New Results on Unconditionally Secure Distributed Oblivious Transfer

(extended abstract)

Carlo Blundo[1], Paolo D'Arco[2], Alfredo De Santis[1], and Douglas R. Stinson[3]

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy
{carblu, ads}@dia.unisa.it
[2] Department of Combinatorics and Optimization
University of Waterloo, Waterloo Ontario, N2L 3G1, Canada
pdarco@cacr.math.uwaterloo.ca
[3] School of Computer Science
University of Waterloo, Waterloo Ontario, N2L 3G1, Canada
dstinson@cacr.math.uwaterloo.ca

Abstract. This paper is about the Oblivious Transfer in the distributed model recently proposed by M. Naor and B. Pinkas. In this setting a Sender has n secrets and a Receiver is interested in one of them. During a set up phase, the Sender gives information about the secrets to m servers. Afterwards, in a recovering phase, the receiver can compute the secret she wishes by interacting with k of them. More precisely, from the answers received she computes the secret in which she is interested but she gets no information on the others and, at the same time, any coalition of $k-1$ servers can neither compute any secret nor figure out which one the receiver has recovered.
We present an analysis and new results holding for this model: lower bounds on the resources required to implement such a scheme (i.e., randomness, memory storage, communication complexity); some impossibility results for one-round distributed oblivious transfer protocols; two polynomial-based constructions implementing 1-out-of-n distributed oblivious transfer, which generalize the two constructions for 1-out-of-2 given by Naor and Pinkas; as well as new one-round and two-round distributed oblivious transfer protocols, both for threshold and general access structures on the set of servers, which are optimal with respect to some of the given bounds. Most of these constructions are basically combinatorial in nature.

1 Introduction

Introduced by Rabin in [27], and subsequently defined in different forms [18, 8], the *oblivious transfer* (OT, for short) has found many applications in cryptographic studies and protocol design. Basically, such a protocol enables one party to transfer knowledge to another in an "oblivious" way. Rabin's definition, for example, enables a Sender to transmit a message to a Receiver in such a way that the Receiver with probability $\frac{1}{2}$ gets the message while, with the same probability, she does not, and the Sender does not know which situation has happened. Rabin showed how this transfer can be used in order to exchange secrets, and subsequently several other researchers have shown some useful applications of this concept. The protocol proposed by Rabin was later strengthened in [19].

The second OT definition was given in [18]. In this form, the Sender has two secrets and the Receiver is interested in one of them. After the execution of the protocol, the receiver gets the secret she wishes to recover, obtaining at the same time no information on the other, while the Sender does not know which secret the receiver has recovered. The author of [18] showed a first application to signing contracts.

The last and more general form of OT was introduced in [8], under the name of *all-or-nothing Disclosure of Secrets*, even if the same concept was born in an artificial intelligence context [33], under the name of *multiplexing*. Here the Sender has n secrets and the Receiver is interested in one of them. After the execution of the protocol, the receiver gets the secret she wishes to recover, obtaining at the same time no information on the others, while the Sender does not know which secret the receiver has recovered.

All these forms were shown to be equivalent [9, 7, 13], and Kilian in [24] showed that the OT is a complete primitive, in the sense that it can be used as building block for any secure function evaluation (multi-party computation).

A variety of slightly different definitions and implementations can be found in the literature as well as papers addressing issues such as the relation of the OT with other cryptographic primitives, the assumptions required to implement such a concept, reductions among "more complex" forms of OT to "simpler ones" and applicative environments (e.g., [13, 9, 17, 16, 3, 14, 23, 26, 22], just to name few examples).

Our Contribution. In this paper we study *unconditionally secure distributed oblivious transfer protocols*, introduced in [25] in order to strengthen the security of protocols designed for electronic auctions [26]. We present an analysis and some new results: lower bounds on the resources required by an implementation such as randomness, memory storage, and communication complexity; some impossibility results for one-round protocols; two polynomial-based constructions implementing 1-out-of-n distributed oblivious transfer which generalize the two constructions for 1-out-of-2 schemes given by M. Naor and B. Pinkas; as well as new one-round and two-round distributed oblivious transfer protocols, both for threshold and general access structures on the set of servers, which are optimal with respect to some of the given bounds. Most of these constructions are basically combinatorial in nature.

Related Work. In the literature there are many papers that address problems related to 1-out-of-n distributed oblivious transfer. In [1], for example, the authors show how to distribute a function between several servers, in such a way that a user can compute the function by interacting with the servers; the servers cannot find out which values of the function the user computes, but the user can compute the function in *more than* one point. Another very close area is represented by PIR (Private Information Retrieval) Schemes, introduced in [11]. A PIR scheme enables a user to retrieve an item of information from a public accessible database in such a way that the database manager cannot figure out from the query which item the user is interested in. However, the user can get information about more than one item. On the other hand, in SPIR (Symmetric Private Information Retrieval) schemes [20], the user can get information about *one and only one* item, i.e. even the privacy of the database is considered. In PIR and SPIR schemes, the emphasis is placed on the *communication complexity* of the interaction of user and servers. Notice that a SPIR Scheme can be seen as a *communication-efficient* 1-out-of-n oblivious transfer scheme and the protocols given in [20] represent the first 1-round distributed implementation of 1-out-of-n oblivious transfer. However, the main differences between the model we are going to consider and (information theoretic) SPIR schemes are that in SPIR schemes the receiver communicates with k out of k servers in order to retrieve an item while in our setting the receiver can choose k out of m servers, where $k \leq m$. Moreover, in SPIR schemes, the security of the sender against *coalitions* of receiver and servers is not of concern. Other PIR papers of interest, for the distribute OT scenario we consider, are [2, 21, 15].

Finally, Rivest's model in [28], where a trusted initializer participates *only* during the set up phase of the system (see also [6]), provides a very close setting to the one described in [25] and considered in this paper. A very recent paper which deals with distributed oblivious transfer implementations, close to the setting introduced in [25] (but not unconditionally secure) is [32].

In our constructions we use secret sharing schemes. Secret sharing were introduced in 1979 by Blakley [4] and Shamir [29], and have been extensively studied during the last years. The reader can find an introduction in [31] and references to the literature in [30].

2 The Distributed Model

Let us define the model we are going to consider. We assume that the Sender holds n secrets and the receiver is interested in one of them. Hence, we are concerned with a 1-out-of-n distributed oblivious transfer.

2.1 An Informal Description

In the distributed setting, the sender $\mathcal{S}$ does not directly interact with the receiver $\mathcal{R}$ in order to carry out the oblivious transfer. Rather, he *delegates* m servers to accomplish this task for him. More precisely, we consider the following scenario:

Initialization Phase. Let $\mathcal{S}_1, \ldots, \mathcal{S}_m$ be m servers. The sender $\mathcal{S}$ generates m programs $P_1, \ldots, P_m$, and, for $i = 1, \ldots, m$ sends, *in a secure way*, program P_i to server $\mathcal{S}_i$.

Oblivious Transfer Phase. The receiver $\mathcal{R}$ holds a program which enables her to interact with a subset $\{\mathcal{S}_{i_1}, \ldots, \mathcal{S}_{i_k}\}$ of the servers at her choice. Using the knowledge acquired by exchanging messages with the servers, $\mathcal{R}$ recovers the secret in which she is interested, but receives no information on the other secrets. At the same time, no subset of $k-1$ servers, gains any information about the secret she has recovered. More precisely, a distributed (k, m)-DOT-$\binom{n}{1}$ must guarantee:

1. **Reconstruction.** If the receiver gets information from k out of the m servers, she can compute the secret.
2. **Sender's Privacy.** Given any k values, the receiver must gain information about a single secret, and no information about the others.
3. **Receiver's Privacy.** No coalition of less than k servers gains information about which secret the receiver has recovered.
4. **Receiver-servers Collusion.** A coalition of the receiver with $k-1$ corrupt servers cannot learn about the n secrets more than can be learned by the receiver herself.

Notice that, in [25], properties 3. and 4. are only guaranteed with respect to a threshold t and a threshold ℓ, respectively, which should be as close to k as possible.

2.2 A Formal Model

Assume that $\mathcal{S}$ holds a program S to generate m programs $P_1, \ldots, P_m$ enabling $\mathcal{S}_1, \ldots, \mathcal{S}_m$ and $\mathcal{R}$ to perform a (k, m)-DOT-$\binom{n}{1}$ oblivious transfer of his n secrets. $\mathcal{R}$ holds an associated program R for interacting with the servers. The $m+1$ programs $P_1, \ldots, P_m$ and R, specify[1] the computations to be performed to achieve (k, m)-DOT-$\binom{n}{1}$. In order to model dishonest behaviors, where a coalition of at most $k-1$ servers tries to figure out which secret $\mathcal{R}$ has recovered from the transfer, we assume that cheating servers $\mathcal{S}_{i_1}, \ldots, \mathcal{S}_{i_{k-1}}$ hold a modified version of the programs, denoted by $\overline{P_{i_1}}, \ldots, \overline{P_{i_{k-1}}}$. These programs could have been generated either by a dishonest $\mathcal{S}$, who holds a cheating generating program $\overline{S}$, or they could have been modified by the dishonest servers. Similarly, a cheating $\mathcal{R}$, who tries to gain some information about other secrets, holds a modified version of the program $\overline{R}$. These programs can be described by random variables and will be represented in bold face type.

An execution of the protocol can be described by using the following additional random variables: for $j = 1, \ldots, m$, let $\mathbf{C}_j$ be the transcript of the communication between $\mathcal{R}$ and S_j. Let W be the set of all length n sequences of secrets, and, for any $w \in W$, let w_i be the i-th secret of the sequence. Denoting by $\mathbf{W}$ the random variable that represents the choice of an element in W, and by $\mathbf{T}$ the random variable representing the choice of an index i in $T = \{1, \ldots, n\}$,

[1] If we are interested in a reduction of a more complex form of DOT to a simpler available one, we can simply assume that these programs encapsulate, as *black box*, a smaller (k, m)-DOT-$\binom{n'}{1}$. Hence, during the execution, $\mathcal{S}_1, \ldots, \mathcal{S}_m$ and $\mathcal{R}$ are able to carry out many times unconditionally secure (k, m)-DOT-$\binom{n'}{1}$.

we can define the conditions that a (k,m)-DOT-$\binom{n}{1}$ oblivious transfer protocol must satisfy as follows[2]:

Definition 1. *The sequence of programs $[S, P_1, \ldots P_m, R]$ is* correct *for (k,m)-DOT-$\binom{n}{1}$ if for any $i \in T$ and $j = 1, \ldots, m$, it holds that*

$$H(\mathbf{C}_j | \mathbf{P}_j\, \mathbf{T}\, \mathbf{R}) = 0, \tag{1}$$

and, for any $w \in W$ and for any $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, m,\}$ it holds that

$$H(\mathbf{W}_T | \mathbf{C}_{i_1} \ldots \mathbf{C}_{i_k}) = 0. \tag{2}$$

Notice that the definition means that, given the program of server S_j and the program of the receiver and her choices, the transcript of the communication is completely determined. Moreover, after interacting with k servers, an honest receiver always recovers the secret in which she is interested.

Assuming that both $\mathcal{S}$ and $\mathcal{R}$ are aware of the joint probability distribution $\mathcal{P}_{W,T}$ on W and T, the probability with which $\mathcal{S}$ chooses the secrets in W and $\mathcal{R}$ chooses an index $i \in T$, the privacy property of (k,m)-DOT-$\binom{n}{1}$ can be defined as follows:

Definition 2. *The sequence of programs $[S, P_1, \ldots P_m, R]$ is* private *for (k,m)-DOT-$\binom{n}{1}$ if*

- *for any set of indices $\{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, m\}$, it holds that*

$$H(\mathbf{T} | \overline{\mathbf{P}}_{i_1}, \ldots, \overline{\mathbf{P}}_{i_{k-1}} \mathbf{C}_{i_1}, \ldots, \mathbf{C}_{i_{k-1}}) = H(\mathbf{T}). \tag{3}$$

- *for any program $\overline{R}$, for any $i \in T$ and for any set of indices $\{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$, it holds that*

$$H(\mathbf{W} \setminus \mathbf{W}_T \,|\, \mathbf{T}\, \overline{\mathbf{R}}\, \mathbf{C}_{i_1} \ldots \mathbf{C}_{i_k} \mathbf{W}_T\,) = H(\mathbf{W} \setminus \mathbf{W}_T). \tag{4}$$

- *for any set of indices $\{i_1, \ldots, i_{k-1}\} \subset \{1, \ldots, m\}$, for any $i \in T$, and for any $\overline{R}$, it holds that*

$$H(\mathbf{W} | \mathbf{T}\, \overline{\mathbf{R}}\, \mathbf{C}_{i_1} \ldots \mathbf{C}_{i_{k-1}}\, \overline{\mathbf{P}}_{i_1}, \ldots, \overline{\mathbf{P}}_{i_{k-1}}) = H(\mathbf{W}). \tag{5}$$

- *for any sets of indices $\{i_1, \ldots, i_{k-1}\} \subseteq \{1, \ldots, m\}$ and $\{j_1, \ldots, j_k\} \subseteq \{1, \ldots, m\}$, for any $i \in T$, and for any $\overline{R}$, it holds that*

$$H(\mathbf{W} \setminus \mathbf{W}_T |\, \mathbf{T}\, \overline{\mathbf{R}}\, \overline{\mathbf{P}}_{i_1}, \ldots, \overline{\mathbf{P}}_{i_{k-1}}\, \mathbf{C}_{j_1} \ldots \mathbf{C}_{j_k} \mathbf{W}_T) = H(\mathbf{W} \setminus \mathbf{W}_T). \tag{6}$$

Conditions (3) and (4) ensure that a dishonest coalition of servers does not gain information about $\mathcal{R}$'s index; and a dishonest $\mathcal{R}$ infers at most one secret among the ones held by $S_1, \ldots, S_m$. Condition (5) takes into account the possibility of an attack against $\mathcal{S}$ performed either by at most $k-1$ servers alone or with the cooperation of $\mathcal{R}$. The condition states that such coalitions do not gain

[2] Since we focus our attention on unconditionally secure DOT protocols, we use the entropy function, which leads to a compact and concise description. The reader is referred to the Appendix for a short introduction to entropy and information theory.

any information about the secrets held by $\mathcal{S}$. Finally, condition (6) states that a coalition of $k-1$ servers and the receiver, once the receiver has obtained a secret, cannot compute any information about the other secrets. In the following, we will show that this condition cannot be achieved if the DOT protocol provides only one round of interaction. On the other hand, with two rounds of interaction, this level of security can be obtained. Notice that, in our model, conditions (4) and (6) are not independent: indeed, (6) implies (4). To simplify the description and the analysis of the security we have chosen to state two different conditions.

3 Lower Bounds and Impossibility Results

Using some information theory tools, we can prove bounds on the memory storage, on the communication complexity and on the randomness needed by a DOT scheme. Moreover, we can show that with one-round protocols condition (6) of the DOT definition cannot be achieved. Actually, we can prove that *a single* server can help the receiver to recover all the secrets, once the receiver has legally retrieved the first one. Due to this result, we will refer to schemes achieving all but condition (6), as to *weak* DOT schemes.

The following bounds (see Table 1) hold for both weak DOT schemes and for DOT schemes, since condition (6) is not used in the proofs that, due to lack of space, will appear in the full version of the paper.

Theorem 1. *(***Memory Storage***.) In any (k,m)-DOT-$\binom{n}{1}$ scheme for each $j = 1,\ldots,m$, it holds that*

$$H(\mathbf{P}_j) \geq H(\mathbf{W}).$$

Theorem 2. *(***Randomness to Set Up the Scheme***.) In any (k,m)-DOT-$\binom{n}{1}$ scheme, it holds that*

$$H(\mathbf{P}_1 \ldots \mathbf{P}_m) \geq kH(\mathbf{W}).$$

Theorem 3. *(***Complexity of each Interaction***.) In any (k,m)-DOT-$\binom{n}{1}$ scheme, for each $j = 1,\ldots,m$, it holds that*

$$H(\mathbf{C}_j) \geq H(\mathbf{W}_T).$$

Theorem 4. *(***Randomness of the whole Communication***.) In any (k,m)-DOT-$\binom{n}{1}$ scheme, for any $1 \leq i_1 < \ldots, < i_k \leq n$, it holds that*

$$H(\mathbf{C}_{i_1} \ldots \mathbf{C}_{i_k}) \geq kH(\mathbf{W}_T).$$

Table 1. Bounds holding on the Model

Notice that if the protocol is one-round, then $C_j = (Q_j, A_j)$, the query of the receiver and the answer of the server. Therefore, condition (1) can be re-phrased saying that for $j = 1, \ldots, m$

$$H(\mathbf{Q}_j|\mathbf{R}\,\mathbf{T}) = 0 \quad \text{and} \quad H(\mathbf{A}_j|\mathbf{Q}_j\mathbf{P}_j) = 0. \tag{7}$$

With this notation, we can prove the following impossibility result:

Theorem 5. *In any one-round scheme for (k,k)-DOT-$\binom{n}{1}$, once the receiver has legally recovered a secret, a single corrupt server and the receiver can recover all the others.*

Proof. Let $q_1, \ldots, q_k$ be the queries sent by the receiver when $T = i$, and let $a_1, \ldots, a_k$ be the answers that $S_1 \ldots, S_k$ send back to the receiver. The Receiver's security property (3) with respect to $k-1$ servers, say $S_2, \ldots, S_k$, implies that there exist queries q_1^s and answers a_1^s, for any $s \neq i$, such that if

$$H(\mathbf{W}_i|\mathbf{Q}_1 = q_1\, \mathbf{Q}_2 = q_2 \ldots \mathbf{Q}_k = q_k, \mathbf{A}_1 = a_1\, \mathbf{A}_2 = a_2 \ldots \mathbf{A}_k = a_k) = 0$$

then

$$H(\mathbf{W}_s|\mathbf{Q}_1 = q_1^s\, \mathbf{Q}_2 = q_2 \ldots \mathbf{Q}_k = q_k, \mathbf{A}_1 = a_1^s\, \mathbf{A}_2 = a_2 \ldots \mathbf{A}_k = a_k) = 0$$

Since the answer given by S_1 depends only on his own program P_1 and on the received query (i.e., $H(\mathbf{A}_1|\mathbf{Q}_1\mathbf{P}_1) = 0$), it holds that

$$H(\mathbf{W}|\mathbf{P}_1\mathbf{A}_2 \ldots \mathbf{A}_k, \mathbf{Q}_2 \ldots \mathbf{Q}_k\mathbf{R}) = 0.$$

Indeed

$$H(\mathbf{W}|\mathbf{P}_1\mathbf{A}_2 \ldots \mathbf{A}_k, \mathbf{Q}_2 \ldots \mathbf{Q}_k\mathbf{R}) \leq \sum_{t \in T} H(\mathbf{W}_t|\mathbf{P}_1\mathbf{A}_2 \ldots \mathbf{A}_k, \mathbf{Q}_2 \ldots \mathbf{Q}_k\mathbf{R}, \mathbf{T} = t)$$

and

$$\begin{aligned} H(\mathbf{W}_t|\mathbf{P}_1\mathbf{A}_2 \ldots \mathbf{A}_k\mathbf{Q}_2 \ldots \mathbf{Q}_k\mathbf{R}, \mathbf{T} = t) &\leq H(\mathbf{W}_t|\mathbf{P}_1\mathbf{A}_2 \ldots \mathbf{A}_k\mathbf{Q}_1\mathbf{Q}_2 \ldots \mathbf{Q}_k) \\ &\leq H(\mathbf{W}_t|\mathbf{A}_1\mathbf{A}_2 \ldots \mathbf{A}_k\mathbf{Q}_1\mathbf{Q}_2 \ldots \mathbf{Q}_k) = 0 \end{aligned}$$

Therefore, the receiver and S_1 can recover all the secrets and the result holds[3].
□

A consequence of this impossibility result for one-round protocols is that the highest security level aimed in [26] with this approach cannot be achieved.

Notice that the model is quite general. If we consider one-round weak DOT schemes such that a sequence of k queries determines $\mathbf{T}$ *uniquely*, i.e.,

[3] Notice that the result can easily be extended to general (k,m)-DOT-$\binom{n}{1}$ and to $\mathcal{A}$-DOT-$\binom{n}{1}$ schemes for general access structures on the set of servers. Moreover, applying the same argument, it is possible to show that, if the receiver's security property (3) must hold, in the threshold case, against a coalition *of size at most t*, then the receiver, after having legally recovered one secret, can recover *all* the others if she colludes with $k-t$ servers.

$$H(\mathbf{T}|\mathbf{Q}_{i_1} \dots \mathbf{Q}_{i_k}) = 0, \tag{8}$$

then we can show a bound on the randomness of the receiver. All the known constructions and the new ones we introduce enjoy this property, which in general is not guaranteed or required by the conditions defining our model.

Theorem 6. *In any one-round weak (k,m)-DOT-$\binom{n}{1}$ scheme satisfying (8) it holds that*

$$H(\mathbf{R}) \geq (k-1)H(\mathbf{T}).$$

4 Protocols Implementing weak (k, m)-DOT-$\binom{n}{1}$

Two protocols for weak (k,m)-DOT-$\binom{2}{1}$ have been proposed in [25]. Their general structure is given in Table 2.

A General Protocol for a weak (k,m)-DOT-$\binom{2}{1}$ Implementation. Let $w_0, w_1 \in GF(q)$ be $\mathcal{S}$'s secrets, and let $\sigma \in \{0,1\}$ be $\mathcal{R}$'s index. - $\mathcal{S}$ generates a bivariate polynomial $Q(x,y)$ with values in $GF(q)$ such that $Q(0,0) = w_0$ and $Q(0,1) = w_1$. - Then, for $i = 1,\dots,m$, he sends the univariate polynomial $Q(i,\cdot)$ to the server $\mathcal{S}_i$. - $\mathcal{R}$ chooses a random polynomial Z such that $Z(0) = \sigma$, and defines a univariate polynomial $V(x) = Q(x, Z(x))$. The degree of V is $k-1$. - Then, she asks server $\mathcal{S}_i$ for the value $V(i) = Q(i, Z(i))$. - After receiving k values of V, $\mathcal{R}$ interpolates V and computes $V(0)$.

Table 2. A General Protocol for a weak (k,m)-DOT-$\binom{2}{1}$

The first protocol uses a *sparse* bivariate polynomial. The second one uses a *full* bivariate polynomial and is secure against coalitions between $\mathcal{R}$ and several servers (under the weaker condition (3)). The constructions can be transformed in an *unconditionally secure* form, by replicating the basic scheme given in Table 2, and using some ad hoc coefficients [25]. Moreover, we can use weak (k,m)-DOT-$\binom{2}{1}$ as a black box to construct "more complex" forms of oblivious transfer in the same distributed model (see [16] for some unconditionally secure reductions). In this situation, any improvement in the design of the available weak (k,m)-DOT-$\binom{2}{1}$, yields an improvement of the performance of the more complex protocols.

In this section, we propose a protocol, based on polynomial interpolation, implementing weak (k,m)-DOT-$\binom{n}{1}$ oblivious transfer. This protocol is a generalization of the weak (k,m)-DOT-$\binom{2}{1}$ protocol proposed in [25]. The protocol is described in Table 3.

First Protocol for a weak (k,m)-DOT-$\binom{n}{1}$. Let $s_0, s_1, \ldots, s_{n-1} \in GF(q)$ be $\mathcal{S}$'s secrets, and let $i \in \{0, \ldots, n-1\}$ be $\mathcal{R}$'s index. - $\mathcal{S}$ generates an n-variate polynomial $Q(x, y_1, \ldots, y_{n-1})$ with values in $GF(q)$ such that $Q(0,0,\ldots,0) = s_0, Q(0,1,0,\ldots,0) = s_1, \ldots, Q(0,0,\ldots,1) = s_{n-1}$. More precisely, $$Q(x, y_1, \ldots, y_{n-1}) = \sum_{j=1}^{k-1} a_j x^j + b_0 + b_1 y_1 + \cdots, b_{n-1} y_{n-1},$$ where $s_0 = b_0$ and, for $i = 1, \ldots, n-1$, $s_i = b_0 + b_i$. - Then, for $i = 1, \ldots, m$, he sends the $n-1$-variate polynomial $Q(i, y_1, \ldots, y_{n-1})$ to the server $\mathcal{S}_i$. - $\mathcal{R}$ chooses $n-1$ random polynomials $Z_{y_1}(x) \ldots, Z_{y_{n-1}}(x)$ of degree $k-1$ such that $(Z_{y_1}(0) \ldots, Z_{y_{n-1}}(0))$ is an $(n-1)$-tuple of zeroes having at most one 1 in position i, the position corresponding to the secret in which she is interested, and defines a univariate polynomial V to be $V(x) = Q(x, Z_{y_1}(x), \ldots, Z_{y_{n-1}}(x))$. The degree of V is $k-1$. - Then, she asks server $\mathcal{S}_{i_j}$ for the value $V(i_j) = Q(i_j, Z_{y_1}(i_j), \ldots, Z_{y_{n-1}}(i_j))$. - After receiving k values of V, say $V(i_1), \ldots V(i_k)$, $\mathcal{R}$ interpolates V and computes $V(0)$.

Table 3. First Protocol for weak (k,m)-DOT-$\binom{n}{1}$.

Correctness. Let $Z_{y_i}(x) = s^0_{y_i} + \sum_{j=1}^{k-1} s^j_{y_i} x^j$ be the polynomials generated by $\mathcal{R}$, random up to $s^0_{y_i}$. The polynomial $V(x)$ interpolated by $\mathcal{R}$,

$$V(x) = Q(x, Z_{y_1}(x), \ldots, Z_{y_{n-1}}(x)),$$

can be written in explicit form as

$$\sum_{j=1}^{k-1} a_j x^j + b_0 + b_1 (s^0_{y_1} + \sum_{j=1}^{k-1} s^j_{y_1} x^j) + \cdots, b_{n-1} (s^0_{y_{n-1}} + \sum_{j=1}^{k-1} s^j_{y_{n-1}} x^j),$$

which can be re-arranged as

$$\sum_{j=1}^{k-1}(a_j + b_1 s^j_{y_1} + \cdots b_{n-1} s^j_{y_{n-1}})x^j + b_0 + b_1 s^0_{y_1} + \cdots + b_{n-1} s^0_{y_{n-1}}.$$

For $x = 0$, and assuming that the $n-1$-tuple $(s^0_{y_1}, \ldots, s^0_{y_{n-1}}) = (0, \ldots, 1, \ldots, 0)$ (i.e., having at most one 1 in position i, where $i \in \{1, \ldots, n-1\}$), then $V(0) = b_i + b_0 = s_i$.

□

It is possible to show that, in the above form, the receiver can learn *a single* linear combination of the secrets, extending the proof given by Naor and Pinkas [25] for two secrets. Moreover, along the same line as [25], the protocol can be used as a building block to set up an unconditionally secure weak DOT that meets the bounds given by Theorems 1 and 2. Indeed, these bounds still hold for weak DOT schemes, since they are obtained without making any use of condition (6). As well as, we can use full n-variate polynomials to set up a protocol which is secure against t servers and a coalition among the receiver and $\ell \leq k-t$ servers. Details will be given in the full version of the paper.

5 Combinatorial Constructions

In this section we propose some combinatorial constructions for distributed oblivious transfer. Some of these constructions require trivial computations once the scheme has been set up by the Sender, and the one-round protocols meet the lower bound on the number of random bits the receiver must use to set up the queries, given by Theorem 6. However, they are not so efficient in terms of memory server storage and communication complexity.

5.1 One-Round Constructions

We start by giving protocols which require one round of interaction to recover a secret. The constructions are based on well-known combinatorial structures used in secret sharing. The first protocol is given in Table 4:

Example. For 3 secrets and 2 servers, suppose that we use the protocol described in Table 4.

Let $A[2, 3^2]$ be the following matrix with values in $GF(q)$:

$$\begin{bmatrix} a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} & a_{1,6} & a_{1,7} & a_{1,8} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} & a_{2,6} & a_{2,7} & a_{2,8} \end{bmatrix}$$

The representations of $\{0, 1, \ldots, 8\}$ in base 3 are:

0	1	2	3	4	5	6	7	8
00	01	02	10	11	12	20	21	22

A Weak (k,k)-DOT-$\binom{n}{1}$.
Let $s_0, s_1, \ldots, s_{n-1} \in GF(q)$ be $\mathcal{S}$'s secrets, and let $A[p,j]$ be a $k \times n^k$ matrix of values in $GF(q)$ such that, for $j \in \{0, \ldots, n^k - 1\}$, the sum of the values of column $A[j]$ is s_i if, assuming that $c_1^j \cdots c_k^j$ is the representation in base n of j, the sum $\sum_{\ell=1}^{k} c_\ell^j \bmod n = i$.

- $\mathcal{S}$ sends the p-th row $A[p]$ of $A[p,j]$ to the server S_p
- The receiver chooses a value $j \in \{0, \ldots, n^k - 1\}$ such that the $\sum_{\ell=1}^{k} c_\ell^j \bmod n = i$, where i is the index of the secret she wishes to recover. Then, for $p = 1, \ldots, k$ she sends the digit c_p^j to server S_p.
- Server S_p sends to the receiver, for $q = 0, \ldots, n^k - 1$, the value $A[p,q]$ if and only if the p-th digit of the n-ary representation of q is equal to c_p^j.
- The receiver sums up the values $A[1,j], \ldots, A[k,j]$, recovering the secret.

Table 4. A Weak (k,k)-DOT-$\binom{n}{1}$ Construction

Columns $0, 5$ and 7 encrypt s_0, $1, 3$ and 8 encrypt s_1, and $2, 4$ and 6 encrypt s_2.

To recover s_1, the receiver chooses, for example, column 8, and sends $c_1 = 2$ to server 1 and $c_2 = 2$ to server 2, receiving the values $a_{1,6}, a_{1,7}, a_{1,8}$ and $a_{2,2}, a_{2,5}, a_{2,8}$. □

Using some well-known combinatorial structures, we can generalize the above construction, in order to set up a weak (k,m)-DOT-$\binom{n}{1}$. More precisely, let t be an integer such that $1 \leq t \leq q$ and $r \geq 2$. An *orthogonal array* $OA_\lambda(t,q,r)$ is a $\lambda r^t \times q$ array A of r symbols, such that within any t columns of A, every possible t-tuple of symbols occurs in exactly λ rows of A. Using an orthogonal array and a collection of secret sharing schemes we can set up a weak (k,m)-DOT-$\binom{n}{1}$ (see Table 5).

Example. We present a weak $(2,3)$-DOT-$\binom{3}{1}$ using the protocol described in Table 5.
Let us consider the following $OA_1(2,4,3)$:

$$A[4,9] = \begin{bmatrix} 0&0&0&1&1&1&2&2&2 \\ 0&1&2&0&1&2&0&1&2 \\ 0&1&2&2&2&0&2&0&1 \\ 0&1&2&2&0&1&1&2&0 \end{bmatrix}$$

Suppose that $\mathcal{R}$ wishes to recover the secret s_0. Hence, she chooses for example the first column, $c = 0$, she chooses a subset of 2 servers, say S_2 and S_3, and she sends 0 to S_2 and 0 to S_3. The contacted servers reply by sending the following values

- S_2 sends $(0, sh_{2,0}), (5, sh_{2,5})$ and $(7, sh_{2,7})$
- S_3 sends $(0, sh_{3,0}), (4, sh_{3,4})$ and $(8, sh_{3,8})$.

A Weak (k, m)**-DOT-**$\binom{n}{1}$

Let $s_0, s_1, \ldots, s_{n-1} \in GF(q)$ be $\mathcal{S}$'s secrets, and let $A[m+1, n^k]$ be an orthogonal array $OA_1(k, m+1, n)$. The first row $A[0]$ of the (public) orthogonal array $A[m+1, n^k]$ establishes "which column encrypts which secret". More precisely, we have the following:

Set up Phase.

- The Sender $\mathcal{S}$, for each $0 \leq c \leq n^k - 1$, shares $s_{A[0,c]}$ according to a (k, m)-threshold scheme. Let us denote such sharing by $sh_{1,c}, \ldots, sh_{m,c}$.
- Then, for $i = 1, \ldots, m$, $\mathcal{S}$ sends $sh_{i,0}, \ldots, sh_{i,n^k-1}$ to server S_i.

Suppose that $\mathcal{R}$ wishes to reconstruct the secret s_ℓ, for some $\ell \in \{0, \ldots, n-1\}$.

Recovering Phase.

- $\mathcal{R}$ chooses a random column c of the matrix A such that $A[0, c] = \ell$, picks a k-subset of $\{1, \ldots, m\}$, say $p_1, \ldots, p_k$, and, for $1 \leq j \leq k$, sends the value $y_j = A[p_j, c]$ to server S_{p_j}.
- For $1 \leq j \leq k$, server S_{p_j} sends $(d, sh_{p_j,d})$ to the receiver $\mathcal{R}$, for all d such that $A[p_j, d] = y_j$. $\mathcal{R}$ gets n shares from each of the k servers.
- Finally, $\mathcal{R}$ uses $sh_{p_1,c}, \ldots, sh_{p_k,c}$ to reconstruct the secret s_ℓ.

Table 5. A Weak (k, m)-DOT-$\binom{n}{1}$ Construction

Therefore, the receiver can recover s_0 using $(0, sh_{2,0})$ and $(0, sh_{3,0})$. □

Complete proofs of correctness and privacy for the above protocols will appear in the full version of the paper.

Protocol for General Access Structures. The main idea underlying the combinatorial schemes is that an orthogonal array is used as an *indexing structure* for several sharings of the secrets[4]. We can pursue the same idea in order to support general access structures. To explain the protocol and how to construct the *indexing structure*, let us consider a simple case. Let S_1, S_2, S_3 and S_4 be four servers, and let $\mathcal{P}_3 = \{\{S_1, S_2\}, \{S_2, S_3\}, \{S_3, S_4\}\}$ be an access structure on the set of servers. This access structure is well-studied in secret sharing scheme theory and its (optimal) information rate ρ is equal to $\frac{2}{3}$ (see [10]). Assume that the secret is a pair of values (k_1, k_2) belonging to $GF(q') \times GF(q')$. It can be shared among $\mathcal{P}_3$ as follows:

[4] Indeed, notice that even the constructions given in Table 4, can be re-phrased along the same line of the protocol described in Table 5. In this case the orthogonal array used is an $OA_1(k, m+1, n)$.

S_1	x	z	
S_2	$k_1 + x$	$k_2 + z$	w
S_3	$k_1 + w$	$k_2 + y$	z
S_4	w	y	

The values x, y, z, and w are random values in $GF(q')$. The dealer computes the above shares and sends to each server a row of the matrix.

We can construct a weak $\mathcal{P}_3$-DOT using the above secret sharing scheme as a building block. More precisely, each secret is shared many times with different instances of the secret sharing scheme. At the same time, an indexing matrix which represents all these sharings can be set up filling in the entries of each column using the same secret sharing scheme.

To exemplify, assume that we have $9 = 3^2$ secrets. Each secret (k_i, k_j) can be indexed by $(i, j) \in GF(3) \times GF(3)$. An indexing matrix can be set up, considering 3^4 sharings for each value of the key (i.e., the number of possible choices for x, y, z, and w when seen as elements belonging to $GF(3)$). For example, the restriction of the indexing matrix to the key (k_1, k_2), indexed by $(1, 2)$, is:

0	$\mathbf{(1,2)}$	$\cdots$	$\mathbf{(1,2)}$
1	$0 \quad 0$	$\cdots$	$2 \quad 2$
2	$1+0 \;\; 2+0 \;\; 0$	$\cdots$	$1+2 \;\; 2+2 \;\; 2$
3	$1+0 \;\; 2+0 \;\; 0$	$\cdots$	$1+2 \;\; 2+2 \;\; 2$
4	$0 \quad 0$	$\cdots$	$2 \quad 2$

Each of the 3^4 columns indexed by $(1, 2)$ represents a sharing of $(k_1, k_2) \in GF(q') \times GF(q')$. The receiver can choose one of those columns and can ask a subset $B \in \mathcal{P}_3$ to get the shares whose indices match the entries of the columns of the matrix corresponding to the servers in B. In our example the receiver, to retrieve (k_1, k_2), can choose the first column and can send $(1, 2, 0)$ to S_3 and $(0, 0)$ to S_4, receiving from S_3 all the shares corresponding to the fourth row of the matrix whose indices are $(1, 2, 0)$ (and, among these, is $(sh_1^{(1,2)}, sh_2^{(1,2)}, sh_0^{(1,2)})$) and from S_4 all the shares corresponding to the fifth row of the matrix whose indices are $(0, 0)$ (and, among these, is $(sh_0^{(1,2)}, sh_0^{(1,2)})$), where each $sh_j^{(1,2)} \in GF(q')$.

It is not difficult to see that the construction is correct, due to the reconstruction property of the secret sharing scheme. In our example $(sh_1^{(1,2)}, sh_2^{(1,2)})$ and $(sh_0^{(1,2)}, sh_0^{(1,2)})$ enable the receiver to recover (k_1, k_2). Moreover the scheme is secure since, for each subset of servers belonging to $\mathcal{P}_3$ and for each pair of different keys, say (k_i, k_j) and $(k_{i'}, k_{j'})$, looking at the restriction of the matrix to the columns indexed by the pairs (i, j) and (i', j'), there is no sequence of queries that enables to recover both secrets: *at least one* of the queries the receiver has to send to the servers to recover the second secret must be different in *at least one* of the entries from the queries that enable to recover the first one[5]. On the other hand, a forbidden subset of servers $F \notin \mathcal{P}_3$ neither get information about the secret $\mathcal{R}$ wishes to recover from the queries sent by her nor can they compute information about any secret, due to the security property of the secret sharing scheme. Notice that, if we have $n = q^2$ secrets, the construction seen before requires q^4 sharings for each secret, and an indexing matrix with q^6 columns.

At this point it is not difficult to figure out how the same strategy can be applied to any access structure. In this extended abstract, without going into details that will be given in the full paper, we would like just to point out the use of secret sharing schemes for the construction of both the indexing structure and the subsequent sharing of the secrets. Perhaps this design technique can be applied successfully to other cryptographic protocols.

5.2 Two-Round Constructions

It is possible to gain in terms of security and reduce the complexity if we allow an additional round of interaction between the receiver and the servers. A simple protocol is described in Table 6.

An easy check shows that the receiver always recovers the secret in which is interested and she gets no information on the other secret, since it is encrypted by the values of the other column. Moreover, a coalition of $k-1$ servers cannot find out which secret $\mathcal{R}$ has recovered, since the "label" specifying which secret each column encrypts can be recovered *only* by *all* the k servers. Finally, a coalition of $k-1$ servers cannot compute *any* secret since the coalition misses the value held by the k-th server.

It is worthwhile to point out that the two-round construction above described enjoys the further security property that is impossible to achieve using a one-round protocol: indeed, a coalition of $k-1$ servers and the receiver, after the latter has recovered one of the secret, still cannot compute the other without the help of the last server.

Notice that, if we compress the above protocol into one round, we can obtain a *random* DOT where the receiver *can recover one secret but she cannot choose which one*. This functionality can be realized if the servers simply send to the receiver the "addressing bits" and *all but one* of the values $v_0[i]$ and $v_1[i]$, for $i = 1, \ldots, k$. For example, one of the servers, say S_i, chooses uniformly at random which of the two values $v_0[i], v_1[i]$ to send to $\mathcal{R}$.

The above protocol can be extended to realize a DOT for a *general access structure* on the set of servers as well as a DOT for any number of secrets. The extensions can be done as follows: in order to implement a DOT for a general access structure $\mathcal{A}$ on the set of servers, say an $\mathcal{A}$-DOT-$\binom{2}{1}$, the bit r, which establishes which vector hides s_0, is shared among the m servers, according

[5] Arguing by contradiction it is possible to show that if there exists a sequence of queries (in our example a sequence of two queries) which enables to recover two different secrets (k_i, k_j) and $(k_{i'}, k_{j'})$ then $i = i'$ and $j = j'$.

A (k,k)-DOT-$\binom{2}{1}$ with one addressing bit Let $s_0, s_1 \in GF(q)$ be $\mathcal{S}$'s secrets. - $\mathcal{S}$ chooses k random bits r_i, and computes the bit r, by xoring the r_i's. Then, he sets up two vectors with entries in Z_q, v_0 and v_1, by choosing the first $k-1$ entries at random and computing $$v_0[k] = s_r - \sum_{i=1}^{k-1} v_0[i] \bmod q, \text{ and } v_1[k] = s_{(1-r)} - \sum_{i=1}^{k-1} v_1[i] \bmod q.$$ - Then, for $i = 1, \ldots, k$, he sends the bit r_i and the values $v_0[i]$ and $v_1[i]$ to server S_i. - In a first round of communication, $\mathcal{R}$ asks each server for the bit r_i and computes r. Then, for $i = 1, \ldots, k$, if $\mathcal{R}$ is interested in s_0 and $r = 0$, she asks server S_i for the value $v_0[i]$; otherwise, if $r = 1$, she asks for $v_1[i]$. Symmetrically, to recover s_1, if $r = 1$, she asks for $v_0[i]$, while if $r = 0$, she asks for $v_1[i]$. - Then, she sums up the received values mod$q$.

Table 6. Two-Round (k,k)-DOT-$\binom{2}{1}$

to a secret sharing scheme for $\mathcal{A}$. Then, if $r = 0$, the secret s_0 is shared by the first vector and s_1 by the second, according to a secret sharing scheme for $\mathcal{A}$; otherwise, s_0 is shared by the second vector and s_0 by the first. Once the receiver has recovered the value of r, contacting a subset of servers belonging to $\mathcal{A}$, she can recover one of the secret by sending a request for shares to a subset of servers (perhaps the same ones that were contacted before) belonging to $\mathcal{A}$. On the other hand, a (k,k)-DOT-$\binom{n}{1}$ requires that, instead of a bit r_i, each server has a value $r_i \in \{0, \ldots, n-1\}$, and, instead of two vectors sharing s_0 and s_1, there are exactly n vectors $v_0, \ldots, v_{n-1}$, sharing the secrets $s_0, \ldots, s_{n-1}$, respectively. The value $r = \sum r_j \bmod n$ establishes the correspondence between vectors and the n secrets. In other words, if $r = 2$ then the third vector v_2 shares s_0, the fourth shares s_1, and so on, following a cyclic order modulo n. Applying the same argument described before for the case of two secrets, it is not difficult to show that even this is correct and secure.

6 Conclusions

In this paper, we have studied unconditionally secure distributed oblivious transfer protocols. We have presented lower bounds on the resources required to implement such schemes, some impossibility results for one-round schemes, and new constructions which are optimal with respect to some of the given bounds. Moreover, we have shown that, with a second round of interaction, the highest possible security level in this model can be achieved with, at the same time, a suitable reduction of resources (randomness, memory storage and communication complexity). It is worthwhile to notice that the same effect can be achieved modifying the model for DOT by allowing the Sender to send information during the set up phase even to the receiver. In this case the two-round protocol we have shown in the previous section can be simply transformed in a one-round protocol. This is another example of a tradeoff.

Interesting open problems include the design of a single protocol meeting all the bounds given by the information theoretic analysis, as well as to find out settings which can benefit from the application of this distributed primitive. Along this line, in the full version of the paper, we will discuss some applications mainly related to contexts in which the privacy (anonymity) of the user must be guaranteed.

Acknowledgment.

D.R. Stinson's research is supported by NSERC grants IRC #216431-96 and RGPIN #203114-02. The authors would like to thank the referees for valuable suggestions and references.

References

1. D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, *Locally Random Reductions: Improvements and Applications*, Journal of Cryptology 10 (1), pp. 17-36, 1997.
2. A. Beimel, Y. Ishai, and T. Malkin, *Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing*, Advances in Cryptology: Proceedings of Crypto 2000, Springer-Verlag, vol. 1880, pp. 55-73, 2000.
3. M. Bellare and S. Micali, *Non-interactive Oblivious Transfer and Applications*, Advances in Cryptology: Crypto '89, Springer-Verlag, pp. 547-559, 1990.
4. G.R. Blakley. Safeguarding Cryptographic Keys. Proceedings of AFIPS 1979 National Computer Conference, Vol. 48, pp. 313–317, 1979.
5. M. Blum, *How to Exchange (Secret) Keys*, ACM Transactions of Computer Systems, vol. 1, No. 2, pp. 175-193, 1993.
6. C. Blundo, B. Masucci, D.R. Stinson and R. Wei, *Constructions and Bounds for Unconditionally Secure Non-Interactive Commitment Schemes*, Designs, Codes, and Cryptography, Vol. 26, pp. 97–110, 2002.
7. G. Brassard, C. Crepéau, and J.-M. Roberts, *Information Theoretic Reductions Among Disclosure Problems*, Proceedings of 27th IEEE Symposium on Foundations of Computer Science, pp. 168-173, 1986.
8. G. Brassard, C. Crepéau, and J.-M. Roberts, *All-or-Nothing Disclosure of Secrets*, Advances in Cryptology: Crypto '86, Springer-Verlag, vol. 263, pp. 234-238, 1987.
9. G. Brassard, C. Crepéau, and M. Sántha, *Oblivious Transfer and Intersecting Codes*, IEEE Transaction on Information Theory, special issue in coding and complexity, Vol. 42, No. 6, pp. 1769-1780, 1996.
10. R.M. Capocelli, A. De Santis, L. Gargano and U. Vaccaro, *On the Size of the Shares in Secret Sharing Schemes*, Advances in cryptology - CRYPTO'91, Lecture Notes in Computer Science, vol. 576, pp. 101–113, 1992.

11. B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, *Private Information Retrieval*, Proc. 36th IEEE Symposium on Foundations of Computer Science (FOCS), 1995, 41-50.
12. T. M. Cover and J. A. Thomas, **Elements of Information Theory**, John Wiley & Sons, 1991.
13. C. Crepéau, *Equivalence between to flavors of oblivious transfers*, Advances in Cryptology: Proceedings of Crypto '87, vol. 293, pp. 350-354, Springer Verlag, 1988.
14. C. Crepéau, *A Zero-Knowledge Poker Protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face*, Advances in Cryptology: Proceedings of Crypto '86, Springer-Verlag, pp. 239-247, 1987.
15. G. Di Crescenzo, Y. Ishai, and R. Ostrovsky, *Universal Service-Providers for Database private Information Retrieval*, Proc. of Seventeenth Annual ACM Symposium on Principles of Distributed Computing (PODC), 1998.
16. P. D'Arco and D.R. Stinson, *Generalized Zig-zag Functions and Oblivious Transfer Reductions*, Selected Areas in Cryptography SAC 2001, vol. 2259, pp. 87-103, 2001.
17. Y. Dodis and S. Micali, *Lower Bounds for Oblivious Transfer Reduction*, Advances in Cryptology: Proceedings of Eurocrypt '99, vol. 1592, pp. 42-54, Springer Verlag, 1999.
18. S. Even, O. Goldreich, and A. Lempel, *A Randomized Protocol for Signing Contracts*, Communications of the ACM 28, pp. 637-647, 1985.
19. M. Fisher, S. Micali, and C. Rackoff, *A Secure Protocol for the Oblivious Transfer*, Journal of Cryptology, vol. 9, No. 3, pp. 191-195, 1996.
20. Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, *Protecting Data Privacy in Private Information Retrieval Schemes*, Proc. of the 30th Annual ACM Symposium on Theory of Computing (STOC), 1998, pp. 151-160.
21. Y. Gertner, S. Goldwasser, and T. Malkin, *A Random Server Model for Private Information Retrieval or How to Achieve Information Theoretic PIR Avoiding Database Replication*, RANDOM 1998, Lecture Notes in Computer Science, Vol. 1518, pp. 200-217, 1998.
22. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan, *The Relationship between Public Key Encryption and Oblivious Transfer*, Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS 2000), pp. 325-339, 2000.
23. O. Goldreich, S. Micali, and A. Wigderson, *How to play ANY mental game or: A Completeness Theorem for Protocols with Honest Majority*, Proceedings of 19th Annual Symposium on Theory of Computing, pp. 20-31, 1987.
24. J. Kilian, *Founding Cryptography on Oblivious Transfer*, Proceedings of 20th Annual Symposium on Theory of Computing, pp. 20-31, 1988.
25. M. Naor and B. Pinkas, *Distributed Oblivious Transfer*, Advances in Cryptology: Proceedings of Asiacrypt '00, Springer-Verlag, pp. 205-219, 2000.
26. M. Naor, B. Pinkas, and R. Sumner, *Privacy Preserving Auctions and Mechanism Design*, ACM Conference on Electronic Commerce, 1999 available at *http://www.wisdom.weizmann.ac.il/ naor/onpub.html*
27. M. Rabin, *How to Exchange Secrets by Oblivious Transfer*, Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.
28. R. Rivest, *Unconditionally Secure Committment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer*, manuscript. Available: http://theory.lcs.mit.edu/~rivest/publications.html
29. A. Shamir. How to Share a Secret. Communications of ACM, vol. 22, n. 11, pp. 612–613, 1979.

30. D.R. Stinson. Bibliography on Secret Sharing Schemes. http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html.
31. D.R. Stinson. An explication of secret sharing schemes. Des. Codes Cryptogr., 2, 357–390, 1992.
32. W. Tzeng, *Efficient* 1-*out-of-n Oblivious Transfer Schemes*, Proceedings of PKC 2002, Lecture Notes in Computer Science, Vol. 2274, pp. 159-171, 2002.
33. S. Wiesner, *Conjugate Coding*, SIGACT News 15, pp. 78-88, 1983.

A Information Theory Elements

In this section we give some basic concepts about Information Theory. However, the reader is referred to [12] for a complete treatment of the subject.

Let $\mathbf{X}$ be a random variable taking values on a set X according to a probability distribution $\{P_{\mathbf{X}}(x)\}_{x\in X}$. The *entropy* of $\mathbf{X}$, denoted by $H(\mathbf{X})$, is defined as

$$H(\mathbf{X}) = -\sum_{x\in X} P_{\mathbf{X}}(x)\log P_{\mathbf{X}}(x),$$

where the logarithm is relative to the base 2. The entropy satisfies $0 \leq H(\mathbf{X}) \leq \log|X|$, where $H(\mathbf{X}) = 0$ if and only if there exists $x_0 \in X$ such that $Pr(\mathbf{X} = x_0) = 1$; whereas, $H(\mathbf{X}) = \log|X|$ if and only if $Pr(\mathbf{X} = x) = 1/|X|$, for all $x \in X$. The entropy of a random variable is usually interpreted as

- a measure of the "equidistribution" of the random variable
- a measure of the amount of information given on average by the random variable

Given two random variables $\mathbf{X}$ and $\mathbf{Y}$, taking values on sets X and Y, respectively, according to a probability distribution $\{P_{\mathbf{XY}}(x,y)\}_{x\in X, y\in Y}$ on their Cartesian product, the *conditional entropy* $H(\mathbf{X}|\mathbf{Y})$ is defined as

$$H(\mathbf{X}|\mathbf{Y}) = -\sum_{y\in Y}\sum_{x\in X} P_{\mathbf{Y}}(y)P_{\mathbf{X}|\mathbf{Y}}(x|y)\log P_{\mathbf{X}|\mathbf{Y}}(x|y).$$

It is easy to see that

$$H(\mathbf{X}|\mathbf{Y}) \geq 0. \tag{9}$$

with equality if and only if X is a function of Y. The conditional entropy is a measure of the amount of information that $\mathbf{X}$ still has, once given $\mathbf{Y}$.

The *mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ is given by

$$I(\mathbf{X};\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}).$$

Since, $I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X})$ and $I(\mathbf{X};\mathbf{Y}) \geq 0$, it is easy to see that

$$H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y}), \tag{10}$$

with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. The mutual information is a measure of the common information between $\mathbf{X}$ and $\mathbf{Y}$.

Given $n+1$ random variables, $\mathbf{X}_1 \dots \mathbf{X}_n \mathbf{Y}$, the entropy of $\mathbf{X}_1 \dots \mathbf{X}_n$ given $\mathbf{Y}$ can be written as

$$H(\mathbf{X}_1 \dots \mathbf{X}_n | \mathbf{Y}) = H(\mathbf{X}_1 | \mathbf{Y}) + H(\mathbf{X}_2 | \mathbf{X}_1 \mathbf{Y}) + \cdots + H(\mathbf{X}_n | \mathbf{X}_1 \dots \mathbf{X}_{n-1} \mathbf{Y}). \quad (11)$$

Therefore, for any sequence of n random variables, $\mathbf{X}_1 \dots \mathbf{X}_n$, it holds that

$$H(\mathbf{X}_1 \dots \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i | \mathbf{X}_1 \dots \mathbf{X}_{i-1}) \leq \sum_{i=1}^{n} H(\mathbf{X}_i). \quad (12)$$

Moreover, the above relation implies that, for each $k \leq n$,

$$H(\mathbf{X}_1 \dots \mathbf{X}_n) \geq H(\mathbf{X}_1 \dots \mathbf{X}_k). \quad (13)$$

Given three random variables, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, the *conditional mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ can be written as

$$I(\mathbf{X}; \mathbf{Y} | \mathbf{Z}) = H(\mathbf{X} | \mathbf{Z}) - H(\mathbf{X} | \mathbf{Z}\ \mathbf{Y}) = H(\mathbf{Y} | \mathbf{Z}) - H(\mathbf{Y} | \mathbf{Z}\ \mathbf{X}) = I(\mathbf{Y}; \mathbf{X} | \mathbf{Z}). \quad (14)$$

Since the conditional mutual information $I(\mathbf{X}; \mathbf{Y} | \mathbf{Z})$ is always non-negative we get

$$H(\mathbf{X} | \mathbf{Z}) \geq H(\mathbf{X} | \mathbf{Z}\ \mathbf{Y}). \quad (15)$$