R}_{\overline{c}}$ has αn positions from $\mathcal{G}$ (defining the subset $\mathcal{R}_{\overline{c}}^{\mathcal{S}}$) and the remaining positions are chosen from $\mathcal{G} \cup \mathcal{B}$, randomly and without repetition. Bob sends the descriptions of $\mathcal{R}_0$ and $\mathcal{R}_1$ to Alice.

5. Alice checks that no position is repeated in the sets $\mathcal{R}_0$ and $\mathcal{R}_1$, otherwise she aborts.

6. Bob sends w to Alice using Interactive Hashing (Protocol 4.2.1). Let w_0, w_1 be the output strings, let $\mathcal{S}_0$, $\mathcal{S}_1$ be the corresponding subsets of cardinality αn and let $b \in \{0,1\}$ be such that $w_b = w$.

7. Bob announces $a = b \oplus c$ as well as $y^{\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}}$ and $y^{\mathcal{R}_1^{\mathcal{S}_a}}$.

8. Alice checks if $y^{\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}}$ and $y^{\mathcal{R}_1^{\mathcal{S}_a}}$ are jointly typical (for a discrete memoryless channel $\{W_0 : \mathcal{X} \to \mathcal{Y}_0\}$) with her input on these positions. If they are not jointly typical, Alice aborts.

9. Alice chooses randomly 2-universal hash functions $g_0, g_1 : \mathcal{X}^{\beta n} \to \{0,1\}^{\beta n[H(X|Y \in \mathcal{Y}_0) + \epsilon']}$ (with $\epsilon' > 0$ such that the output length is integer). She computes $g_0(x^{\mathcal{R}_0})$ and $g_1(x^{\mathcal{R}_1})$. She also randomly chooses 2-universal hash functions $h_0, h_1 : \mathcal{X}^{\beta n} \to \{0,1\}^{\delta n}$, where $\delta = (\beta - 5\alpha)H(X) - \beta(H(X|Y \in \mathcal{Y}_0) + \epsilon') - \gamma$ and $\gamma > 0$ such that the output length is integer. She sends $g_0(x^{\mathcal{R}_0})$, $g_1(x^{\mathcal{R}_1})$ and the descriptions of g_0, g_1, h_0, h_1 to Bob. Alice outputs $r_0 = h_0(x^{\mathcal{R}_0})$ and $r_1 = h_1(x^{\mathcal{R}_1})$.

10. Bob computes all possible $\widetilde{x}^{\mathcal{R}_c}$ that are jointly typical with $y^{\mathcal{R}_c}$ and satisfy $g_c(\widetilde{x}^{\mathcal{R}_c}) = g_c(x^{\mathcal{R}_c})$. If there exists exactly one such $\widetilde{x}^{\mathcal{R}_c}$, Bob outputs $r_c = h_c(\widetilde{x}^{\mathcal{R}_c})$. Otherwise, he outputs $r_c = 0^{\delta n}$.

6.3 SECURITY PROOF

In this section we prove Theorem 6.2.1. According to Definition 5.1.1 we have to prove the correctness, the security for Alice (sender) and the security for Bob (receiver).

6.3.1 Correctness

When Alice and Bob are honest, Bob does not obtain the correct output if he aborts in Step 3, or if he does not obtain exactly $\widetilde{x}^{\mathcal{R}_c} = x^{\mathcal{R}_c}$ in Step 10.

We first analyze Step 3. Bob aborts if $|\mathcal{G}| < (1 - p^* - \alpha)n$. Note that $y^n = \{y_1, y_2, \ldots, y_n\}$, $y_i \in \{0, 1\}$ and $Pr[y \in \mathcal{Y}_0] = 1 - p^*$. Then, the expected value of $y_i \in \mathcal{Y}_0$ is $E(y_i | y_i \in \mathcal{Y}_0) = (1 - p^*)p$, where $p = Pr[y_i = 1]$. By the Chernoff bound (Lemma I.0.5 in the appendix) and setting $\eta = \left\{1 - \frac{1}{p} + \frac{\alpha}{(1-p^*)p}\right\}$, the probability that Bob aborts in Step 3 is

$$Pr\left\{\frac{1}{n}\sum_{i=1}^{n} y_i \leq 1 - p^* - \alpha\right\} \leq e^{-(1-p^*)n}.$$

So, the event $|\mathcal{G}| < (1-p^*-\alpha)n$ occurs with probability $\epsilon < e^{-(1-p^*)n}$, which is a negligible function of n.

Analyzing Step 3, Bob does not obtain exactly $\widetilde{x}^{\mathcal{R}_c} = x^{\mathcal{R}_c}$ either if $x^{\mathcal{R}_c}$ is not jointly typical with $y^{\mathcal{R}_c}$ or if there exists another $\overline{x}^{\mathcal{R}_c}$ that is both jointly typical with $y^{\mathcal{R}_c}$ and has $g_c(\overline{x}^{\mathcal{R}_c}) = g_c(x^{\mathcal{R}_c})$.

However, by Theorem 2.2.9, item 1, $Pr((x^{\mathcal{R}_c}, y^{\mathcal{R}_c}) \in A_\epsilon^n) \leq \varepsilon$ as $n \to \infty$. Then, the probability of $x^{\mathcal{R}_c}$ is not jointly typical with $y^{\mathcal{R}_c}$ is negligible with n. By Theorem 2.2.9, item 2, the number of $\overline{x}^{\mathcal{R}_c}$ that are jointly typical with $y^{\mathcal{R}_c}$ is

$$|A_\epsilon^n| \leq 2^{\beta n[H(X|Y \in \mathcal{Y}_0) + \epsilon']}$$

for $0 < \epsilon' < \epsilon$ and n sufficiently large. So, by the Leftover-Hash Lemma, we have that the probability of a $\overline{x}^{\mathcal{R}_c}$ jointly typical with $y^{\mathcal{R}_c}$ and with $g_c(\overline{x}^{\mathcal{R}_c}) = g_c(x^{\mathcal{R}_c})$ is

$$\frac{2^{\beta n[H(x|y \in \mathcal{Y}_0) + \epsilon']}}{2^{\beta n[H(x|y \in \mathcal{Y}_0) + \epsilon]}} = 2^{\beta n(\epsilon' - \epsilon)} \to 0 \quad \text{as} \quad n \to \infty.$$

As all the failure probabilities are negligible in n, the protocol meets the correctness requirement

$$Pr[V = B_C] \geq 1 - \epsilon.$$

6.3.2 Security for Bob

In order to provide security for Bob we have to prove that at the end of the protocol the probability of Alice has any information beyond her point of view (all the information in possession of player at the end of the protocol including the results of all local computations, local random samplings, local inputs and messages exchanged) about the choice bit c of Bob since her input B is negligible.

Alice just can obtain information about c when Bob sends to her the descriptions of $\mathcal{R}_0$ and $\mathcal{R}_1$ in Step 4 and in Step 7 when Bob announces $a = b \oplus c$ as well as $y^{\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}}$ and $y^{\mathcal{R}_1^{\mathcal{S}_a}}$.

We first analyze Step 4. Since in GEC, every input symbol x is erased (i.e. ends up in $\mathcal{Y}^*$) with probability p^* independent of x, Alice does not know which input symbols were erased. Hence, the distribution of $(\mathcal{R}_0, \mathcal{R}_1)$ is independent of c from Alice's point of view. So Alice cannot tell if $\mathcal{R}_c = \mathcal{R}_0$ or $\mathcal{R}_c = \mathcal{R}_1$.

In Step 7, Alice receiving a can correctly guess c if and only if she can correctly guess b, but the security of Interactive Hashing protocol provides that Alice's view is the same for $b = 0$ and $b = 1$.

Note that no matter what the malicious Alice actually sends in Step 9, Bob will not abort. In particular, this prevents reaction attacks [2], i.e. receiving the bits of Bob from the previous step, a malicious Alice will gain nothing if she tries to cheat by sending some different function, depending on the bits received, that g_0, g_1, h_0, h_1 set the protocol in Step 9.

Therefore, the distribution of Alice's view of the protocol does not depend on c. Thus, given a small positive constant $\varepsilon = \varepsilon(p^*, \eta')$, where η' comes from the Interactive Hashing (see Chapter 4), we have

$$I_S(C; U|B) \leq \varepsilon.$$

6.3.3 Security for Alice

Our proof follows the lines of Savvides' proof [20, Section 5.1]. In order to provide the security for Alice, we had to prove that a malicious Bob cannot complete the protocol without being caught with

[2] Reaction attack is an attack where the adversary observes the behavior of the player and tries to extract some information from there.

overwhelming probability and if he deviated from the protocol a little, he will have information about both Alice's inputs with a negligible probability.

Note that Bob has about $(1-p^*)n$ good positions except with a probability negligible in n (see 6.3.1). Therefore a malicious Bob can follow two strategies: either (i) Bob uses good positions in both R_0 and R_1 as much as possible, and thus both have many bad positions, or (ii) Bob uses only a few good positions in one of R_0 and R_1 in order to complete the protocol and try to get information from both Alice's input. We will show bellow both strategies does not give information to Bob of two Alice's input. We first present some definitions.

Definition 6.3.1 *Let $u(\mathcal{R})$ be the number of positions contained in $\mathcal{R}$ such that the corresponding output at this position was an erasure.*

Definition 6.3.2 *$\mathcal{S}$ is called* good *for $\mathcal{R}$ if $u(\mathcal{R}^{\mathcal{S}}) < \alpha^2 n$, otherwise it is called* bad *for $\mathcal{R}$.*

Then we can reduce the two strategies of malicious Bob to: (i) both $u(\mathcal{R}_0)$ and $u(\mathcal{R}_1)$ are greater than or equal to $2\alpha n$, (ii) either $u(\mathcal{R}_0)$ or $u(\mathcal{R}_1)$ is less than $2\alpha n$.

For the first case we need the following two lemmas from [20, Section 5.1]. Lemma 6.3.3 follows from the Chernoff bound and the properties of GEC, and Lemma 6.3.4 follows from the last lemma, the union bound and the properties of encoding scheme used.

Lemma 6.3.3 *Let $\mathcal{R}$ be a set of cardinality βn such that $u(\mathcal{R}) \geq 2\alpha n$. Then the fraction f of subsets $\mathcal{S}$ of cardinality αn that are good for $\mathcal{R}$ satisfies $f < e^{-\alpha^2 n/4}$.*

Lemma 6.3.4 *Let $\mathcal{R}_0, \mathcal{R}_1$ be sets of cardinality βn such that $u(\mathcal{R}_0) \geq 2\alpha n$ and $u(\mathcal{R}_1) \geq 2\alpha n$. Then the fraction of strings w that decode to subsets $\mathcal{S}$ that are good for either $\mathcal{R}_0$ or $\mathcal{R}_1$ is no larger than $4e^{-\alpha^2 n/4}$.*

Strategy 1 Since the fraction of the strings $w \in_R \{0,1\}^m$ that are good for either $\mathcal{R}_0$ or $\mathcal{R}_1$ is no larger than $4e^{-\alpha^2 n/4}$, we can set the security parameter s of the interactive hashing protocol to $\log(4e^{-\alpha^2 n/4}2^m) = m - \frac{\alpha^2 n}{4}\log e + 2$.

We also have that $m = \lceil \log \binom{\beta n}{\alpha n} \rceil \leq \lceil \log 2^{\beta n} \rceil = \lceil \beta n \rceil = O(n)$. Therefore we have that $\rho = 2^{-(m-s)+O(\log m)} = 2^{-\alpha^2 \log(e) n/4 + O(\log n)}$ and so, by the security of the interactive hashing protocol, the probability that Bob gets both w_0 and w_1 to be good for either $\mathcal{R}_0$ or $\mathcal{R}_1$ is

$$Pr[w_0, w_1 \in S] < \rho = 2^{-\alpha^2 \log(e) n/4 + O(\log n)}$$

a negligible function of n.

Then, with overwhelming probability one of the sets, without loss of generality $\mathcal{R}_0$, will have $u(\mathcal{R}_0^{\mathcal{S}_{\overline{a}}}) \geq \alpha^2 n$, i.e., $\mathcal{R}_c^{\mathcal{S}_{\overline{b}}}$ will have more than $\alpha^2 n$ positions corresponding with an erasure, and Bob succeeds in the test of Step 8 only if he correctly guesses y's values for these positions that are jointly typical with Alice's input.

For n sufficiently large, by Theorem 2.2.10, there are at most $2^{\alpha^2 n[H(Y\in\mathcal{Y}_0|X)+\epsilon]}$ ($\epsilon > 0$) sequences of y's values that are jointly typical with Alice's input, and by the definition of jointly typical sequences, the sequences of y's values had to be a typical sequence. However, by Theorem 2.2.7, item 2, there are at least $2^{\alpha^2 n[H(Y\in\mathcal{Y}_0)-\epsilon]}$ typical sequences for the values of y.

Then the probability that Bob succeeds in the test is less than $2^{\alpha^2 n[H(Y\in\mathcal{Y}_0|X)-H(Y\in\mathcal{Y}_0)+2\epsilon]} = 2^{-\alpha^2 n[C(W_0)-2\epsilon]}$, which is a negligible function of n.[3]

In other words, a malicious Bob should correctly guess all the information that an honest Bob would have received if these positions were good.

So the probability that Bob successfully cheat in the protocol ζ when both $u(\mathcal{R}_0)$ and $u(\mathcal{R}_1)$ are at least $2\alpha n$ is a negligible function of n:

$$\zeta = 2^{-\alpha^2 n[H(Y\in\mathcal{Y}_0)+H(Y\in\mathcal{Y}_0|X)+2\epsilon+\frac{log(\epsilon)}{4}]+O(n)}$$

Strategy 2 We now analyze the security if either $u(\mathcal{R}_0)$ or $u(\mathcal{R}_1)$ is less than $2\alpha n$ without loss of generality we assume that $u(\mathcal{R}_0) < 2\alpha n$.

By the Chernoff bound (lemma I.0.5 in the appendix) and setting $\eta = \left\{\frac{(p^*-\alpha)}{(1-p^*)p} - 1\right\}$, the probability that $|\mathcal{B}| > (p^* - \alpha)n$ is

$$Pr\left\{\frac{1}{n}\sum_{i=1}^{n} y_i \leq p^* - \alpha\right\} \leq \exp\left(-\frac{n}{2\ln 2}\frac{p^*}{(1-p^*)}\frac{(p^*-\alpha)}{p}\right) = \vartheta.$$

Since only $(1-2\beta)n$ positions were not used in $\mathcal{R}_0, \mathcal{R}_1$, then $u(\mathcal{R}_0)+u(\mathcal{R}_1)+(1-2\beta)n > (p^*-\alpha)n$ and so $u(\mathcal{R}_1) > (1-p^*-7\alpha)n = \beta n - 5\alpha n$.

Observe that more than $\beta n - 5\alpha n$ positions from $\mathcal{R}_1$ are erasures and Alice only sends $\beta n[H(X|Y \in \mathcal{Y}_0) + \epsilon]$ bits of information about $x^{\mathcal{R}_1}$ in Step 9. So we have that $H_\infty(X^{\mathcal{R}_1}|\text{View}_{\text{Bob}}) > n[(\beta -$

[3]Note than $H(Y \in \mathcal{Y}_0|X) - H(Y \in \mathcal{Y}_0) \leq 0$.

$5\alpha)H(X) - \beta H(X|Y \in \mathcal{Y}_0) - \beta\epsilon]$, where $\mathrm{View}_{\mathrm{Bob}}$ denotes all the information that Bob knows. So the property of the 2-universal hash function h_1 for extracting $n[(\beta-5\alpha)H(X)-\beta H(X|Y \in \mathcal{Y}_0)-\beta\epsilon-\gamma]$ bits of information (with $\gamma > 0$) follows from the Leftover-Hash Lemma. Therefore, Bob has only negligible information about r_1, it means, the probability that Bob obtains some information about r_1 is

$$\xi < \frac{1}{2}2^{-\frac{\gamma n}{2}}$$

Hence, we have that

$$I_S(B; C'|C) \leq \zeta$$

and

$$I_S(B; V|C, C', B_{C'}) \leq \xi.$$

Therefore the protocol above is secure according to the definition 5.1.1.

7 ACHIEVING THE OBLIVIOUS TRANSFER CAPACITY

In this chapter we will prove theorem 5.2.4. For sake of reability we repeat it below:

Theorem 5.2.4 For a Generalized Erasure Channel with $p^* \geq \frac{1}{2}$, the oblivious transfer capacity in the case of malicious adversaries is $(1 - p^*)C(W_0)$, where $C(W_0)$ is the Shannon capacity of the discrete memoryless channel $\{W_0 : \mathcal{X} \to \mathcal{Y}_0\}$.

First, we will show that the protocol 6.2.2 achieves the OT rate $(1 - p^*)C(W_0)$ (Direct Part). Subsequently, we will show that this rate is the largest achievable OT rate based on GEC (Converse Part). Hence, the OT capacity being the supremum of all achieves OT rates and the protocol being secure against malicious adversaries, follows the result of theorem.

7.1 DIRECT PART

Theorem 7.1.1 *There exists arbitrarily small positive constants ϵ, α and γ such that using a Generalized Erasure Channel with probability $p^* \geq \frac{1}{2}$, for n sufficiently large, protocol 6.2.2 achieves the OT rate $(1 - p^*)C(W_0)$, where $C(W_0)$ is the Shannon capacity for the discrete memoryless channel $\{W_0 = \mathcal{X} \to \mathcal{Y}_0\}$.*

Proof According to protocol 6.2.2, α indicates the proportion of bits sacrificed in the Step 8 and we have $\alpha < \frac{1-p^*}{3}$, so it can be chosen arbitrarily small. Similarly the parameters ϵ and γ can be chosen arbitrarily small since ϵ is negligible in n and γ is a parameter that help to extract random strings in such way that a malicious Bob cannot learn about more than one of the Alice's input. Note that γ can be chosen arbitrarily small without compromise the security of protocol.

Since the protocol in Step 2 allows Alice chooses x^n according to the probability distribution that achieves the Shannon capacity of W_0, we have that the length of the strings r_0, r_1 obliviously transmitted

when $n \to \infty$ is

$$\begin{aligned} l &= n[(\beta - 5\alpha)H(X) - \beta(H(X|Y \in \mathcal{Y}_0) + \epsilon') - \gamma] \\ &= n\beta[H(X) - H(Y \in \mathcal{Y}_0)] \\ &= n(1-p^*)[H(X) - H(X|Y \in \mathcal{Y}_0)] \\ &= n(1-p^*)I(X; Y \in \mathcal{Y}_0) \\ &= n(1-p^*)C(W_0) \end{aligned}$$

As the OT ratio is defined as the ratio of the length of the strings obliviously transmitted and the number of use of noisy channel we have:

$$R_{OT} = (1-p^*)C(W_0).$$

□

7.2 CONVERSE PART

We will use to this part the theorem 1, its proof, and the lemma 1 of Ahlswede and Csiszár [16]. We will consider Alice as the sender and Bob as the receiver in the OT protocol.

Theorem 7.2.1 *The OT capacity of a Generalized Erasure Channel with probability $p^* \geq \frac{1}{2}$ is $C_{OT} = (1-p^*)C(W_0)$.*

The follow lemma will be used in the proof of above theorem.

Lemma 7.2.2 ([16]) *For arbitrary RVs U, V, Z with values in finite sets $\mathcal{U}, \mathcal{V}, \mathcal{Z}$, and any z_0, z_1 in $\mathcal{Z}$ with $Pr\{Z = z_0\} = p > 0$, $Pr\{Z = z_1\} = q > 0$,*

$$|H(U|V, Z = z_0) - H(U|V, Z = z_1)| \leq c\sqrt{I(UV;Z)}\log|\mathcal{U}| + h\left(\min\left[c\sqrt{I(UV;Z)}, \frac{1}{2}\right]\right)$$

where $h(t) = -t\log t - (1-t)\log(1-t)$ and c is a constant depending on p and q.

Proof of theorem 7.2.1 The idea of this proof is that if there exists a secure protocol that uses a GEC n times to transmit a string of length k, and with players also connected with a noiseless channel, it has $\frac{k}{n}$ less than or equal to $\max[I(X;Y), H(X|Y)]$.

We will use the following notation (all parameters are random variables):

- $K_0, K_1 \in \{0,1\}^k$ are the strings transmitted;
- X^n is the Alice's input on GEC;
- B^n is the Bob's output on GEC;
- Z is the choice bit of Bob;
- $\hat{K}_Z$ is the estimated output of Bob and
- F is the total communication in the noisyless channel.

Without loss of generality $Z = 0$. So we have these conditions of security:

$$Pr\{\hat{K}_0 \neq \hat{K}_0\} \to 0 \tag{7.1}$$

$$I(K_0 X^n F; Z) \to 0 \tag{7.2}$$

$$\frac{1}{k} I(Y^n F; Z = 1) \to 0 \tag{7.3}$$

Note that these conditions are weaker than the conditions of security (Definition 5.1.1, Chapter 5). Making a comparison between the two conditions we have that equation (7.1) is the correctness, equation (7.2) is the security for receiver and equation (7.3) is the security of the sender.

Claim 7.2.3

$$\frac{k}{n} \leq \frac{1}{n} \sum_{t=1}^{n} I(X_t; Y_t) + \varepsilon, \quad \varepsilon \to 0.$$

Demonstration of claim 7.2.3

From lemma 7.2.2, we have

$$|H(K_0|X^n F, Z = z_0) - H(K_0|X^n F, Z = z_1)| \leq c\sqrt{I(K_0 X^n F; Z)}\log k + h\left(\min\left[c\sqrt{I(K_0 X^n F; Z)}, \frac{1}{2}\right]\right)$$

However by equation (7.2) $I(K_0 X^n F; Z) = \delta$, $h(\delta') = \delta''$ and $c\sqrt{I(K_0 X^n F; Z)}\log k = o(k)$. Then

$$\begin{aligned} & H(K_0|X^n F, Z = 0) - H(K_0|X^n F, Z = 1) = o(k) \\ \Rightarrow \quad & H(K_0|F, Z = 0) - H(K_0|F, Z = 1) = o(k) \end{aligned} \tag{7.4}$$

since conditioning reduces entropy.

On the other hand, we know that $H(K_0|Z=0) = H(K_0|Z=1) = H(K_0) = k$, because K_0 is independent of Z. Thus, by 7.4,

$$\begin{aligned} I(K_0;F|Z=0) &= H(K_0|Z=0) - H(K_0|FZ=0) \\ &= k + o(k) + H(K_0|F, Z=1) \end{aligned}$$

Nevertheless, by the equation (7.3)

$$\begin{aligned} I(Y^nF;K_0|Z=1) &= k\vartheta \\ I(Y^nF;K_0|Z=1) &= I(F;K_0|Z=1) + I(Y^n;K_0|FZ=1) \\ I(F;K_0|Z=1) &= I(Y^nF;K_0|Z=1) - I(Y^n;K_0|FZ=1) \\ I(F;K_0|Z=1) &= o(k) \end{aligned}$$

since, if the protocol is secure, $I(Y^n;K_0|FZ=1) \to 0$.

So

$$\begin{aligned} I(K_0;F|Z=0) &= k + o(k) + o(k) \\ I(K_0;F|Z=0) &= o(k) \end{aligned} \tag{7.5}$$

If the equations 7.1 and 7.5 held without conditioning on $Z=0$ then K_0 would be a secret key for Alice and Bob, with security from an eavesdropper observing the public communication F. Then we can use Theorem 3 of [43]. With our equations we had to use $I(X_t;Y_t|Z=0)$, but we can chance it to $I(X_t;Y_t)$ since $I(K_0;F|Z=0) \leq I(K_0;F)$ and the conditional distribution of X_t on the condition $Z=0$ differs negligible from the unconditional distribution.

Thus the claim follows.

Claim 7.2.4

$$\frac{k}{n} \leq \frac{1}{n}\sum_{t=1}^{n} H(X_t;Y_t) + \varepsilon, \quad \varepsilon \to 0.$$

Demonstration of claim 7.2.4

Note that $K_0 \to X^nF \to Y^nFZ$ is a Markov chain since $P_{K_0|X^nF,Y^nFZ} = P_{K_0|X^nF}$. Then by the

Fano's inequality (theorem I.0.6 on Appendix)

$$H(K_0|X^nF, Z=0) \leq H(K_0|Y^nF, Z=0) \quad = \quad o(k) \tag{7.6}$$

By (7.3) we have

$$\begin{aligned} I(Y^nF; K_0|Z=1) &= o(k) \\ H(K_0|Z=1) - H(K_0|Y^nF, Z=1) &= o(k) \end{aligned} \tag{7.7}$$

By (7.4) and (7.7) we have

$$\begin{aligned} H(K_0|X^nF, Z=1) &= H(K_0|X^nF, Z=0) - o(k) \\ H(K_0|X^nF, Z=1) &\leq H(K_0|X^nF, Z=0) \\ &\leq H(K_0|Y^nF, Z=0) = o(k) \\ \Rightarrow \quad H(K_0|X^nY^nF, Z=1) &\leq H(K_0|X^nF, Z=1) \\ &\leq o(k) \end{aligned} \tag{7.8}$$

Thus, we can say

$$\begin{aligned} k &= H(K_0|Z=1) = H(K_0|Y^nF, Z=1) + o(k), \quad \text{by 7.7} \\ &\leq H(K_0|X^nY^nF, Z=1) + H(X^n|Y^nF, Z=1) + o(k) \\ &\leq H(X^n|Y^n, Z=1) + o(k) \quad \text{by 7.8 and by } H(X^n|Y^nF, Z=1) < H(X^n|Y^n, Z=1) \\ &\leq \sum_{t=1}^{n} H(X_t|Y_t, Z=1) + o(k) \quad \text{by chain rule} \\ &\leq \sum_{t=1}^{n} H(X_t, Y_t) + o(k) \quad \text{since conditioning reduces entropy and chain rule.} \end{aligned}$$

Finally, by the claims (7.2.3 and 7.2.4)

$$\frac{k}{n} \leq \frac{1}{n}\sum_{t=1}^{n} I(X_t;Y_t) + \varepsilon \leq I(X_T;Y_T)$$
$$\frac{k}{n} \leq \frac{1}{n}\sum_{t=1}^{n} H(X_t;Y_t) + \varepsilon \leq H(X_T;Y_T)$$

where T is a RV uniformly distribution on $[n]$ independent of the RVs X_t, Y_t.

Hence,

$$\begin{aligned}
\frac{k}{n} &\leq \max[I(X;Y), H(X;Y)] \\
\frac{k}{n} &\leq \max[I(X;Y), H(X) + H(Y) - I(X;Y)] \\
\frac{k}{n} &\leq C(W_0)(1-p^*)
\end{aligned}$$

Since there exists a secure protocol that achieves the OT ratio $C(W_0)(1-p^*)$, this is the capacity of OT of a Generalized Erasure Channel with $p^* \geq \frac{1}{2}$.

□

Proof of theorem 5.2.4

This follows from Theorems 7.1.1 and 7.2.1 and the observation that the protocol in question is secure for malicious adversaries.

□

8 CONCLUSION

In this work our gosl was to implement oblivious transfer using a noisy resource efficiently. We improve the results of Ahlswede and Csiszár [16]. That paper presents oblivious transfer capacity of generalized erasure channel on honest-but-curious model; we show OT capacity of GEC in the malicious model. Namely, both models have the same OT capacity to GEC.

Our work presents some advantages as to obtain a constant round protocol, i.e., the amount of exchange of messages in the protocol is independent of size of the string obliviously transmitted - this result comes from interactive protocol of Ding et al [23]; and to obtain a protocol sequentially composable secure, since our protocol follows the definition of Crépeau and Wullschleger [18] which implies it. It means that if a protocol properly reduces to our implementation of String Oblivious Transfer, the resulting protocol also will be secure.

However, a disadvantage of our protocol is that our protocol is not computationally efficiently, because Bob uses the output of universal hash functions to correct errors in his string received from GEC. This does not impact our result since our goal was to reach OT capacity in the case considered. So, to determine a computationally efficiently protocol that achieves the OT capacity of GEC on malicious model is an open problem.

A Generalized Erasure Channel is a channel that the message to transmit can be erasure, flipped or transmitted without errors. But it is yet a specific kind of channel. Then, another open problem is to determine the OT capacity of a general channel on the malicious model. To determine OT capacity to continuous channels is a open problem too.

REFERENCES

[1] WIESNER, S. Conjugate coding. *SIGACT News*, ACM Press, v. 15, n. 1, p. 78–88, 1983.

[2] RABIN, M. O. How to exchange secrets by oblivious transfer. *Technical Report TR-81, Aiken Computation Laboratory, Harvard University*, 1981.

[3] CREPEAU, C. Equivalence between two flavours of oblivious transfers. *CRYPTO '87: A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, v. 293, p. 350 – 354, 1987.

[4] GOLDREICH, O.; MICALI, S.; WIGDERSON, A. How to play any mental game, or: A completeness theorem for protocols with honest majority. *Proceedings 19th ACM STOC*, p. 20 – 31, 1997.

[5] KILIAN, J. Founding cryptography on oblivious transfer. *STOC 1988*, p. 20–31, 1988.

[6] CREPEAU, C.; GRAAF, J. v. d.; TAPP, A. Committed oblivious transfer and private multi-party computation. *CRYPTO 1995*, p. 110–123, 1995.

[7] YAO, A. C. Protocols for secure computation. *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '82)*, p. 160 – 164, 1982.

[8] WYNER, A. D. The wire-tap channel. *Bell Syst. Tech. J*, v. 54, p. 1355 – 1387, 1975.

[9] CREPEAU, C.; KILIAN, J. Achieving oblivious transfer using weakened security assumptions (extended abstract). *FOCS 1988*, p. 42–52, 1988.

[10] CREPEAU, C. Efficient cryptographic protocols based on noisy channels. *EUROCRYPT 1997*, p. 306–317, 1997.

[11] KORJIK, V.; MOROZOV, K. K. Generalized oblivious transfer protocols based on noisy channels. *Proc. Workshop MMM ACNS 2001*, p. 219 – 229, 2001.

[12] STEBILA, D.; WOLF, S. Efficient oblivious transfer from any non-trivial binary-symmetric channel. *IEEE Int. Symp. InformationTheory (ISIT)*, p. 293, 2002.

[13] CREPEAU, C.; MOROZOV, K.; WOLF, S. Efficient unconditional oblivious transfer from almost any noisy channel. *SCN 2004*, p. 47–59, 2004.

[14] NASCIMENTO, A. C. A.; WINTER, A. On the oblivious-transfer capacity of noisy resources. *IEEE Transactions on Information Theory*, p. 2572–2581, 2008.

[15] IMAI, H.; MOROZOV, K.; NASCIMENTO, A. C. A. On the oblivious transfer capacity of the erasure channel. *2006 IEEE International Symposium on Information Theory*, p. 1428 – 1431, 2006.

[16] AHLSWEDE, R.; CSISZAR, I. On oblivious transfer capacity. *IEEE International Symposium on Information Theory*, p. 2061–2064, 2007.

[17] IMAI, H.; NASCIMENTO, A. C. A.; WINTER, A. Commitment capacity of discrete memoryless channels. *IMA Int. Conf. 2003*, p. 35–51, 2003.

[18] CREPEAU, C.; WULLSCHLEGER, J. Statistical security conditions for two-party secure function evaluation. *ICITS 2008*, 2008.

[19] OSTROVSKY, R.; VENKATESAN, R.; YUNG, M. Fair games against an all-powerful adversary. *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, p. 155 – 169, 1993.

[20] SAVVIDES, G. *Interactive Hashing and reductions between Oblivious Transfer variants*. Tese (Doutorado) — School of Computer Science, McGill University, 2007.

[21] CREPEAU, C.; SAVVIDES, G. G. Optimal reductions between oblivious transfers using interactive hashing. *EUROCRYPT 2006*, p. 201–221, 2006.

[22] CACHIN, C.; CREPEAU, C.; MARCIL, J. Oblivious transfer with a memory-bounded receiver. *Proceedings.39th IEEE Annual Symposium on Foundations of Computer Science*, p. 493 – 502, 1998.

[23] DING, Y. Z. et al. Constant-round oblivious transfer in the bounded storage model. *J. Cryptology and Conference version appeared at TCC '04*, p. 165–202, 2007.

[24] COVER, T. M.; THOMAS, J. A. *Elements of Information Theory*. [S.l.]: Wiley-Interscience, 2006.

[25] NISAN, N.; ZUCKERMAN, D. Randomness is linear in space. *Journal of Computer and System Sciences*, p. 43 – 53, 1996.

[26] DODIS, Y. et al. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput. Conference version appeared in EUROCRYPT 2004.*, p. 97–139, 2008.

[27] SHALTIEL, R. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, p. 67 – 95, 2002.

[28] RADHAKRISHNAN, J.; TA-SHMA, A. Bounds for dispersers, extractors, and depthtwosuperconcentrators. *SIAM Journal on Discrete Mathematics*, p. 2 – 24, 2000.

[29] CARTER, J. L.; WEGMAN, M. N. Universal classes of hash functions. *Journal of Computer and System Sciences*, p. 143 – 154, 1979.

[30] IMPAGLIAZZO, R.; LEVIN, L. A.; LUBY, M. Pseudo-random generation from one-way functions (extended abstracts). *STOC 1989*, p. 12–24, 1989.

[31] HASTAD, J. et al. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, p. 1364 – 1396, 1999.

[32] BENNETT, C. H.; BRASSARD, G.; ROBERT, J. Privacy amplification by public discussion. *SIAM J. Comput.*, p. 210–229, 1988.

[33] BENNETT, C. H. et al. Generalized privacy amplification. *IEEE Transactions on Information Theory*, p. 1915–1923, 1995.

[34] GOLDREICH, O. *Foundations of Cryptography. Volume II: Basic Applications*. [S.l.]: Cambridge University Press, 2004.

[35] GOLDWASSER, S.; MICALI, S.; RACKOFF, C. The knowledge complexity of interactive proof systems. *SIAM Journal on Computation*, p. 186–208, 1989.

[36] CANNETI, R. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, p. 143–202, 2000.

[37] NAOR, M. et al. Perfect zero-knowledge arguments for np using any one-way permutation. *Journal of Cryptology. Preliminary version in CRYPTO '92*, p. 87–108, 1998.

[38] GOWERS, W. T. An almost m-wise independent random permutation of the cube. *Combinatorics, Probability & Computing*, p. 119 – 130, 1996.

[39] KAPLAN, E.; NAOR, M.; REINGOLD, O. Derandomized constructions of k-wise (almost) independent permutations. *RANDOM-APPROX Š05, Lecture Notes*, p. 354 – 365, 2005.

[40] EVEN, S.; GOLDREICH, O.; LEMPEL, A. A randomized protocol for signing contracts. *Comm. ACM*, p. 637 – 647, 1985.

[41] BEAVER, D. Precomputing oblivious transfer. *Lecture Notes in Computer Science - CRYPTO 95*, p. 97–109, 1995.

[42] COVER, T. M. Enumerative source encoding. *IEEE Transactions on Information Theory*, p. 73 – 77, 1975.

[43] MAURER, U. M. Secret key agreement by public discussion. *IEEE Transactions on Information Theory*, p. 733 – 742, 1993.

[44] CHERNOFF, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statistics*, p. 493–507, 1952.

APPENDIX

I. FORMAL TECHNICALITIES

Lemma I.0.5 (Chernoff Bound [44]) *For i.i.d random variables $X_1, \ldots, X_n$ with $0 \leq X_n \leq 1$ and with expectation $E(X_n) = p$:*

$$Pr\left\{\frac{1}{N}\sum_{n=1}^{N} X_n \geq (1+\eta)p\right\} \leq \exp\left(-N\frac{p\eta^2}{2\ln 2}\right),$$

$$Pr\left\{\frac{1}{N}\sum_{n=1}^{N} X_n \leq (1-\eta)p\right\} \leq \exp\left(-N\frac{p\eta^2}{2\ln 2}\right).$$

Theorem I.0.6 (Fano's Inequality) *For any estimator $\hat{X}$ such that $X \to Y \to \hat{X}$ with $P_e = Pr(\hat{X} \neq X)$, we have*

$$H(P_e) + P_e \log|\mathcal{X}| \geq H(X|\hat{X}) \geq H(X|Y).$$