Information-Theoretically Secure Oblivious Polynomial Evaluation in the Commodity-Based Model

Rafael Tonicelli[1], *IEEE Member*,
Anderson C. A. Nascimento[1], Rafael Dowsley [2], Jörn Müller-Quade[3],
Hideki Imai[4], *IEEE Fellow*, Goichiro Hanaoka[4], Akira Otsuka[4]

Abstract

Oblivious polynomial evaluation (OPE) consists of a two-party protocol where a sender inputs a polynomial $p(x)$, and a receiver inputs a single value x_0. At the end of the protocol, the sender learns nothing and the receiver learns $p(x_0)$. This paper deals with the problem of oblivious polynomial evaluation under an information-theoretic perspective, which is based on recent definitions of Unconditional Security developed by Crépeau *et al.* [7]. In this paper, we propose an information-theoretic model for oblivious polynomial evaluation relying on pre-distributed data, and prove very general lower bounds on the size of the pre-distributed data, as well as the size of the communications in any protocol. It is demonstrated that these bounds are tight by obtaining a round-optimal OPE protocol, which meets the lower bounds simultaneously. We present a natural generalization to OPE called oblivious linear functional evaluation. Additionally, the proposed model is applied to solving the Oblivious Equality Testing Problem.

Keywords: Information-theoretic cryptography, cryptographic primitives, oblivious polynomial evaluation, commodity-based model.

[1]Department of Electrical Engineering, University of Brasilia, Campus Darcy Ribeiro, 70910-900, Brasilia, Brazil. *E-mail addresses*: tonicelli@redes.unb.br, andclay@ene.unb.br.

[2]Department of Computer Science and Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. *E-mail address*: rdowsley@cs.ucsd.edu.

[3]Institut für Kryptographie und Sicherheit, Karlsruhe Institute of Technology, Am Fasanengarten 5, 76128 Karlsruhe, Germany. *E-mail address*: mueller-quade@kit.edu.

[4]National Institute of Advanced Industrial Science and Technology (AIST), 1-18-13, Sotokanda, Chyioda-ku, 101-0021, Tokyo, Japan.*E-mail addresses*:{h-imai,hanaoka-goichiro,a-otsuka}@aist.go.jp.

1. Introduction

1.1. Secure Function Evaluation

Assume the existence of n players, $1, \ldots, n$; each player i has a private input x_i, which is known only to him/her. Their goal is to collaboratively compute $f(x_1, \ldots, x_n)$ in such a way that no player has to reveal unnecessary information about his/her input. A protocol allowing two or more parties to achieve this goal and satisfying both the correctness and the privacy constraints is called a *secure function evaluation protocol*. The correctness constraint implies that the values the protocol returns are correct, even if some party in the system fails (i.e., the faulty party can choose his/her input to the function, but cannot force the protocol to output a wrong value as the result of the function evaluation); and the privacy constraint implies that the joint computation of $f(x_1, \ldots, x_n)$ does not reveal to each participant i more information that can be deduced from $f(x_1, \ldots, x_n)$ and his/her own private input x_i.

There are two main ways of defining the security of a cryptographic system: *Information-Theoretic Security* (also called *Unconditional Security*) and *Computational Security*. For the former no assumption is assumed about the computational power of the adversary. For the later, the security is defined in terms of an adversary with limited computational resources. In order to prove that a system is computationally secure, it is necessary to invoke certain unproven intractability assumptions, e.g., the hardness of computing the discrete logarithm. In contrast, information-theoretically secure systems do not rely on any hypotheses about the complexity of certain problems, but rely on physical assumptions, e.g., the existence of noisy channels. In spite of being considered less practical, information-theoretic security is a permanent and stronger definition of security. This work focuses on the oblivious polynomial evaluation problem from an information-theoretic point of view, and is based on formal definitions of unconditionally secure evaluation schemes proposed by Crépeau *et al.* [7] which corrected many drawbacks present in several ad-hoc definitions of security proposed in the past.

1.2. Oblivious Transfer

Oblivious Transfer (OT), a cryptographic primitive introduced by Rabin [14], is of particular interest in secure multi-party computation. It has been proven that any function can be evaluated unconditionally securely if oblivious transfer is available [11]. This property is called *completeness*.

A useful variant of oblivious transfer is *one-out-of-n string OT*, which is denoted by $\binom{n}{1}$-OT^k. It allows a sender to send n strings ($x_0, \ldots, x_{n-1}$) of length k to a receiver, who is allowed to learn one of them according to his choice c. This process is illustrated in figure 1.

An OT protocol is said to be correct, if for honest players, the receiver obtains the desired output x_c and both players do not abort the protocol. It is said to be private

if the sender learns no information on the receiver's choice c, while the receiver gets information concerning at most one of the sender's inputs.

It has been proven that unconditionally secure OT is impossible to achieve without further assumptions. Traditional assumptions used with this purpose include noisy communications and correlations. In an ingenuous approach, Rivest [15] demonstrated that the use of a trusted third party, who pre-distributes data to the players, also allows one to design unconditionally secure OT protocols.

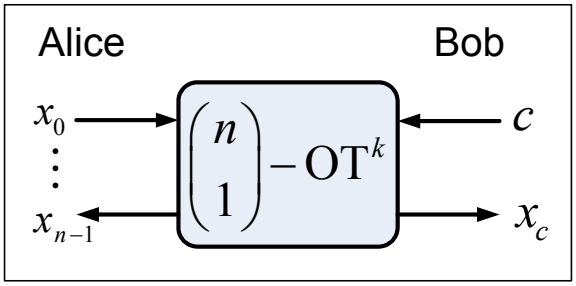

Figure 1: One-out-of-n string OT.

1.3. Oblivious Polynomial Evaluation

Oblivious Polynomial Evaluation (OPE) is a variant of Oblivious Function Evaluation and was introduced in [13]. Similarly to OT, OPE is a very useful tool for achieving secure distributed computations.

OPE is a two-party protocol where a sender (Alice) inputs a polynomial over a finite field and a receiver (Bob) inputs a single point of the same finite field. At the end of the protocol, Alice receives nothing and Bob should receive the polynomial input by Alice evaluated on the point chosen by him. The protocol is secure if Bob evaluates the polynomial input by Alice on at most one point and Alice learns nothing on which point was chosen by Bob.

One can identify several applications where OPE is particularly valuable, such as protocols for private comparison of data and protocols for anonymity preservation. Some examples include: solutions for the list intersection problem, private metering, anonymous coupons, among others.

Since its introduction in [13] OPE has been extensively studied. In [6] the problem of implementing OPE was efficiently reduced to that of achieving OT. Also, in [6] an information-theoretically secure protocol for implementing OPE was proposed. The security of that protocol was based on the trustiness of a third party which took an active role in the protocol execution.

In this paper, we analyze the problem of achieving unconditionally secure oblivious polynomial evaluation without using an active (on-line) trusted party.

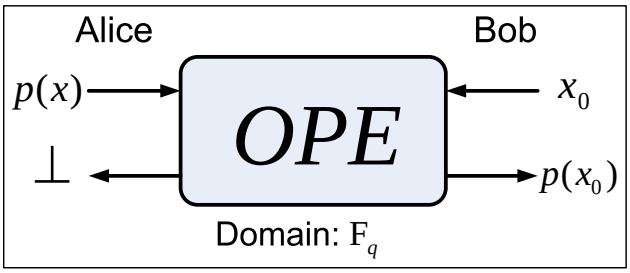

Figure 2: Oblivious Polynomial Evaluation. Note that $\mathbb{F}_q$ denotes a finite field.

1.4. Commodity-Based Cryptography

Many security schemes demand an active server for intermediating the interactions among the participants of the protocol. Thus, the information exchanged among the participants will depend on the reliability and trustiness of the server during all the protocol execution. One alternative to this is the so-called *Commodity-Based Cryptography*, introduced by Beaver [1].

The protocols proposed in this paper rely on the commodity cryptographic model, where players buy cryptographic primitives from "off-line" servers, usually called *trusted initializers* (TIs). These primitives can be used later on to implement general cryptographic protocols. The commodity-based model was inspired on the Internet architecture, which is usually based on the "client-server" paradigm. Once the primitives, or commodities as they are called by Beaver, are acquired, no further interactions between server and users are required (figure 3). Therefore, the servers need not to know the values which are computed by the players. Moreover, if several servers are available, they need not to be completely trusted, that is, the system is secure even if some servers collude with the users of the commodities and/or among themselves. Another interesting feature of Beaver's model is that no interaction among the servers is required.

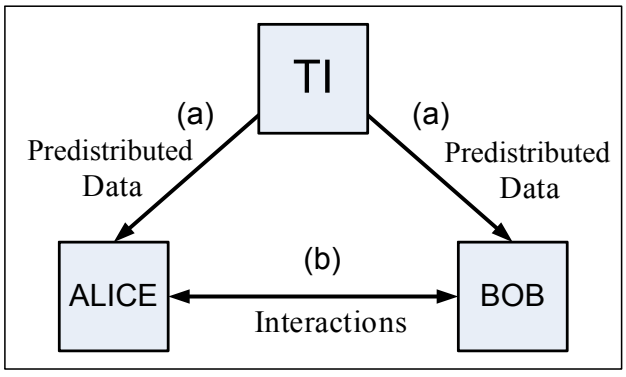

Figure 3: Process **(a)** represents the setup phase and process **(b)** represents the interactions where no further intervention of the commodity server is needed.

In this contribution, we show that the use of "off-line" servers provides very efficient and simple protocols for secure oblivious polynomial evaluation over a finite field.

Although this model was formalized just in [1], several independent works share the same flavor. We cite key-pre-distribution schemes [12], unconditionally secure bit commitments [15, 5] and unconditionally secure digital signature schemes [10].

1.5. Contributions and Related Works

Seminal works on secure function evaluation and OPE were computationally secure. For instance, Naor and Pinkas proposed in [13] an OPE scheme which was based on the intractability assumption of *noisy polynomial interpolation*. Later, Bleichenbacher and Nguyen demonstrated in [4] that this assumption could be less strong than expected and proposed a new intractability assumption based on the *polynomial reconstruction problem*. While the hardness of these problems remains an open question in the foundations of Computer Science, our OPE model is information-theoretically secure, i.e., it is secure even against a computationally unbounded adversary and does not rely on unproven computational hypotheses.

Recently, Crépeau *et al.* [7] constructed a new formal definition of unconditional security, which is based on the *ideal/real model paradigm*, and established conditions for two-party secure function evaluation in a scenario where the players have infinite computational resources. By proving the security of our model, this work aims at revisiting the problem of oblivious polynomial evaluation from this new information-theoretic point of view.

We propose and solve the problem of implementing information-theoretically secure OPE in the commodity-based cryptography model [1]. Our solution is optimal in terms of communication complexity.

We provide a model (section 2), bounds for the amount of memory which is required from players taking part in the protocol (section 3) and a construction which achieves these bounds, thus showing their tightness (section 4).

Finally, we propose a more general protocol called *oblivious linear functional evaluation* (OLF) in section 5. In OLF Alice inputs a linear functional while Bob evaluates this linear functional on a vector of his choice. As a side result of our bounds, we prove the optimality of oblivious transfer protocols proposed by Rivest [15] and Beaver [1]. We also demonstrate that these ideas can be used to solve the "Oblivious Equality Testing Problem" (section 6).

2. Definitions

In this section, the general OPE model and important definitions used throughout the text are provided. These definitions include the security requirements for OPE realization and the scenarios in which our model is applicable.

2.1. Notation

In the following, we denote random variables by upper-case letters (X) and their realizations by lower-case letters (x). The set of values taken by a random variable is denoted by calligraphic letters ($\mathcal{X}$) and we use $|\cdot|$ to denote its corresponding cardinality ($|\mathcal{X}|$). The Shannon entropy of a random variable X is denoted by $H(X)$ and the mutual information of two random variables X and Y is denoted by $I(X;Y)$. Similarly, $I(X;Y|Z)$ and $H(X|Z)$ denote the conditional mutual information and the conditional entropy when conditioned on the random variable Z.

2.2. Security Definitions

A two-party protocol consists of a program which describes a series of messages to be exchanged and local computations to be performed by the two parties. The protocol is said to halt if no more local computations or message exchanges are required. At the end of an execution of a protocol, each party emits an accept/reject message, depending on the messages he/she received and on the result of local computations.

Defining the security of a two-party protocol, where oblivious polynomial evaluation is an important special case, represents a challenging task. We consider scenarios where the parties are computationally unbounded and the existence of *active* and *passive* adversaries. An active (or malicious) adversary may change his/her behavior arbitrarily and cooperate in order to disrupt the correctness and privacy of the computation. On the other hand, a passive (or semi-honest) adversary is the one who follows the protocol, but may try to acquire more information than what he/she is allowed to know.

The definitions for information-theoretically secure two-party function evaluation used in this text are strongly related to the *real/ideal model paradigm*. In the ideal model, the parties are admitted to have access to a trusted third party, who would receive their private inputs, compute the outcome of the desired functionality f and send to each party the corresponding output. In the real model, no trusted party for computing the functionality f exists (possibly the parties have access to some functionality g), and the mutually distrustful parties should run some protocol to compute f. Intuitively speaking, if the real life protocol can emulate the ideal model, the protocol is said to be secure. In other words, a real life protocol is considered secure, if no adversary can cause more damage in a real execution than an ideal adversary can cause in an execution of the ideal protocol. Thus, if a protocol is secure according to this paradigm, an attack against the real life protocol has an effect similar to an attack against the ideal model, where the participants have only a black-box access to the desired functionality.

We shall now define when a protocol perfectly securely evaluates a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{U} \times \mathcal{V}$. To accomplish this task, we will use the formalism and definitions of [7]. Let $x \in \mathcal{X}$ be the input of the first player and $y \in \mathcal{Y}$ the input of the second player. Consider also an additional auxiliary input $z \in \{0,1\}^*$ that can be potentially used by both players. For instance, this auxiliary input can be the data generated by previous

protocol executions, or any other information that could give an illegal advantage to a dishonest party. Thus, an honest player will ignore this additional input. A g-hybrid protocol consists of a pair of algorithms $\Pi = (A_1, A_2)$ that can interact by means of two-way message exchange and have access to some functionality g. A pair of algorithms $\overline{A} = (\overline{A}_1, \overline{A}_2)$ is admissible for protocol Π if at least one of the parties is honest, that is, if at least one of the equalities $\overline{A}_1 = A_1$ and $\overline{A}_2 = A_2$ is true. Note that no security is required when both parties are dishonest $((\overline{A}_1 \neq A_1) \wedge (\overline{A}_2 \neq A_2))$.

The Real Model. In the real model, the players have no access to a trusted intermediary and must compute the desired functionality by means of a g-hybrid protocol $\Pi = (A_1, A_2)$. Let $\overline{A} = (\overline{A}_1, \overline{A}_2)$ be an admissible pair of algorithms for the protocol Π. The joint execution of Π under $\overline{A}$ in the real model,

$$\textsc{Real}^{g}_{\Pi,\overline{A}(z)}(x, y),$$

denotes the resulting output pair, given the input pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the auxiliary input z and the functionality g used by the admissible pair $\overline{A}$.

The Ideal Model. In the ideal model, both players have access to a trusted third party to evaluate the functionality f. The trivial protocol $B = (B_1, B_2)$ is the protocol where both parties send their inputs to the functionality f and output the values that the functionality f outputs to them. The algorithms $\overline{B}_1$ and $\overline{B}_2$ of the protocol $\overline{B} = (\overline{B}_1, \overline{B}_2)$ receive the inputs x and y, respectively, and the auxiliary input z. The algorithms send the values x' and y' to the trusted party, who returns the value $(u', v') = f(x', y')$. Finally, $\overline{B}_1$ and $\overline{B}_2$ output the values U and V. Let $\overline{B} = (\overline{B}_1, \overline{B}_2)$ be an admissible pair of algorithms for B. The joint execution of f under $\overline{B}$ in the ideal model on input pair (x, y) and auxiliary input z, given by

$$\textsc{Ideal}_{f,\overline{B}(z)}(x, y),$$

represents the output pair that results from the interaction between $\overline{B}_1(x, z)$ and $\overline{B}_2(y, z)$ under the functionality f.

Figure 4 illustrates the abstraction of an admissible protocol $\overline{B}$ in the ideal model. At first, Alice receives input (x, z) and Bob receives input (y, z). Then, the parties produce the values $\overline{B_1^{IN}}(x, z) = (x', z_1)$ and $\overline{B_2^{IN}}(y, z) = (y', z_2)$. The parties send the inputs x' and y' to the trusted entity. The trusted entity performs the desired computation $f(x', y') = (u', v')$, and sends u' to Alice and v' to Bob. Upon having in their possession the outcomes u' and v' and the auxiliary inputs z_1 and z_2, Alice outputs $\overline{B_1^{OUT}}(u', z_1) = u$ and Bob outputs $\overline{B_2^{OUT}}(v', z_2) = v$. Note that when one of the parties is honest, he/she does not perform any modification on his/her inputs nor on his/her outputs. Additionally, auxiliary inputs are not used by honest participants.

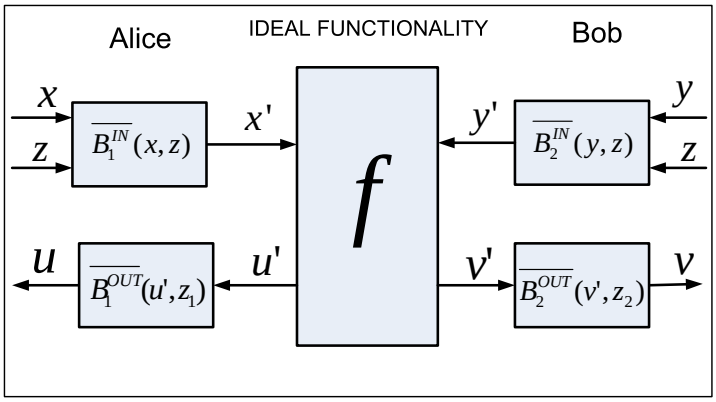

Figure 4: Illustration of an admissible protocol $\overline{B}$ in the ideal model, where the parties have access to a trusted external entity.

Figure 5 exhibits the corresponding simplification of the model when one of the parties behaves honestly.

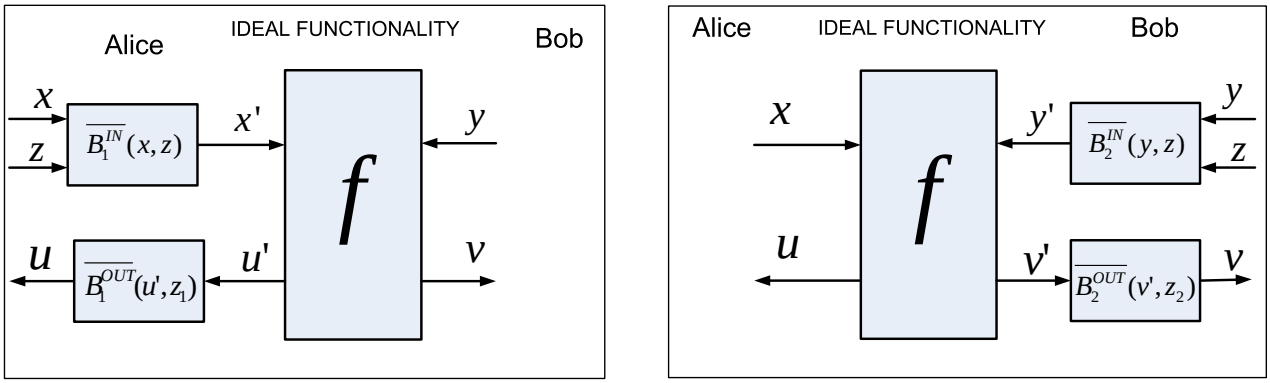

Figure 5: Simplification of the ideal model for an honest Bob and an honest Alice, respectively.

Definition 1 (Perfect Security)**.** *A g-hybrid protocol* Π *evaluates a function* f *perfectly securely if for every admissible pair of algorithms* $\overline{A} = (\overline{A}_1, \overline{A}_2)$ *in the real model for the protocol* Π*, there exists an admissible pair of algorithms* $\overline{B} = (\overline{B}_1, \overline{B}_2)$ *in the ideal model for the trivial protocol* B*, such that*

$$\textsc{Real}^{g}_{\Pi,\overline{A}(z)}(x,y) \equiv \textsc{Ideal}_{f,\overline{B}(z)}(x,y).$$

for all input pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$ *and auxiliary input* $z \in \{0,1\}^*$*. Note that the symbol* $\equiv$ *denotes identical distributions.*

Next we present a theorem from [7] which states the conditions for secure function evaluation.

Theorem 1. *A protocol* Π *is said to securely evaluate the deterministic functionality* f *perfectly, if and only if for every pair of algorithms* $\overline{A} = (\overline{A}_1, \overline{A}_2)$ *that is admissible in the real model for the protocol* Π *and for all inputs* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *and for all auxiliary input* $z \in \{0,1\}^*$, $\overline{A}$ *produces outputs* (U, V), *such that the following conditions are satisfied:*

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

$$(U, V) = f(X, Y).$$

- (Security for Alice) *If Alice is honest then there exist random variables* Y' *and* V' *such that* $(U, V') = f(X, Y')$,

$$I(X; Y'|ZY) = 0, \text{ and } I(UX; V|ZYY'V') = 0.$$

- (Security for Bob) *If Bob is honest then there exist random variables* X' *and* U' *such that* $(U', V) = f(X', Y)$,

$$I(Y; X'|ZX) = 0, \text{ and } I(VY; U|ZXX'U') = 0.$$

The security definitions are now applied to the oblivious polynomial evaluation problem. The ideal functionality f_{OPE} is denoted by

$$f_{\text{OPE}}(P, X_0) := (\perp, P(X_0))$$

such that $X_0, P(X_0) \in \mathbb{F}_q$, where $\mathbb{F}_q$ is a finite field, P is a polynomial defined over $\mathbb{F}_q$ and $\perp$ denotes a constant random variable. The random variables P and X_0 can present arbitrary probability distributions.

The application of our security definitions to the specific case of OPE yields to the following theorem.

Theorem 2. *A protocol* Π *realizes an OPE perfectly securely if and only if for every admissible pair of algorithms* $\overline{A} = (\overline{A}_1, \overline{A}_2)$ *for protocol* Π *and for all inputs* (P, X_0) *and auxiliary input* Z, $\overline{A}$ *produce outputs* (U, V) *such that the following conditions are satisfied:*

- (Correctness) *If both players are honest, then*

$$(U, V) = (\perp, P(X_0)).$$

- (Security for Alice) *If Alice is honest, then we have* $U = \perp$ *and there exists a random variable* X_0'*, such that*

$$I(P; X_0'|ZX_0) = 0, \text{ and } I(P; V|ZX_0X_0'P(X_0')) = 0.$$

- (Security for Bob) *If Bob is honest, then we have*

$$I(X_0; U|ZP) = 0.$$

Proof. We have to prove the equivalence between the privacy conditions for Bob in Theorems 1 and 2. This proof is analogous to the one presented in [7] for one-out-of-n string OT.

According to Theorem 1, we must have that

$$I(X_0; P'|ZP) = 0 \text{ and } I(P(X_0)X_0; U|ZPP'U') = 0$$

or equivalently,

$$I(X_0; P'|ZP) + I(P'(X_0)X_0; U|ZPP') = 0.$$

$P'(X_0)$ is a function of X_0 and the polynomial P' so

$$\begin{aligned} I(P'(X_0)X_0; U|ZPP') &= I(X_0; U|ZPP') + I(P'(X_0); U|X_0ZPP') \\ &= I(X_0; U|ZPP'). \end{aligned}$$

Then, the expression $I(P'(X_0)X_0; U|ZPP') = 0$ is equivalent to $I(X_0; U|ZPP') = 0$.

Applying the chain rule for mutual information we obtain

$$\begin{aligned} I(X_0; P'|ZP) + I(X_0; U|ZPP') &= I(X_0; P'U|ZP) \\ &= I(X_0; U|ZP) + I(X_0; P'|ZPU) \\ &= I(X_0; U|ZP). \end{aligned}$$

The last equality follows from the fact that, in a secure OPE implementation, P' and X_0 have to be independent given ZPU. One can observe that, at the beginning of the protocol, Bob chooses his entry X_0 independently from P', the polynomial input actually supplied by Alice. Furthermore, we should require that the action of providing to the protocol a different input does not lead to any advantage to a malicious Alice. I.e., given that Alice knows ZPU, her knowledge on P' does not give her any additional information on Bob's input X_0. Mathematically, X_0, ZPU and P' will form a Markov Chain:

$$X_0 \leftrightarrow ZPU \leftrightarrow P' \text{ and } \Pr(X_0|ZPU, P') = \Pr(X_0|ZPU) \Rightarrow I(X_0; P'|ZPU) = 0.$$

Therefore, the security condition for an honest Bob is reduced to:

$$I(X_0; U|ZP) = 0.$$

□

2.3. Important Observation on the Auxiliary Inputs

It is important to notice that the security conditions in theorems 1 and 2 must hold for all probability distributions of the inputs (X, Y). More specifically, the security conditions have to be valid for any input distribution $P_{XY|Z=z}$. Thus, all the underlying requirements are conditioned on the random variable Z. In order to prove the security of a protocol, one has to prove that the conditions are satisfied for all distributions P_{XY}, omitting the random variable Z in all the expressions.

Moreover, in [8], Crépeau and Wullschleger demonstrated that if a protocol is unconditionally secure against adversaries without auxiliary input, then it is also unconditionally secure against adversaries with auxiliary input.

As a consequence of this analysis, we will ignore the random variable Z henceforth.

2.4. Commodity-Based OPE

In our model we have three players: Alice, Bob and Ted. We assume that the players are interconnected by private pairwise channels. The adversary is malicious and may deviate from the original protocol in an arbitrary way. Ted is a trusted center who pre-distributes some secret data to Alice and Bob during a setup phase, but does not take part in the protocol later on. The data received by Alice and Bob is denoted by the random variables $U_a \in \mathcal{U}_a$ and $U_b \in \mathcal{U}_b$, respectively. The pre-distributed data are chosen independently of the inputs.

In the computing phase, Alice and Bob interact in order to perform an OPE protocol. We assume that Alice and Bob are randomized players that are supplied by independent sources of randomness. By using this approach, we simplify notation. In our model, without loss of generality, we consider that Bob initiates the communications between them. The data transferred from Bob to Alice is denoted by the random variable $E = \theta_b(X_0, U_b)$, where $\theta_b(\cdot)$ is a publicly known deterministic function. Similarly, all the data transferred from Alice to Bob is denoted by $R = \theta_a(E, P, U_a)$, where $\theta_a(\cdot)$ is a publicly known deterministic function. As usual, we consider that the messages exchanged by the players are taken from $\{0,1\}^*$. The crucial ingredient of our approach is that we use U_a and U_b as one-time pad encryption keys.

$$\begin{array}{cc} \textit{Setup} & \textit{Computing} \\ \left\{\begin{array}{l} \text{Ted} \longrightarrow \text{Alice: } U_a \\ \text{Ted} \longrightarrow \text{Bob: } U_b \end{array}\right. & \left\{\begin{array}{l} \text{Bob} \longrightarrow \text{Alice: } E = \theta_b(X_0, U_b) \\ \text{Alice} \longrightarrow \text{Bob: } R = \theta_a(E, P, U_a) \end{array}\right. \end{array}$$

Note that in this way, all the messages generated by Alice and Bob are well-defined random variables, depending on the polynomial P defined over $\mathbb{F}_q$ that Alice chose and on the evaluation point $X_0 \in \mathbb{F}_q$ that Bob chose. Assuming that both parties behave honestly, the views of Alice, $\textsc{View}_A$, and Bob, $\textsc{View}_B$, after the protocol execution will be given by:

$$U = \textsc{View}_A = \{U_a, P, E, R\} \text{ and } \textsc{View}_B = \{U_b, X_0, E, R, P(X_0)\}$$

3. Bounds

3.1. Remarks on the Adversarial Strategy

More generally, the privacy conditions for an unconditionally secure two-party protocol can be translated into the following mathematical requirements:

$$\begin{cases} I(\textsc{View}_A; V|\ \textsc{View}_B) = 0\ , \text{thus}\ \ \textsc{View}_A \leftrightarrow \textsc{View}_B \leftrightarrow V. \\ I(\textsc{View}_B; U|\ \textsc{View}_A) = 0\ , \text{thus}\ \ \textsc{View}_B \leftrightarrow \textsc{View}_A \leftrightarrow U. \end{cases}$$

It means that, if a given protocol fulfills the preceding requirements, the optimal strategy for a corrupted player, who wishes to gain additional information on the other player's input, is to apply some arbitrary processing function (say $\varphi(\cdot)$) over his/her protocol view. So, we assume that the outputs, U and V, of a corrupted Alice and a corrupted Bob are processed versions of their protocol views, $\textsc{View}_A$ and $\textsc{View}_B$, respectively:

$$U = \varphi_A(\textsc{View}_A)\ \text{ and }\ V = \varphi_B(\textsc{View}_B)$$

3.2. Computation of the Optimal Bounds

In the sequel, we prove lower bounds on the memory and communication costs for oblivious polynomial evaluation in the commodity-based model. Since we are interested on OPE protocols that can be used with any input probability distribution, we assume that the input probability distribution has some properties (P and X_0 are independent and uniformly distributed random variables). To the specific task of calculating these lower bounds, we consider semi-honest adversaries. In section 4, we prove that these bounds are tight by presenting a protocol, secure against active cheating, that achieves all of them.

It is natural to think that, in our scenario, if Bob is given access to Alice's secret data U_a, he would be able to break the secrecy condition completely, that is he should be able to learn all the information about Alice's input P. We formally prove this fact in the next proposition.

Proposition 1. *Bob learns all the information on P if he is given access to Alice's pre-distributed data U_a after completing a successful execution of oblivious polynomial evaluation. Mathematically, $H(P|ERU_aU_b) = 0$.*

Proof. Assume that a successful OPE execution, such that $P = p(x)$ and $ER = er$, has taken place. After obtaining Alice's pre-distributed data, Bob can try to compute $ER = er$ for all the possible inputs. The correct input will produce a transcript equal to the one obtained during the protocol execution. Furthermore, as previously stated $R = \theta_a(E, P, U_a)$ and $E = \theta_b(X_0, U_b)$, where U_a and U_b are uniformly distributed random variables generated by Ted independently from X_0. As a result, the joint

random variable ERU_a does not provide any information regarding Bob's input X_0. Mathematically, one can observe that

$$H(X_0|ERU_aP) = H(X_0|P) = H(X_0),$$

where the last equality follows from the fact that P and X_0 are independent. It follows that no two different polynomials should produce the same view, otherwise Alice would obtain knowledge on Bob's inputs (if two polynomials produce the same transcript, Bob's choice must be limited to the points where those polynomials coincide). □

An equivalent result holds for Alice: if she is given access to the secret data that Bob received from Ted, she is able to completely break Bob's privacy condition, i.e., she learns X_0.

Proposition 2. *Alice learns the point which was chosen by Bob if she is given access to Bob's pre-distributed data. Mathematically,* $H(X_0|ERU_aU_b) = 0$.

Proof. Assume a successful OPE execution, such that $P = p(x)$ and $ER = er$. After this real execution is finished, we know from proposition 1 that $H(P|ERU_aU_b) = 0$. Then, Alice can simulate Bob's inputs and determine those that are compatible with the transcript $ER = er$. By the security condition for Alice, there cannot be two different values α_1 and α_2 compatible with the transcript, otherwise the correctness condition would allow Bob to discover $p(\alpha_1)$ and $p(\alpha_2)$ (violating Alice's security). So it follows that $H(X_0|ERU_aU_b) = 0$. □

We now prove another auxiliary result: namely, that the messages exchanged are independent of Alice's and Bob's inputs P and X_0.

Proposition 3. *In a secure commodity-based oblivious polynomial evaluation protocol,* $I(PX_0; ER) = 0$. *In particular,* $H(P|ER) = H(P)$.

Proof. We start by rewriting the mutual information of interest by applying the chain rule for mutual information:

$$I(PX_0; ER) = I(P; ER) + I(X_0; ER|P)$$

According to our design principles, $R = \theta_a(E, P, U_a)$ and $E = \theta_b(X_0, U_b)$, where U_a and U_b are uniformly distributed random variables that are independent from the parties' inputs (P and X_0). Resultantly, the joint random variable ER is independent from the parties'inputs, P and X_0, and does not supply any information on them. As a deduction of this analysis:

$$0 \leq I(X_0; ER|P) \leq I(X_0; ER) = 0 \Rightarrow I(X_0; ER|P) = 0.$$

and

$$I(P; ER) = 0 \Rightarrow H(P|ER) = H(P).$$

Thus, it follows that $I(PX_0; ER) = 0$ what proves our claim.

□

Another auxiliary result is actually just a corollary of proposition 3:

Proposition 4. *In any unconditionally secure commodity-based polynomial evaluation protocol,* $H(X_0P(X_0)|ER) = H(X_0P(X_0)) = H(X_0) + H(P(X_0)|X_0)$.

Proof. According to the first part of proposition, 3: $I(PX_0; ER) = 0$. Rewriting this expression, we get

$$\begin{aligned} I(PX_0; ER) &= I(PX_0P(X_0); ER) \\ &= I(X_0P(X_0); ER) + I(P; ER|X_0P(X_0)) \end{aligned}$$

The second part of proposition 3 states that $I(P; ER) = 0$. As a deduction, one can observe that:

$$0 \leq I(P; ER|X_0P(X_0)) \leq I(P; ER) = 0 \;\Rightarrow\; I(P; ER|X_0P(X_0) = 0.$$

Finally,

$$I(X_0; P(X_0); ER) = I(PX_0; ER) = 0 \;\Rightarrow\; H(X_0P(X_0)|ER) = H(X_0P(X_0)).$$

□

Now, we use the previous propositions to prove a lower bound on the size of the data which is pre-distributed to Alice.

Theorem 3. *In any unconditionally secure commodity-based OPE protocol, the size of the data which is pre-distributed to Alice is as large as the size of the polynomial to be evaluated. In other words,* $H(U_a) \geq H(P)$.

Proof. Consider $I(U_a; P|ERU_b)$. On the one hand we can rewrite it as

$$\begin{aligned} I(U_a; P|ERU_b) &= H(P|ERU_b) - H(P|ERU_aU_b) \\ &= H(P) - 0. \end{aligned} \tag{1}$$

Equation 1 follows from propositions 1 and 3 and the fact that P is independent of U_b.

On the other hand, one can observe that

$$I(U_a; P|ERU_b) \leq H(U_a|ERU_b) \leq H(U_a).$$

Thus, we conclude that $H(U_a) \leq H(P)$.

□

Here, we show a bound on the size of the data pre-distributed to Bob.

Theorem 4. *In any unconditionally secure commodity-based OPE protocol, the size of the data which is pre-distributed to Bob is bounded by the following expression: for any* $X_0 \in \mathbb{F}_q$, $H(U_b) \geq H(X_0) + H(P(X_0)|X_0)$.

Proof. Consider the following expression:

$$\begin{aligned} I(U_b; P(X_0)X_0|ERU_a) &= H(P(X_0)X_0|ERU_a) - H(P(X_0)X_0|ERU_aU_b) \\ &= H(X_0) + H(P(X_0)|X_0) - 0 \end{aligned}$$

using proposition 4 for the first entropy term, and proposition 2 (plus correctness of the protocol) for the second: X_0 is a function of E, R, U_a and U_b, and all these data together determine the polynomial value $P(X_0)$. On the other hand,

$$I(U_b; P(X_0)X_0|ERU_a) \leq H(U_b|ERU_a) \leq H(U_b),$$

which, put together with our previous identity, proves the theorem. □

We end this section with bounds on the size of the messages which have to be exchanged between Alice and Bob.

Theorem 5. *Let* $R \in \{0,1\}^*$ *and* $E \in \{0,1\}^*$ *be the random variables denoting, respectively, Alice's and Bob's communication during the computing phase. An unconditionally secure OPE protocol presents the following bounds:*

$$H(E) \geq H(X_0) \text{ and } H(R) \geq H(P).$$

Proof. For the first bound, use proposition 2 for the first step in the following chain and then independence of X_0 and RU_aU_b:

$$\begin{aligned} H(X_0) &= I(X_0; ERU_aU_b) \\ &= I(X_0; RU_aU_b) + I(X_0; E|RU_aU_b) \\ &= I(X_0; E|RU_aU_b) \leq H(E|RU_aU_b) \leq H(E) \end{aligned}$$

For the second one, use proposition 1 for the first step in the following chain and then independence of P and EU_aU_b:

$$\begin{aligned} H(P) &= I(P; ERU_aU_b) \\ &= I(P; EU_aU_b) + I(P; R|EU_aU_b) \\ &= I(P; R|EU_aU_b) \leq H(R|EU_aU_b) \leq H(R). \end{aligned}$$

□

4. An Optimal Construction

In this section we present a construction based on polynomials over finite fields which matches the lower bounds we proved in the last section and is round optimal, thus proving their tightness. The intuition behind the protocol is that Ted distributes a random evaluation performed on a random polynomial to Alice and Bob during a setup phase. Later on, they will exchange messages to turn the random evaluation into the desired one. The protocol is described below.

Protocol OPE

Setup Phase:

• Ted selects with uniform randomness a polynomial $s(x)$ of degree n and a point $d \in \mathbb{F}_q$.

• Ted sends $s(x)$ to Alice and $\{d; g = s(d)\}$ to Bob.

Computing Phase:

$$\begin{cases} \text{Alice's input: } p(x) \text{ of degree } n. \\ \text{Bob's input: } x_0 \in \mathbb{F}_q. \end{cases}$$

• Bob sends $t = x_0 - d$ to Alice.

• Alice computes $f(x) = p(x + t) + s(x)$ and sends it to Bob.

• Bob computes $f(d) - g = p(d+t) + s(d) - s(d) = p(x_0)$, the desired output.

Theorem 6. *The above stated protocol is a secure implementation of an oblivious polynomial evaluation protocol. Moreover, it is optimal regarding its space complexity.*

Proof.
(*Correctness*)
It is easily verifiable the correctness of the protocol. Considering both parties to be honest, we obtain

$$f(d) - g = p(d + t) + s(d) - s(d) = p(x_0)$$

which proves the correctness property.

(*Security for Alice*)
Let Alice be honest and $x_0' = d + t$. Then,

$$I(P; X_0'|X_0) = I(P; D + T|X_0) = 0$$

since D is independent of P.

Now we demonstrate that the second condition for Alice is satisfied. Let $\text{VIEW}_B = \{D, G, T, F, X_0, X_0', P(X_0')\}$ be Bob's view of the protocol execution. Bob's output V will be a processed version of VIEW_B, consequently,

$$\begin{aligned}
I(P; V|X_b X_b' P(X_b')) &\leq I(P; \text{VIEW}_B|X_0 X_0' P(X_0')) \\
&= I(P; DGTF|X_0 X_0' P(X_0')) \\
&= I(P; DGF|X_0 X_0' P(X_0')) \qquad (2) \\
&= I(P; DG|X_0 X_0' P(X_0')) \qquad (3) \\
&= 0.
\end{aligned}$$

Equation (2) follows from the fact that $t = x_0' - d$ is a function of d and x_0'. Equation (3) follows from the fact that $f(x) = p(x + t) + s(x)$, where S is uniformly random and independent of P. The last step results from the independence between the pre-distributed data D, G and the polynomial P.

(*Security for Bob*)
Let Bob be honest and $f(x) = p'(x+t)+s(x)$. Let $\text{VIEW}_A = \{S, T, F, P, P'\}$ be Alice's view of the protocol execution. Alice's output U will be a processed version of VIEW_A, consequently,

$$\begin{aligned}
I(X_0; U|P) &\leq I(X_0; \text{VIEW}_A|P) \\
&= I(X_0; STFP'|P) \\
&= I(X_0; STF|P) \qquad (4) \\
&= I(X_0; SF|P) \qquad (5) \\
&= 0.
\end{aligned}$$

Equation (4) follows from the fact that $p'(x) = f(x - t) - s(x - t)$, i.e., P' is fully determined by S, T and F. Equation (5) follows from the fact that $t = x_0 - d$, where D is uniformly random and independent of X_0. The last step results from the fact that $f(x) = p'(x + t) + s(x)$, where the data pre-distributed to Alice, S, is independent of Bob's input, X_0.

(*Optimality*)
Finally, our theorems 3, 4 and 5, show that indeed the size of the pre-distributed data as well as of the communicated data meet the lower bounds. □

5. Oblivious Linear Functional Evaluation

A linear functional l on a vector space $\mathcal{W}$ is defined as a function $l : \mathcal{W} \to \mathbb{R}$, which satisfies the following properties:

$$\begin{cases} l(\mathbf{v}+\mathbf{w}) &= l(\mathbf{v}) + l(\mathbf{w}). \\ l(\alpha\mathbf{w}) &= \alpha l(\mathbf{w}). \end{cases}$$

We generalize the previous OPE protocol to the case where Bob inputs $\mathbf{w} \in \mathcal{W}$ (vector space) and Alice inputs a linear functional $l \in \mathcal{W}^*$ (the dual vector space of linear functionals on $\mathcal{W}$). This task is called *Oblivious Linear Functional* (OLF) *Evaluation*. First, notice that evaluating a polynomial $p(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ on a point x_0 is the same as evaluating the linear functional $l = (a_0, a_1, \ldots, a_n)$ (as a row vector) on the (column) vector $\mathbf{w} = (1, x_0, x_0^2, \ldots, x_0^n)^T$. Thus OPE can be seen as a particular case of oblivious linear functional evaluation. This idea can be generalized to affine linear functionals, but we chose not to break the inherent beautiful symmetry via duality of the problem.

The choices of Alice and Bob are modeled by the random variables L and W, which can have arbitrary probability distributions. The security conditions are analogous to the ones for OPE. The ideal functionality f_{OLF} is denoted by

$$f_{\text{OLF}}(L, W) := (\perp, L(W)).$$

Next theorem formalizes the conditions for a secure implementation of OLF.

Theorem 7. *A protocol Π realizes an OLF perfectly securely if and only if for every admissible pair of algorithms $\overline{A} = (\overline{A}_1, \overline{A}_2)$ for protocol Π and for all inputs (L, W) and auxiliary input Z, $\overline{A}$ produce outputs (U, V) such that the following conditions are satisfied:*

- (Correctness) *If both players are honest, then*

$$(U, V) = (\perp, L(W)).$$

- (Security for Alice) *If Alice is honest, then we have $U = \perp$ and there exists a random variable W', such that*

$$I(L; W'|ZW) = 0, \text{ and } I(L; V|ZWW'L(W')) = 0.$$

- (Security for Bob) *If Bob is honest, then we have*

$$I(W;U|ZL) = 0.$$

Proof. We have to prove the equivalence between the privacy conditions for Bob in Theorems 1 and 7. This proof is analogous to the previous one.

According to Theorem 1, we must have that

$$I(W;L'|ZL) = 0 \text{ and } I(L(W)W;U|ZLL'U') = 0.$$

or equivalently,

$$I(W;L'|ZL) + I(L'(W)W;U|ZLL') = 0.$$

$L'(W)$ is a function of W and the linear functional L' so

$$\begin{aligned} I(L'(W)W;U|ZLL') &= I(W;U|ZLL') + I(L'(W);U|WZLL') \\ &= I(W;U|ZLL'). \end{aligned}$$

Then, the expression $I(L'(W)W;U|ZLL') = 0$ is equivalent to $I(W;U|ZLL') = 0$.

Applying the chain rule for mutual information we obtain

$$\begin{aligned} I(W;L'|ZL) + I(W;U|ZLL') &= I(W;L'U|ZL) \\ &= I(W;U|ZL) + I(W;L'|ZLU) \\ &= I(W;U|ZL). \end{aligned}$$

The last equality follows from the fact that, in a secure OLF implementation, L' and W have to be independent given ZLU. One can observe that, at the beginning of the protocol, Bob chooses his vector W independently from L', the linear functional actually supplied by Alice. Furthermore, we should require that the action of providing to the protocol a different input does not lead to any advantage to a malicious Alice. I.e., given that Alice knows ZLU, her knowledge on L' does not give her any additional information on Bob's input W. Mathematically, W, ZLU and L' will form a Markov Chain:

$$W \leftrightarrow ZLU \leftrightarrow L' \text{ and } \Pr(W|ZLU, W') = \Pr(W|ZLU) \Rightarrow I(W;L'|ZLU) = 0.$$

Therefore, the security condition for an honest Bob is reduced to:

$$I(W;U|ZL) = 0.$$

□

Next, we present our construction of oblivious equality testing. The intuition behind the protocol is similar to the one behind OPE. In the pre-distribution phase, Ted selects a random affine linear function and a random evaluation on the function and sends them to Alice and Bob, respectively. Subsequently, during the computing phase, Alice and Bob exchange information in order to obtain the desired result.

Protocol OLF

Setup Phase:

• Ted selects with uniform randomness an affine linear function m and a uniformly random $\mathbf{d} \in \mathcal{W}$.

• Ted transmits the function m to Alice and the point $\{\mathbf{d}\ ;\ c = m(\mathbf{d})\}$ to Bob.

Computing Phase:

$$\begin{cases} \text{Alice's input: } l \in \mathcal{W}^*. \\ \text{Bob's input: } \mathbf{w} \in \mathcal{W}. \end{cases}$$

• Bob sends $\mathbf{t} = \mathbf{w} - \mathbf{d}$ to Alice.

• Alice sends the function $n := l + m + l(\mathbf{t})$ to Bob.

• Bob computes $n(\mathbf{d}) - c = l(\mathbf{d}) + m(\mathbf{d}) + l(\mathbf{w} - \mathbf{d}) - m(\mathbf{d}) = l(\mathbf{w})$.

Theorem 8. *The above stated protocol is a secure implementation of an oblivious linear functional evaluation protocol.*

Proof.
(*Correctness*)
It is immediate to verify the correctness of the protocol. Considering both parties to be honest, we obtain

$$n(\mathbf{d}) - c = n(\mathbf{d}) - m(\mathbf{d}) = l(\mathbf{d}) + l(\mathbf{w} - \mathbf{d}) = l(\mathbf{w})$$

(*Security for Alice*)
Let Alice be honest and $\mathbf{w}' = \mathbf{d} + \mathbf{t}$. As a consequence,

$$I(L; W'|W) = I(L; D + T|W) = 0$$

since D is independent of L.

We shall now demonstrate that the second security condition for Alice also holds. Let $\textsc{View}_B = \{D, C, T, N, W, W', L(W')\}$ be Bob's view of the protocol execution. Bob's output V will be a processed version of $\textsc{View}_B$, consequently,

$$\begin{aligned} I(L; V|WW'L(W')) &\leq I(L; \textsc{View}_B|WW'L(W')) \\ &= I(L; DCTN|WW'L(W')) \\ &= I(L; DCT|WW'L(W')) \quad (6) \\ &= I(L; DC|WW'L(W')) \quad (7) \\ &= 0. \end{aligned}$$

Equation (6) follows from the fact that $n = l + m + l(\mathbf{t})$ is a function of $l, m,$ and$\mathbf{t}$, such that M is a random variable uniformly distributed and independent of L. Equation (7) follows from the fact that $\mathbf{t} = \mathbf{w} - \mathbf{d}$ is a function of $\mathbf{d}$ and $\mathbf{w}$. The last step is a consequence from the fact that the data pre-distributed to Bob, D, C is independent of Alice's input L.

(*Security for Bob*)
Let Bob be honest. Let $\textsc{View}_A = \{M, N, T, L, L'\}$. Alice's output U will be a processed version of her view, $\textsc{View}_A$, consequently,

$$\begin{aligned} I(W; U|L) &\leq I(W; \textsc{View}_A|L) \\ &= I(W; MNTL'|L) \\ &= I(W; MNT|L) \quad (8) \\ &= I(W; MN|L) \quad (9) \\ &= 0. \end{aligned}$$

Equation (8) follows from the fact that $l' + l'(\mathbf{t}) = m - n$, i.e., L' is fully determined by M, N, L and T. Equation (9) follows from the fact that $\mathbf{t} = \mathbf{w} - \mathbf{d}$, where D is uniformly random and independent of W. The last step results from the fact that $n = l' + m + l'(\mathbf{t})$, where the data pre-distributed to Alice, M, is independent of Bob's input, W. □

6. Applications

We present a solution for the Oblivious Equality Testing Problem as an illustration of the applicability of OPE.

6.1. Oblivious Equality Testing Problem

In the *Oblivious Equality Testing Problem*, Alice and Bob possess private inputs x_a and x_b, respectively. Bob wants to know if their inputs are equal in such a way that no unnecessary information is leaked in the process. I.e., at the end of the protocol, Bob will learn if $x_a = x_b$ and Alice will learn nothing ($\perp$).

The protocol is as follows:

Oblivious Equality Testing

Setup Phase:

• Ted generates with uniform randomness a linear function $f(x) = ax + b$ and a value x_0.

• Ted transmits the function $f(x)$ to Alice and the single evaluation point $\{x_0; f(x_0)\}$ to Bob.

Computing Phase:

$$\begin{cases} \text{Alice's input: } x_a \in \mathbb{F}_q. \\ \text{Bob's input: } x_b \in \mathbb{F}_q. \end{cases}$$

• Bob sends to Alice $t = x_b - x_0$

• Alice picks up a function $g(x)$ at random such that $g(x_a) = 0$.

• Alice computes the masked function $d(x)$, such that $d(x) = g(x+t) + f(x)$ and sends it to Bob.

• Bob checks if Alice sent the all zero function. If it is the case, he aborts. Otherwise, Bob computes $d(x_0) - f(x_0)$. If the result is zero Bob knows the inputs are the same, otherwise they are different.

$$\begin{cases} \text{If } x_b = x_a, \text{ then } d(x_0) - f(x_0) = 0 \\ \text{If } x_b \neq x_a, \text{ then } d(x_0) - f(x_0) \neq 0 \end{cases}$$

The solution to the oblivious equality testing problem consists of an execution of the OPE protocol, in which Alice's input is a polynomial of degree 1 with root x_a and Bob's input is his value x_b. Thus, it is straightforward to prove its security.

We have shown how to solve the equality testing problem by using only one OPE instantiation. Furthermore, our solution is secure against active adversaries and does not present any error probability.

In [3], Beimel and Malkin studied the problem of obtaining general secure function evaluation protocols against computationally unbounded passive adversaries when the parties are given black-box access to AND gates. They determined how many AND gates are necessary to securely compute a function f and provided solutions to the oblivious equality testing problem that achieved those bounds. Their results allow one to obtain lower bounds on the number of OTs required to compute the oblivious equality function. In summary, their results show that the amount of OTs required to compute the equality function is exponentially bigger in the perfect case (error-free) than in the statistical case (where a small probability of error is tolerated).

Relying on the work of [3, 16], we compare our solution for the oblivious equality testing problem with solutions based on other OT variants. As table 6.1 illustrates, our solution based on OPE is resistant against a more powerful adversary and is much more reliable and efficient.

Primitives	OPE	$\binom{4}{1}$-OT^1	$\binom{2}{1}$-OT^1
Number of instances	1	k	$3k$
Adversarial Model	active adversaries	passive adversaries	passive adversaries
Probability of Failure	0	2^{-k}	2^{-k}

Table 1: Comparison among solutions based on different primitives for the oblivious equality testing problem.

7. Conclusions

In this paper we introduced and solved the problem of efficiently evaluating polynomials obliviously within the so-called commodity-based cryptography, as proposed by Beaver [1]. We proposed a model and then proved bounds on the amount of "commodities" which have to be pre-distributed by the trusted center. Thus, we provided

bounds for the amount of memory required by the players engaged in the protocol, as well as bounds on their communications.

Then, we proved the tightness of our bounds by showing an explicit construction which meets them.

We also presented in this paper a definition of security for oblivious polynomial evaluation which is equivalent to the standard definition based on the *real/ideal model paradigm*. In the light of this new definition, we proved the unconditional security of our scheme.

Finally, we proposed a generalization of oblivious polynomial evaluation: oblivious linear functional evaluation and provided, as an application of our approach, an efficient solution to the oblivious equality testing problem.

References

[1] D. Beaver. Commodity-Based Cryptography (Extended Abstract). STOC 1997, pp. 446–455, 1997.

[2] D. Beaver, S. Micali, and P. Rogaway. The Round Complexity of Secure Protocols. STOC 1990, pp. 503–513, 1990.

[3] A. Beimel and T. Malkin. A Quantitative Approach to Reductions in Secure Computation. Theory of Cryptography Conference 2004, LNCS, vol. 2951, Springer-Verlag, pp. 238–257, 2004.

[4] D. Bleichenbacher and P. Nguyen. Noisy Polynomial Interpolation and Noisy Chinese Remaindering. EUROCRYPT 2000, LNCS, vol. 1807, Springer-Verlag, pp. 53–69, 2000.

[5] C. Blundo, B. Masucci, D.R. Stinson, and R. Wei. Constructions and Bounds for Unconditionally Secure Non-Interactive Commitment Schemes. Designs, Codes, and Cryptography, 26(1-3), pp. 97–110, 2002.

[6] Yan-Cheng Chang, Chi-Jen Lu. Oblivious Polynomial Evaluation and Oblivious Neural Learning. ASIACRYPT 2001, LNCS, vol. 2248, Springer-Verlag, pp. 369–384, 2001.

[7] C. Crépeau, G. Savvides, G. Schaffner, J. Wullschleger. Information-Theoretic Conditions for Two-Party Secure Function Evaluation. EUROCRYPT 2006, LNCS, 4004, Springer-Verlag, pp. 528–554, 2006.

[8] C. Crépeau, J. Wullschleger. Statistical Security Conditions for Two-Party Secure Function Evaluation. ICITS 2008, LNCS, vol. 5155, Springer-Verlag, pp. 86–99, 2008.

[9] O. Goldreich. Foundations of Cryptography, volume II: Basic Applications. Cambridge University Press, 2004.

[10] G. Hanaoka, J. Shikata, Y. Zheng, and H. Imai. Unconditionally Secure Digital Signature Schemes Admitting Transferability. ASIACRYPT 2000, LNCS, vol. 1976, Springer-Verlag, pp. 130–142, 2000.

[11] J. Kilian. Founding Cryptography on Oblivious Transfer. STOC 1988, pp. 20–31, 1988.

[12] T. Matsumoto and H. Imai. On the Key Predistribution Systems. A Practical Solution to the Key Distribution Problem. CRYPTO 1987, LNCS, vol. 293, Springer-Verlag, pp. 185–193, 1988.

[13] M. Naor and B. Pinkas. Oblivious Transfer and Polynomial Evaluation. STOC 1999, pp. 245–254, 1999.

[14] M. O. Rabin. How to Exchange Secrets by Oblivious Transfer. Technical Report TR-81, Harvard, 1981.

[15] R. Rivest. Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Concealing Channels and a Trusted Initializer. Preprint available at http://people.csail.mit.edu/rivest/Rivest-commitment.pdf.

[16] S. Winkler and J. Wullschleger. On the Efficiency of Classical and Quantum Oblivious Transfer. CRYPTO 2010, LNCS, vol. 6223, Springer-Verlag, pp. 707 – 723, 2010.

[17] S. Wolf and J. Wullschleger. Oblivious Transfer is Symmetric. EUROCRYPT 2006, LNCS, vol. 4004, Springer-Verlag, pp. 222-232, 2006.

[18] A.C. Yao. Protocols for Secure Computations. FOCS 1982, pp. 160–164, 1982.