On Unconditionally Secure Distributed Oblivious Transfer*

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

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy
e-mail: {carblu, paodar, ads}@dia.unisa.it

[2] School of Computer Science
University of Waterloo, N2L 3G1, Waterloo, Ontario, Canada
e-mail: dstinson@cacr.math.uwaterloo.ca

April 16, 2003

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 any 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 [38], and subsequently defined in different forms [24, 10], 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

*A preliminary version of this paper appeared in the Proceedings of the SAC 2002 Workshop.

†This research was partially done while the author was a Post-Doc Fellow at the Department of Combinatorics and Optimization of the University of Waterloo, Ontario, Canada.

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 [25].

The second OT definition was given in [24]. 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 [24] showed a first application to signing contracts.

The last and more general form of OT was introduced in [10], under the name of *all-or-nothing Disclosure of Secrets*, even if the same concept was born in an artificial intelligence context [44], 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 [11, 9, 15], and Kilian in [30] 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., [15, 11, 23, 19, 3, 21, 22, 16, 29, 36, 28], just to name few examples).

OUR CONTRIBUTION. In this paper we study *unconditionally secure distributed oblivious transfer protocols*, introduced in [34] in order to strengthen the security of protocols designed for electronic auctions [36]. 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 value 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 [13]. 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 [26], 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 [26] 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, 27, 18].

Rivest's model in [39], where a trusted initializer participates *only* during the set up phase of the system (see also [8]), provides a very close setting to the one described in [34] and considered in this paper. A paper which deals with distributed oblivious transfer implementations, close to the setting introduced in [34] (but not unconditionally secure) is [43]. Finally, unconditionally secure distributed oblivious transfer schemes for general access structures have been recently studied in [37].

In our constructions we use secret sharing schemes. Secret sharing schemes were introduced in 1979 by Blakley [4] and Shamir [40], and have been extensively studied during the last years. The reader can find an introduction in [42] and references to the literature in [41].

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 (see Figure 1):

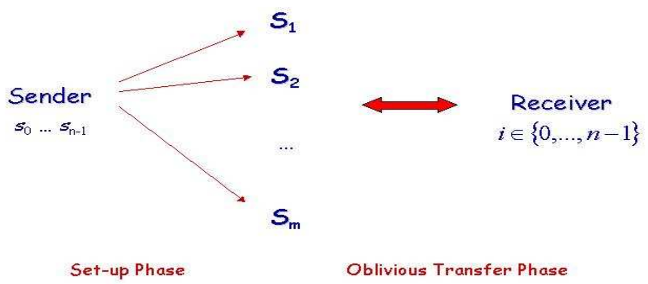

Figure 1: Distributed Oblivious Transfer

SET-UP 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[1]. 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 $k-1$ corrupted servers cannot learn about the n secrets more than can be learned by the Receiver herself.

Notice that, in [34], 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[2] 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 $\mathcal{S}_{j_1}, \ldots, \mathcal{S}_{j_{k-1}}$ hold a modified version of the programs, denoted by $\overline{P_{j_1}}, \ldots, \overline{P_{j_{k-1}}}$. 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 sequences of n 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

[1]Along the same line of [34], we assume the existence of an external mechanism which guarantees that the receiver can contact no more than k servers. This issue is independent of the distributed oblivious transfer scheme and, hence, it is not considered in this paper. The reader is referred to [34] for some techniques to solve the problem.

[2]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 a *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-$\binom{n'}{1}$.

in W, by $\mathbf{T}$ the random variable representing the choice of an index i in $T = \{1, \ldots, n\}$, and by $\mathbf{W}_T$ the random variable representing the secret defined by the choice of the Receiver, we can define the conditions that a (k, m)-DOT-$\binom{n}{1}$ oblivious transfer protocol must satisfy as follows[3]: First of all, we require that the program of the Receiver, her choice, and the programs of the servers, completely determine the transcript of the interaction Receiver-servers. In other words, for any subset of k indices $X \subseteq \{1, \ldots, m\}$, it holds that[4]:

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

Then, we state the following:

Definition 2.1 *The sequence of programs $[S, P_1, \ldots P_m, R]$ is* correct *for (k, m)-DOT-$\binom{n}{1}$ if, for any subset of k indices $X \subseteq \{1, \ldots, m\}$, it holds that*

$$H(\mathbf{W}_T | \mathbf{C}_X, \mathbf{R}, \mathbf{T}) = 0. \tag{2}$$

Notice that the above definition means that, after interacting with any 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 2.2 *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 $k-1$ indices $X \subset \{1, \ldots, m\}$, and for any sequence $\overline{\mathbf{P}}_X$, it holds that*

$$H(\mathbf{T} | \overline{\mathbf{P}}_X, \mathbf{C}_X) = H(\mathbf{T}). \tag{3}$$

- *for any $\overline{\mathbf{R}}$, for any set of k indices $X \subseteq \{1, \ldots, m\}$, and for any $\mathbf{T}$, there exists a random variable $\tilde{\mathbf{T}} = f(\mathbf{T}, \overline{\mathbf{R}})$ such that*

$$H(\mathbf{W} | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \mathbf{W}_{\tilde{T}}) = H(\mathbf{W} | \mathbf{W}_{\tilde{T}}). \tag{4}$$

- *for any subset of $k-1$ indices $X \subset \{1, \ldots, m\}$, for any $\overline{\mathbf{R}}$, and for any sequence $\overline{\mathbf{P}}_X$, it holds that*

$$H(\mathbf{W} | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X) = H(\mathbf{W}). \tag{5}$$

Moreover, we will say that the sequence of programs $[S, P_1, \ldots P_m, R]$ defines a *strong* (k, m)-DOT-$\binom{n}{1}$ if a further security condition is satisfied. More precisely,

Definition 2.3 *The sequence of programs $[S, P_1, \ldots P_m, R]$ defines a* strong *(k, m)-DOT-$\binom{n}{1}$ if it is correct and private and it holds that:*

- *for any subset of $k-1$ indices $X \subset \{1, \ldots, m\}$ and any subset of k indices $Y \subseteq \{1, \ldots, m\}$, for any sequence $\overline{\mathbf{P}}_X$, for any $\overline{\mathbf{R}}$, and for any $\mathbf{T}$, there exists a random variable $\tilde{\mathbf{T}} = f(\mathbf{T}, \overline{\mathbf{R}})$ such that*

$$H(\mathbf{W} | \mathbf{T}, \overline{\mathbf{R}}, \overline{\mathbf{P}}_X, \mathbf{C}_Y, \mathbf{W}_{\tilde{T}}) = H(\mathbf{W} | \mathbf{W}_{\tilde{T}}). \tag{6}$$

---

[3] 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.

[4] To simplify the notation, if, for $i = 1, \ldots, n$, $\mathbf{A}_i$ is a random variable, and $X = \{j_1, \ldots, j_m\} \subseteq \{1, \ldots, n\}$ is a subset of indices such that $j_1 < \ldots < j_m$, then $\mathbf{A}_X$ denotes the sequence of random variables $\mathbf{A}_{j_1} \ldots \mathbf{A}_{j_m}$.

Conditions (3) and (4) ensure that a dishonest coalition of $k-1$ 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 $S_1, \ldots, S_m$. Moreover, condition (4) rules out the possibility for the Receiver of getting joint information about some of the secrets. Condition (5) takes into account the possibility of an attack against $\mathcal{S}$ performed either by $k-1$ servers alone or with the cooperation of $\mathcal{R}$. The condition states that such coalitions do not gain 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. Notice that, in our model (inspired by [11]), conditions (4) and (6) are not independent: indeed, (6) implies (4).

Later on we will show that condition (6) cannot be achieved with *only one round of interaction*. In other words, a strong (k,m)-DOT-$\binom{n}{1}$ cannot be realized by means of a one-round protocol. On the other hand, with two rounds of interaction, this level of privacy can be obtained.

REMARK. It is straightforward to see that conditions (3), (5), and (6) hold even if the coalition of dishonest servers has size less than $k-1$. Formally, such a property can be derived by applying to conditions (3), (5), and (6) property (21) of Appendix A.

3 Lower Bounds

Using some information theory tools, we can prove bounds on the memory storage, on the communication complexity, and on the randomness needed by a correct and private DOT scheme.

The following simple lemma shows that, given four random variables $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, if $\mathbf{B}$ is a function of $\mathbf{C}$ and $\mathbf{D}$, then $\mathbf{B}$ and $\mathbf{D}$ give less information on $\mathbf{A}$ than $\mathbf{C}$ and $\mathbf{D}$.

Lemma 3.1 *Let $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ be four random variables such that $H(\mathbf{B}|\mathbf{C},\mathbf{D})=0$. Then, $H(\mathbf{A}|\mathbf{B},\mathbf{D}) \geq H(\mathbf{A}|\mathbf{C},\mathbf{D})$.*

Proof. Notice that, (14) and (21) of Appendix A imply

$$0 \leq H(\mathbf{B}|\mathbf{A},\mathbf{C},\mathbf{D}) \leq H(\mathbf{B}|\mathbf{C},\mathbf{D}) = 0.$$

Therefore, from (20) of Appendix A, we get

$$I(\mathbf{A};\mathbf{B}|\mathbf{C},\mathbf{D}) = H(\mathbf{A}|\mathbf{C},\mathbf{D}) - H(\mathbf{A}|\mathbf{B},\mathbf{C},\mathbf{D}) = H(\mathbf{B}|\mathbf{C},\mathbf{D}) - H(\mathbf{B}|\mathbf{A},\mathbf{C},\mathbf{D}) = 0.$$

Hence, $H(\mathbf{A}|\mathbf{C},\mathbf{D}) = H(\mathbf{A}|\mathbf{B},\mathbf{C},\mathbf{D})$. Finally, from (21) of Appendix A, it results $H(\mathbf{A}|\mathbf{B},\mathbf{D}) \geq H(\mathbf{A}|\mathbf{B},\mathbf{C},\mathbf{D})$. Thus, we have that $H(\mathbf{A}|\mathbf{B},\mathbf{D}) \geq H(\mathbf{A}|\mathbf{C},\mathbf{D})$, which proves the lemma. ∎

Notice that condition (5) of Definition 2.2 implies that a coalition of $k-1$ servers and the Receiver get no information about *any* single secret. More precisely, we state the following:

Lemma 3.2 *In any correct and private (k,m)-DOT-$\binom{n}{1}$ scheme, for any set of $k-1$ indices $X \subset \{1,\ldots,m\}$, for any $\overline{R}$, and for any sequence of programs $\overline{P}_X$, it holds that*

$$H(\mathbf{W}_T|\mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X) = H(\mathbf{W}_T). \tag{7}$$

Proof. Indeed, denoting by $\mathbf{W} \setminus \mathbf{W}_T$ the sequence of all secrets in $\mathbf{W}$ but $\mathbf{W}_T$, using condition (5), and properties (17) and (16) of Appendix A, it results

$$\begin{aligned} H(\mathbf{W}) &= H(\mathbf{W}_T, \mathbf{W} \setminus \mathbf{W}_T | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X) \\ &= H(\mathbf{W}_T | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X) + H(\mathbf{W} \setminus \mathbf{W}_T | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X, \mathbf{W}_T) \\ &\leq H(\mathbf{W}_T) + H(\mathbf{W} \setminus \mathbf{W}_T | \mathbf{W}_T) = H(\mathbf{W}), \end{aligned}$$

which implies $H(\mathbf{W}_T | \mathbf{T}, \overline{\mathbf{R}}, \mathbf{C}_X, \overline{\mathbf{P}}_X) = H(\mathbf{W}_T)$. ■

The following lemma shows that the programs held by a subset of k servers and the information held by the Receiver are enough to recover all the secrets. More precisely:

Lemma 3.3 *In any correct and private (k, m)-DOT-$\binom{n}{1}$ scheme, for any subset of k indices $X \subset \{1, \ldots, m\}$ and any sequence of programs P_X, it holds that*

$$H(\mathbf{W} | \mathbf{P}_X, \mathbf{R}, \mathbf{T}) = 0.$$

Proof. From (17) and (21) of Appendix A, it holds that

$$H(\mathbf{W} | \mathbf{P}_X, \mathbf{R}, \mathbf{T}) \leq \sum_{i=1}^{n} H(\mathbf{W}_T | \mathbf{P}_X, \mathbf{R}, \mathbf{T} = i).$$

Since condition (1) establishes that $H(\mathbf{C}_X | \mathbf{P}_X, \mathbf{R}, \mathbf{T}) = 0$, setting $\mathbf{A} = \mathbf{W}_T, \mathbf{B} = \mathbf{C}_X, \mathbf{C} = \mathbf{P}_X$, and $\mathbf{D} = (\mathbf{R}, \mathbf{T} = i)$, and applying Lemma 3.1, it holds that

$$H(\mathbf{W}_T | \mathbf{P}_X, \mathbf{R}, \mathbf{T} = i) \leq H(\mathbf{W}_T | \mathbf{C}_X, \mathbf{R}, \mathbf{T} = i).$$

The above inequality and Definition 2.1 imply that

$$\sum_{i=1}^{n} H(\mathbf{W}_T | \mathbf{P}_X, \mathbf{R}, \mathbf{T} = i) \leq \sum_{i=1}^{n} H(\mathbf{W}_T | \mathbf{C}_X, \mathbf{R}, \mathbf{T} = i) = 0.$$

■

Every program P_j, held by server S_j, given $k - 1$ other programs and the information held by the Receiver, has at least as much information as the whole sequence of n secrets. This property is formally stated by the following:

Lemma 3.4 *In any correct and private (k, m)-DOT-$\binom{n}{1}$ scheme, for any subset of indices $X \subset \{1, \ldots, m\}$, where $1 \leq |X| \leq k - 1$, and for any index $j \notin X$, it holds that*

$$H(\mathbf{P}_j | \mathbf{P}_X, \mathbf{R}, \mathbf{T}) \geq H(\mathbf{W}).$$

Proof. Let $Y \subset \{1, \ldots, m\}$, such that $|Y| = k - |X| - 1$, $j \notin Y$, and $X \cap Y = \emptyset$. According to Appendix A, the mutual information $I(\mathbf{W}; \mathbf{P}_j | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T})$ can be written as

$$H(\mathbf{W} | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T}) - H(\mathbf{W} | \mathbf{P}_{X \cup Y \cup \{j\}}, \mathbf{R}, \mathbf{T}) = H(\mathbf{P}_j | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T}) - H(\mathbf{P}_j | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T}, \mathbf{W}).$$

From (21) of Appendix A and condition (5) of Definition 2.2, we get $H(\mathbf{W} | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T}) = H(\mathbf{W})$, from (14) of Appendix A we get $H(\mathbf{P}_j | \mathbf{P}_{X \cup Y}, \mathbf{W}, \mathbf{R}, \mathbf{T}) \geq 0$, and, from Lemma 3.3, we get $H(\mathbf{W} | \mathbf{P}_{X \cup Y \cup \{j\}}, \mathbf{R}, \mathbf{T}) = 0$. Therefore,

$$H(\mathbf{P}_j | \mathbf{P}_{X \cup Y}, \mathbf{R}, \mathbf{T}) \geq H(\mathbf{W}).$$

Applying property (21) of Appendix A, we get

$$H(\mathbf{P}_j|\mathbf{P}_X, \mathbf{R}, \mathbf{T}) \geq H(\mathbf{P}_j|\mathbf{P}_{X\cup Y}, \mathbf{R}, \mathbf{T}) \geq H(\mathbf{W}).$$

■

When we want to set up a cryptographic protocol we need random bits. This resource is usually referred to as the *randomness* [5]. The randomness of a scheme can be measured in different ways. Knuth and Yao [31] proposed the following approach: Let `Alg` be an algorithm that generates the probability distribution $P = \{p_1, \ldots, p_n\}$, using only independent and unbiased random bits. Denote by $T(\texttt{Alg})$ the average number of random bits used by `Alg` and let $T(\mathbf{P}) = \min_{\texttt{Alg}} T(\texttt{Alg})$. The value $T(\mathbf{P})$ is a measure of the average number of random bits needed to simulate the random source described by the probability distribution P. In [31] the following result was shown:

Theorem 3.5 $H(\mathbf{P}) \leq T(\mathbf{P}) < H(\mathbf{P}) + 2$.

Thus, the entropy of a random source is very close to the average number of unbiased random bits necessary to simulate the source. Hence, it is a natural measure of the randomness of a scheme. It is easy to see that the randomness needed to set up the m servers can be lower bounded by $H(\mathbf{P}_1, \ldots, \mathbf{P}_m)$.

Using the above results, we can establish a lower bound on the information content of the program that each server has to store and on randomess needed to set up a correct and private (k, m)-DOT-$\binom{n}{1}$ scheme. More precisely, we can show the following result:

Theorem 3.6 *In any correct and private (k, m)-DOT-$\binom{n}{1}$ scheme, for any subset $X \subseteq \{1, \ldots, m\}$, where $1 \leq |X| \leq k$, it holds that*

$$H(\mathbf{P}_X) \geq |X| \cdot H(\mathbf{W}).$$

Proof. Applying (17) and (21) of Appendix A, and Lemma 3.4, it holds that

$$\begin{aligned} H(\mathbf{P}_X) &\geq \sum_{\ell \in X} H(\mathbf{P}_\ell|\mathbf{P}_{X\setminus\{\ell\}}, \mathbf{R}, \mathbf{T}) \\ &\geq |X| \cdot H(\mathbf{W}). \end{aligned}$$

■

The above theorem implies the following results:

- **Server Memory Storage.** Each server S_j has to store at least $H(\mathbf{W})$ bits, since $H(\mathbf{P}_j) \geq H(\mathbf{W})$.

- **Randomness.** In order to set up the scheme, the Sender needs at least $kH(\mathbf{W})$ random bits, since if $|X| = k$, we get that $H(\mathbf{P}_1, \ldots, \mathbf{P}_m) \geq H(\mathbf{P}_X) \geq kH(\mathbf{W})$.

Notice that Theorem 3.6 holds only for subsets X such that $1 \leq |X| \leq k$. For any X of size $|X| \geq k$, the bound stays the same (i.e., $H(\mathbf{P}_X) \geq kH(\mathbf{W})$).

The following lemma enables us to establish a lower bound on the complexity of each interaction receiver-servers.

[5] A detailed analysis of the randomness in distribution protocols can be found in [6].

Lemma 3.7 *In any correct and private (k,m)-DOT-$\binom{n}{1}$ scheme, for any subset of indices $X \subset \{1,\ldots,m\}$, where $1 \le |X| \le k-1$, and for any index $j \notin X$,*

$$H(\mathbf{C}_j|\mathbf{C}_X,\mathbf{R},\mathbf{T}) \ge H(\mathbf{W}_T).$$

Proof. Let $Y \subset \{1,\ldots,m\}$, such that $|Y| = k-|X|-1$, $j \notin Y$, and $X \cap Y = \emptyset$. The mutual information $I(\mathbf{W}_T;\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T})$ can be written either as

$$H(\mathbf{W}_T|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T}) - H(\mathbf{W}_T|\mathbf{C}_{X\cup Y\cup\{j\}},\mathbf{R},\mathbf{T})$$

or as

$$H(\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T}) - H(\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T},\mathbf{W}_T).$$

Since Lemma 3.2 and property (21) of Appendix A imply that $H(\mathbf{W}_T|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T}) = H(\mathbf{W}_T)$, from Definition 2.1 and (21) of Appendix A we get that $H(\mathbf{W}_T|\mathbf{C}_{X\cup Y\cup\{j\}},\mathbf{R},\mathbf{T}) = 0$, and from (14) of Appendix A we get that $H(\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T},\mathbf{W}_T) \ge 0$, it holds that

$$H(\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T}) \ge H(\mathbf{W}_T).$$

The result follows observing that (21) of Appendix A implies

$$H(\mathbf{C}_j|\mathbf{C}_X,\mathbf{R},\mathbf{T}) \ge H(\mathbf{C}_j|\mathbf{C}_{X\cup Y},\mathbf{R},\mathbf{T}).$$

■

Using the above lemma, we can state the following theorem.

Theorem 3.8 *In any correct and private (k,m)-DOT-$\binom{n}{1}$ scheme, for any $X \subseteq \{1,\ldots,m\}$, where $1 \le |X| \le k$, it holds that*

$$H(\mathbf{C}_X) \ge |X| \cdot H(\mathbf{W}_T).$$

Proof. From (21), (17) of Appendix A, and Lemma 3.7, it holds that

$$\begin{aligned} H(\mathbf{C}_X) &\ge H(\mathbf{C}_X|\mathbf{R},\mathbf{T}) \\ &\ge \sum_{\ell\in X} H(\mathbf{C}_\ell|\mathbf{R},\mathbf{T},\mathbf{C}_{X\setminus\{\ell\}}) \\ &\ge |X| \cdot H(\mathbf{W}_T). \end{aligned}$$

■

The above theorem implies the following results:

- **Interaction Receiver-Server.** The Receiver and a Server need to exchange at least $H(\mathbf{W}_T)$ bits, since $H(\mathbf{C}_j) \ge H(\mathbf{W}_T)$.
- **Interaction Receiver-Servers.** The Receiver and k Servers need to exchange at least $k \cdot H(\mathbf{W}_T)$ bits, since $H(\mathbf{C}_X) \ge k \cdot H(\mathbf{W}_T)$.

AN IMPROVEMENT OF THE LOWER BOUNDS. Notice that Theorems 3.6 and 3.8 can be strengthened by using previous results obtained in the analysis of Secret Sharing Schemes. First of all, notice that there could be some value $\omega \in W$ such that $Pr(\mathbf{W} = \omega) = 0$ (i.e., such sequences of secrets are never chosen by the Sender). Therefore, let

$$\text{Support}(\mathbf{W}) = \{\omega \in W | Pr(\mathbf{W} = \omega) > 0\}$$

be the set of values assumed by $\mathbf{W}$ with positive probability. Since Theorems 3.6 and 3.8 must hold *for any* probability distribution on the set of secrets W, then they must hold also for the special case of the *uniform* one. In such a case $H(\mathbf{W}) = \log|Support(\mathbf{W})|$ and $H(\mathbf{W}_T) = \log|Support(\mathbf{W}_T)|$. Applying the same techniques employed in [7], the above intuition can be formalized, and we can prove the following result:

Theorem 3.9 *In any correct and private (k,m)-DOT-$\binom{n}{1}$ scheme, for any $X = \{j_1,\ldots,j_s\} \subseteq \{1,\ldots,m\}$, where $1 \leq |X| \leq k$, it holds that*

$$H(\mathbf{P}_X) \geq |X| \cdot \log|\text{Support}(\mathbf{W})|, \text{ and } H(\mathbf{C}_X) \geq |X| \cdot \log|\text{Support}(\mathbf{W}_T)|.$$

TIGHTNESS OF THE BOUNDS. The lower bound on the randomness needed to set up a (k,m)-DOT-$\binom{n}{1}$, derived from Theorem 3.8, is tight since the protocol we give in Tables 3 and 4 meets the bound by equality. The same protocol shows that both the lower bounds (randomness and memory storage), given by Theorem 3.6, are tight when the Sender has $n = 2$ secrets.

DISTRIBUTED OBLIVIOUS TRANSFER (k,m)-DOT-$\binom{n}{r}$. An extended version of (k,m)-DOT-$\binom{n}{1}$, which we denote by (k,m)-DOT-$\binom{n}{r}$, enables the receiver to recover, by interacting with a subset of k servers at his own choosing, r secrets instead of a single one. Such a protocol can be defined by means of Definitions 2.1, 2.2, and 2.3 as well, by introducing a minor modification: Instead of a single index i, the Receiver holds an r-tuple of indices. Therefore, $\mathbf{T}$ is a random variable taking values over T^r, and $\mathbf{W}_T$ is an r-tuple of random variables, representing an r-tuple of secrets the receiver can recover. Therefore, the analysis we have done holds also for such an extension of the model, i.e., Theorems 3.6 and 3.8 apply.

4 The One-Round Case

As we were claiming before, we can show that with a one-round protocol a strong (k,m)-DOT-$\binom{n}{1}$ cannot be realized. Before all, notice that if the protocol is one-round, then the interaction between the Receiver and server S_j is given by a query Q_j, sent by the Receiver, and an answer A_j, sent by S_j. Hence, $C_j = (Q_j, A_j)$. Therefore, condition (1) becomes: for any $j = 1,\ldots,m$, it holds that

$$H(\mathbf{Q}_j|\mathbf{R}, \mathbf{T}) = 0 \quad \text{and} \quad H(\mathbf{A}_j|\mathbf{Q}_j, \mathbf{P}_j) = 0. \tag{8}$$

Moreover, Definition 2.1 is re-stated as follows:

Definition 4.1 *The sequence of programs $[S, P_1, \ldots P_m, R]$ is* correct *for one-round (k,m)-DOT-$\binom{n}{1}$ if, for any $X \subseteq \{1,\ldots,m\}$, it holds that*

$$H(\mathbf{W}_T|\mathbf{Q}_X, \mathbf{A}_X, \mathbf{R}, \mathbf{T}) = 0. \tag{9}$$

Along the same line can be re-stated Definition 2.2. For one-round schemes we can prove that a single server can help the Receiver to recover all the secrets, once the Receiver has legally retrieved the secret of his choice. Because of Lemma 3.2, a set of $k-1$ query-answer pairs, and the information held by the Receiver, do not give any information about the secret the Receiver is trying to recover. This property, along with Definition 4.1, implies that, for any sequence of $k-1$ pairs, the k-th pair query-answer enables the recovering of *any* secret (otherwise, the sequence of $k-1$ query-answer pairs would leak partial information, i.e., that some secret cannot be reconstructed). Therefore, if the Receiver, after having legally recovered one secret, colludes with a single server, using a subset of $k-1$ query-answer pairs from the transcript of the previous interaction, and a k-th pair, constructed with the help of the cheating server, can recover any other secret. More formally, we can show the following:

Theorem 4.2 *In any one-round protocol for (k,m)-DOT-$\binom{n}{1}$, for any subset $X \subseteq \{1,\ldots,m\}$, and any $j \notin X$, it holds that*

$$H(\mathbf{W}|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T}) = 0.$$

Proof. Let Q_X be a set of $k-1$ possible queries for servers S_X, and let A_X be the corresponding answers. Lemma 3.2 implies that $H(\mathbf{W}_T|\mathbf{T}, \overline{\mathbf{R}}, \mathbf{Q}_X, \mathbf{A}_X) = H(\mathbf{W}_T)$, while Definition 4.1 implies that, given another pair (Q_j, A_j) for a k-th server, it holds that $H(\mathbf{W}_T|\mathbf{T}, \mathbf{R}, \mathbf{Q}_{X\cup\{j\}}\, \mathbf{A}_{X\cup\{j\}}) = 0$. Therefore, given Q_X and A_X, *for any* index i, there exists a query Q_j^i and an answer A_j^i, such that

$$H(\mathbf{W}_T|\mathbf{Q}_j = Q_j^i,\ \mathbf{Q}_X = Q_X, \mathbf{A}_j = A_j^i,\ \mathbf{A}_X = A_X, \mathbf{R}, \mathbf{T} = i) = 0.$$

Since, from condition (8), the answer given by S_j depends only on his own program P_j, and on the received query (i.e., $H(\mathbf{A}_j|\mathbf{Q}_j\mathbf{P}_j) = 0$), we claim that

$$H(\mathbf{W}|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T}) = 0.$$

Indeed, from (17) and (21) of Appendix A,

$$H(\mathbf{W}|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T}) \leq \sum_{i=1}^{n} H(\mathbf{W}_T|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T} = i)$$

and, applying properties (8) and (22) of Appendix A, and Lemma 3.1, we get that

$$\begin{aligned} H(\mathbf{W}_T|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T} = i) &= H(\mathbf{W}_T|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_j, \mathbf{Q}_X, \mathbf{R}, \mathbf{T} = i) \\ &\leq H(\mathbf{W}_T|\mathbf{A}_j, \mathbf{A}_X, \mathbf{Q}_j, \mathbf{Q}_X, \mathbf{R}, \mathbf{T} = i) = 0 \end{aligned}$$

Therefore, the Receiver and S_j can recover all the secrets and the result holds. ■

A consequence of this impossibility result for one-round protocols is that the highest privacy level aimed in [36] with this approach cannot be achieved.

Remark 1. Notice that the above result can be easily extended to $\mathcal{A}$-DOT-$\binom{n}{1}$ schemes for a general access structure $\mathcal{A}$ on the set of servers, which is a *specification* of all the possible subsets of servers that a receiver has to contact in order to recover a secret. If X denotes a subset of indices of servers such that $S_X \notin \mathcal{A}$ but $S_X \cup \{S_j\} \in \mathcal{A}$, where $j \notin X$, we can apply exactly the same argument applied before. Once the receiver has recovered a secret by interacting with $S_X \cup \{S_j\}$, a collusion with S_j enables him to recover all the secrets.

REMARK 2. It is possible to show that, if conditions (3), (5), and (6) of Definitions 2.2 and 2.3 must hold, in the threshold case, against a coalition of servers S_X such that $|X| \leq t$, then the Receiver, after having legally recovered one secret, can recover *all* the others if she colludes with $k-t$ servers. In other words, $H(\mathbf{W}|\mathbf{P}_j, \mathbf{A}_X, \mathbf{Q}_X, \mathbf{R}, \mathbf{T}) = 0$. Also this extension of Theorem 4.2 is quite straightforward.

It is not difficult to see that, in any one-round DOT scheme, a sequence of k queries determines $\mathbf{T}$ *uniquely*. More precisely, we can show the following:

Lemma 4.3 *In any correct and private one-round (k, m)-DOT-$\binom{n}{1}$ scheme, for any subset $X \subset \{1, \ldots, m\}$, it holds that*

$$H(\mathbf{T}|\mathbf{Q}_X) = 0 \tag{10}$$

Proof. Applying properties (17) and (21) of Appendix A, and condition (8), it is easy to see that $H(\mathbf{Q}_X|\mathbf{R}, \mathbf{T}) = 0$. Hence, the sequence of queries Q_X is function only of R and T. However, if there were two pairs (R_1, i_1) and (R_2, i_2), where $i_1 \neq i_2$, such that $H(\mathbf{Q}_X|\mathbf{R} = R_1, \mathbf{T} = i_1) = 0$, and $H(\mathbf{Q}_X|\mathbf{R} = R_2, \mathbf{T} = i_2) = 0$, property (8) implies that

$$H(\mathbf{W}_T|\mathbf{A}_X, \mathbf{Q}_X, \mathbf{R} = R_1, \mathbf{T} = i_1) = 0,$$

and

$$H(\mathbf{W}_T|\mathbf{A}_X, \mathbf{Q}_X, \mathbf{R} = R_2, \mathbf{T} = i_2) = 0.$$

But this is clearly a contradiction with respect to condition (4) of Definition 2.2 which, for each choice of the Receiver, excludes the possibility of getting information about two different secrets. Therefore, there must exist only one i. Hence, $H(\mathbf{T}|\mathbf{Q}_X) = 0$. ∎

Using the above lemma, we can show a bound on the randomness of the Receiver.

Theorem 4.4 *In any correct and private one-round (k, m)-DOT-$\binom{n}{1}$ scheme it results*

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

Proof. Let $X = \{j_1, \ldots, j_k\} \subseteq \{1, \ldots, m\}$ be a subset of k indices. First notice that, from (15) of Appendix A,

$$\begin{aligned} I(\mathbf{R}; \mathbf{Q}_X|\mathbf{T}) &= H(\mathbf{R}|\mathbf{T}) - H(\mathbf{R}|\mathbf{Q}_X, \mathbf{T}) \\ &= H(\mathbf{Q}_X|\mathbf{T}) - H(\mathbf{Q}_X|\mathbf{R}, \mathbf{T}). \end{aligned}$$

Since, from (17) and (21) of Appendix A and condition (8), we get $H(\mathbf{Q}_X|\mathbf{R}, \mathbf{T}) = 0$, and, from (14) of Appendix A, we get $H(\mathbf{R}|\mathbf{Q}_X, \mathbf{T}) \geq 0$, it follows that

$$H(\mathbf{R}|\mathbf{T}) \geq H(\mathbf{Q}_X|\mathbf{T}). \tag{11}$$

Moreover, from (21) and (17) of Appendix A we get that

$$H(\mathbf{Q}_X|\mathbf{T}) \geq H(\mathbf{Q}_{X\setminus\{j_k\}}|\mathbf{T}) = \sum_{\ell=1}^{k-1} H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{j_1}, \ldots, \mathbf{Q}_{j_{\ell-1}}, \mathbf{T}). \tag{12}$$

On the other hand, from (15) of Appendix A, denoting by $Y_\ell = \{j_1, \ldots, j_{\ell-1}\}$, we get

$$\begin{aligned} I(\mathbf{T}; \mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}) &= H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}) - H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}, \mathbf{T}) \\ &= H(\mathbf{T}|\mathbf{Q}_{Y_\ell}) - H(\mathbf{T}|\mathbf{Q}_{Y_{\ell+1}}). \end{aligned}$$

From which it follows that

$$H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell},\mathbf{T}) = H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}) + H(\mathbf{T}|\mathbf{Q}_{Y_\ell}) - H(\mathbf{T}|\mathbf{Q}_{Y_{\ell+1}}).$$

Moreover, in any one-round (k,m)-DOT-$\binom{n}{1}$, from condition (3) of Definition 2.1, for $\ell = 1,\ldots,k$, it follows that $H(\mathbf{T}|\mathbf{Q}_{Y_\ell}) = H(\mathbf{T})$. Therefore, for each $\ell = 1,\ldots,k-1$, $H(\mathbf{T}|\mathbf{Q}_{Y_\ell}) = H(\mathbf{T}|\mathbf{Q}_{Y_{\ell+1}}) = H(\mathbf{T})$. Hence,

$$H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell},\mathbf{T}) = H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}). \tag{13}$$

Now we claim that, for $\ell = 1,\ldots,k$, $H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}) \geq H(\mathbf{T})$. Indeed, properties (17) and (21) of Appendix A imply that

$$\begin{aligned} H(\mathbf{T}) &= H(\mathbf{T}|\mathbf{Q}_{Y_k}) \\ &\leq H(\mathbf{T},\mathbf{Q}_{j_k}|\mathbf{Q}_{Y_k}) \\ &= H(\mathbf{Q}_{j_k}|\mathbf{Q}_{Y_k}) + H(\mathbf{T}|\mathbf{Q}_{Y_{k+1}}) \\ &= H(\mathbf{Q}_{j_k}|\mathbf{Q}_{Y_k}), \end{aligned}$$

where the last inequality holds due to Lemma 10, which states that $H(\mathbf{T}|\mathbf{Q}_{Y_{k+1}}) = 0$. Therefore, from (11), (12) and (13), it results

$$H(\mathbf{R}|\mathbf{T}) \geq \sum_{\ell=1}^{k-1} H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell},\mathbf{T}) \geq \sum_{\ell=1}^{k-1} H(\mathbf{Q}_{j_\ell}|\mathbf{Q}_{Y_\ell}) \geq (k-1)H(\mathbf{T}).$$

Hence, applying (16) of Appendix A, we get

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

■

TIGHTNESS OF THE BOUND. Notice that the above lower bound is tight. Indeed, the (one-round) combinatorial constructions we will present in the following section, meet the bound by equality.

5 One-Round Protocols Based on Polynomial Interpolation

Two one-round protocols for (k,m)-DOT-$\binom{2}{1}$ have been proposed[6] in [34]. The first one uses *sparse* bivariate polynomials. The second one uses *fully* bivariate polynomials. Both constructions of (k,m)-DOT-$\binom{2}{1}$ use, as a building block, a sub-protocol from which a cheating Receiver can infer at most a linear combination of the secrets held by the Sender. The (k,m)-DOT-$\binom{2}{1}$ protocol is then obtained by composing in a certain way multiple instances of the sub-protocol. The general structure of the subprotocol is given in Table 1.

In this section we describe one-round (k,m)-DOT-$\binom{n}{1}$ oblivious transfer protocols, which generalize the one-round (k,m)-DOT-$\binom{2}{1}$ protocols proposed in [34]. In Table 2 we describe

[6] Notice that, a (k,m)-DOT-$\binom{2}{1}$ can be used as a black box to set up "more complex" oblivious transfer protocols in the same distributed model (see [23, 19, 11] for unconditionally secure reductions). In this case, any improvement in the design of the available (k,m)-DOT-$\binom{2}{1}$, implies directly an improvement of the performance of the more complex protocols.

Structure of the sub-protocols used in the design of one-round (k,m)-DOT-$\binom{2}{1}$ in [34].
Let $s_0, s_1 \in Z_p$ be $\mathcal{S}$'s secrets, and let $i \in \{0,1\}$ be $\mathcal{R}$'s index.

Set-up Phase.

- $\mathcal{S}$ generates a bivariate polynomial $Q(x,y)$ with values in Z_p such that $Q(0,0) = s_0$, and $Q(0,1) = s_1$.
- Then, for $j = 1, \ldots, m$, $\mathcal{S}$ sends the univariate polynomial $Q(j, \cdot)$ to server $\mathcal{S}_j$.

Oblivious Transfer Phase.

- $\mathcal{R}$ chooses a random polynomial Z such that $Z(0) = i$, and defines a univariate polynomial V to be $V(x) = Q(x, Z(x))$ such that the degree of V is $k-1$.
- Then, for $j = 1, \ldots, m$, the receiver $\mathcal{R}$ sends server $\mathcal{S}_j$ the value $Z(j)$, and receives from $\mathcal{S}_j$ the value $V(j) = Q(j, Z(j))$.
- After receiving k values of V, the receiver $\mathcal{R}$ interpolates V and computes $V(0)$.

Table 1: The structure of the sub-protocol

the sub-protocol used in the design of the one-round (k,m)-DOT-$\binom{n}{1}$ oblivious transfer protocol, which generalizes the first one-round (k,m)-DOT-$\binom{2}{1}$ protocol. Then, we show how to compose such a sub-protocol in order to set up a (k,m)-DOT-$\binom{n}{1}$, and we exhibit a proof of correctness and privacy of the overall construction. Along the same line, in Table 5, we describe the sub-protocol used in the design of a *t-private* one-round oblivious transfer protocol, we will define later on, which generalizes the second distributed oblivious transfer protocol. Due to the similarity of the strategy to construct such a scheme with the strategy to set up the first one, we will only sketch the description of the full scheme, resulting by composing multiple instances of the subprotocol.

Let us start by analizing the sub-protocol to set up a (k,m)-DOT-$\binom{n}{1}$ (see Table 2).

From the above description, it is easy to see that condition (8) is satisfied. Indeed, every query Q_j is uniquely determined by the index of the secret chosen by the Receiver and her own program, as well as every answer A_j is uniquely determined by the query Q_j and the program P_j held by S_j.

Correctness. We show that the sub-protocol given in Table 2 satisfies Definition 4.1.
For $\ell = 1, \ldots, n-1$, let $Z_{y_\ell}(x) = \sum_{j=0}^{k-1} s_{y_\ell}^j x^j$ be the polynomial generated by $\mathcal{R}$, where $s_{y_\ell}^j$, for $j = 1, \ldots, k-1$ are random values. 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

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

which can be re-arranged as

A Sub-Protocol for one-round (k,m)-DOT-$\binom{n}{1}$.
Let $s_0, s_1, \ldots, s_{n-1} \in Z_p$ be $\mathcal{S}$'s secrets, and let $i \in \{0, \ldots, n-1\}$ be $\mathcal{R}$'s index.

Set-up Phase.

- $\mathcal{S}$ generates $k-1$ random values $a_1, \ldots, a_{k-1} \in Z_p$ and sets up an n-variate polynomial

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

  where $b_0 = s_0$ and $b_i = s_i - b_0$, for $i = 1, \ldots, n-1$.
  It follows that $Q(0,0,\ldots,0) = s_0, Q(0,1,0,\ldots,0) = s_1, \ldots, Q(0,0,\ldots,1) = s_{n-1}$.
- Then, for $j = 1, \ldots, m$, $\mathcal{S}$ sends the $(n-1)$-variate polynomial $Q(j, y_1, \ldots, y_{n-1})$ to server $\mathcal{S}_j$.

Oblivious Transfer Phase.

- $\mathcal{R}$ 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 if $\mathcal{R}$ is interested in s_0, i.e., $i = 0$, or an $(n-1)$-tuple of zeroes and a single 1 in position i, if $\mathcal{R}$ is interested in s_i, i.e., $i \in \{1, \ldots, n-1\}$.
- Then, $\mathcal{R}$ sends server $\mathcal{S}_j$ the values $Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j)$ and receives the value $V(j) = Q(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$.
- After receiving k values, say $V(j_1), \ldots, V(j_k)$, the receiver $\mathcal{R}$ interpolates a univariate polynomial $V(x) = Q(x, Z_{y_1}(x), \ldots, Z_{y_{n-1}}(x))$ of degree $k-1$, and computes $V(0)$.

Table 2: A sub-protocol for one-round (k,m)-DOT-$\binom{n}{1}$.

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

For $x = 0$, the polynomial becomes $b_0 + b_1 s_{y_1}^0 + \cdots + b_{n-1} s_{y_{n-1}}^0$. If $(s_{y_1}^0, \ldots, s_{y_{n-1}}^0) = (0, \ldots, 0)$, then $V(0) = b_0 = s_0$. On the other hand, if $(s_{y_1}^0, \ldots, s_{y_{n-1}}^0) = (0, \ldots, 1, \ldots, 0)$ where 1 is in position i, for a certain $i \in \{1, \ldots, n-1\}$), then $V(0) = b_i + b_0 = s_i$. Therefore, $V(0)$ is exactly the desired secret, that is W_T.

Privacy. About the privacy property, stated by Definition 2.2, notice that:

- Condition (3) is satisfied due to the degree of the polynomials chosen by the Receiver; Indeed, a coalition of $k-1$ servers gets, only $k-1$ points of the polynomial $Z_{y_j}(x)$, for each $j = 1, \ldots, n-1$. Therefore, for any possible choice of $Z_{y_j}(0) \in Z_p$, the coalition interpolates a different polynomial $Z_{y_j}(x)$ of degree k, which agrees with the $k-1$ received values. Hence, the $k-1$ points do not give any information about the first coefficient of $Z_{y_j}(x)$. As a consequence, the choice of the receiver private.

- Condition (5) follows from the same observation: $k-1$ values do not give any information about $V(0)$, since $V(x)$ has degree $k-1$ and, for any possible value of $V(0) \in Z_p$, the Receiver interpolates a different polynomial which agrees with the $k-1$ received values. Hence, the $k-1$ values do not give information about the secrets.
- Condition (4) is *not* satisfied. Indeed, it is possible to show that in the protocol given in Table 2 the Receiver can learn a linear combination of the secrets. Indeed, if the Receiver does not follow the protocol and chooses certain values $(Z_{y_1}(0), \ldots, Z_{y_{n-1}}(0))$, say for example $(2, 3, \ldots, 1)$, then she gets a linear combination of the secrets $s_0, \ldots, s_{n-1}$.

Notice that, in [34], for the case of two secrets, a proof that the Receiver can get *no more than a single* linear combination by running the sub-protocol described in Table 2 with k servers was given. It is not difficult to show that the proof easily generalises to the case of n secrets (we refer the interested reader to [34].) Such a result can be used to construct a (k,m)-DOT-$\binom{n}{1}$, forcing the Receiver to get *at most one of the secret* held by the Sender and no joint information about the secrets, by using *multiple instances* of the sub-protocol. More precisely, the sub-protocol given in Table 2, can be used as a building block to set up a (k,m)-DOT-$\binom{n}{1}$. The scheme is given in Tables 3 and 4.

A Protocol for (k,m)-DOT-$\binom{n}{1}$.
Let $s_0, s_1, \ldots, s_{n-1} \in Z_p$ be $\mathcal{S}$'s secrets, and let $i \in \{0, \ldots, n-1\}$ be $\mathcal{R}$'s index.

Set-up Phase.

- The Sender, sets up n polynomials

$$Q^\ell(x, y_1, \ldots, y_{n-1}) = P^\ell(x) + P^\ell(y_1, \ldots, y_{n-1}),$$

  in the following way: For $\ell = 1, \ldots, n$, the polynomial $P^\ell(x)$ is defined by $P^\ell(x) = \sum_{j=1}^{k-1} a_j^\ell x^j$, where $a_1^\ell, \ldots, a_k^\ell$ are randomly chosen in Z_p. Hence, it holds that $P^\ell(0) = 0$. Then, denoting by $a_1, a_2, \ldots, a_{n-1}, b$ randomly chosen values in Z_p, the polynomials $P^1(y_1, \ldots, y_{n-1})$, ..., $P^n(y_1, \ldots, y_{n-1})$ are given by:

$$\begin{aligned} P^1(y_1, \ldots, y_{n-1}) &= (a_1 s_1 - b s_0) y_1 + \ldots + (a_{n-1} s_{n-1} - b s_0) y_{n-1} + b s_0 \\ P^2(y_1, \ldots, y_{n-1}) &= (a_1 - b) y_1 + b \\ &\vdots \\ P^n(y_1, \ldots, y_{n-1}) &= (a_{n-1} - b) y_{n-1} + b \end{aligned}$$

- Once the above polynomials have been set up, every server $\mathcal{S}_j$ receives from $\mathcal{S}$, for $\ell = 1, \ldots, n$, the polynomial $Q^\ell(j, y_1, \ldots, y_{n-1})$. Such a distribution ends the Set-up Phase.

Table 3: A one-round (k,m)-DOT-$\binom{n}{1}$ scheme: Set-up.

We show that the protocol given in Tables 3 and 4 implements a (k,m)-DOT-$\binom{n}{1}$.

CORRECTENESS. From the description of the Oblivious Transfer Phase, it is easy to see that Definition 4.1 is satisfied. Indeed, the correctness of the sub-protocol guarantees that the

A Protocol for (k,m)-DOT-$\binom{n}{1}$.

Oblivious Transfer Phase.

- Let X be a subset of k servers. The Receiver sends to each server in X the same queries described in Table 2, that is, $\mathcal{R}$ sends server $\mathcal{S}_j \in X$ the values $Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j)$, but receives, from each of the k servers, n points, e.g., from $\mathcal{S}_j$ she gets $Q^\ell(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$, for $\ell = 1, \ldots, n$.
- If $\mathcal{R}$ has sent queries to recover s_0, then she can compute bs_0 by interpolating the k values $Q^1(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$, for $j \in X$, and b by interpolating one of the sequences, defined by $\ell \in \{1, \ldots, n-1\}$, of k values $Q^\ell(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$, for each $j \in X$. Similarly, if $\mathcal{R}$ has sent queries to recover s_i, for $i \in \{1, \ldots, n-1\}$, then she can compute $(a_i s_i - bs_0) + bs_0 = a_i s_i$ by interpolating the sequence of k values $Q^1(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$ for each $j \in X$, and b by interpolating one of the sequences, defined by $\ell \in \{1, \ldots, n-1\}$, of k values $Q^\ell(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$, for each $j \in X$. In both cases, a simple division in Z_p, i.e., $a_i s_i / a_i$ or bs_0/b, yields the desired secret.

Table 4: A one-round (k,m)-DOT-$\binom{n}{1}$ scheme: Oblivious Transfer.

Receiver gets a masked secret and the mask itsself. Then, a simple computation (i.e. division in Z_p) enables recovering the secret by removing the mask.

Privacy. The privacy property, stated by Definition 2.2, can be shown as follows:

- Condition (3) and Condition (5) follow exactly from the same argument we have applied discussing the protocol given in Table 2. Actually, notice that we are basically repeating n times the protocol of Table 2, with the constraint that the Receiver sends a *single* sequence of queries instead of n distinct sequences of queries.
- Condition (4) can be shown by noticing that if the Receiver cheats by sending a disallowed sequence of queries, in order to learn information about more than one secret, she gets some linear combinations $\gamma_1, \ldots, \gamma_n$ of $s_0, \ldots, s_{n-1}$, from which she does not learn anything about the secrets. More precisely, denoting by $\alpha_1, \ldots, \alpha_{n-1}, \beta$ coefficients that depend on the specific queries sent by the dishonest Receiver, the linear combinations she gets are given by:

$$
\begin{aligned}
\gamma_1 &= \alpha_1(a_1 s_1 - bs_0) + \ldots + \alpha_{n-1}(a_{n-1}s_{n-1} - bs_0) + \beta bs_0 \\
\gamma_2 &= \alpha_1(a_1 - b) + \beta b \\
&\ \vdots \\
\gamma_n &= \alpha_{n-1}(a_{n-1} - b) + \beta b
\end{aligned}
$$

which can be expressed as a product as follows:

$$\begin{bmatrix} \alpha_1 s_1 & \alpha_2 s_2 & \cdots & \alpha_{n-1} s_{n-1} & s_0(\beta - (\alpha_1 + \cdots + \alpha_{n-1})) \\ \alpha_1 & 0 & \cdots & 0 & \beta \\ 0 & \alpha_2 & & & \beta \\ \vdots & 0 & \cdots & \vdots & \vdots \\ & \vdots & & 0 & \\ 0 & \cdots & 0 & \alpha_{n-1} & \beta \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \\ \vdots \\ \\ a_{n-1} \\ b \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \\ \vdots \\ \\ \\ \gamma_n \end{bmatrix}$$

We have to prove that the receiver gets no information about $s_0, \ldots, s_{n-1}$. Let us assume that the secrets s_i are all distinct (it is enough to pad each secret with a unique identifier). In such a case, the above matrix is singular only if some α_j is equal to zero. Let us consider two cases: If $\alpha_j \neq 0$ for each $j = 1, \ldots, m$, then, for any random choice of the values $[a_1, a_2, \cdots, a_{n-1}, b]$, performed by the Sender, independently of the queries sent by the Receiver, a different product $[\gamma_1, \ldots, \gamma_n]$ is defined. Therefore, $[\gamma_1, \ldots, \gamma_n]$ does not give any information about $s_0, \ldots, s_{n-1}$. On the other hand, if for some j the coefficient $\alpha_j = 0$, then the product vector does not depend on s_j. Hence, surely the Receiver does not get any information about s_j. Moreover, the part of the product vector from which the Receiver can get information about the other $n-1$ secrets, can be seen as the product of an $(n-1) \times (n-1)$ matrix, and the $(n-1)$ random values $[a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_{n-1}, b]$ (Just delete column and row corresponding to $\alpha_j = 0$ in the above matrix). But, since this $(n-1) \times (n-1)$ submatrix has rank $n-1$, then $[\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_n]$ does not give any information about $\{s_1, \ldots, s_n\} \setminus \{s_j\}$. The same argument can be applied if more than one coefficient α_j is equal to zero. Hence, the privacy condition holds.

Remark. The above protocol meets the bound on the randomness needed to set up a correct and private DOT scheme. Indeed, in order to set up the scheme, the Sender needs $(k-1)n + n = kn$ random values in Z_p to construct the polynomials $Q^\ell(x, y_1, \ldots, y_{n-1})$. Hence, $H(\mathbf{P}_1, \ldots, \mathbf{P}_m) = kH(\mathbf{W}) = kn \log p$.

Generalising the second construction, which uses fully n-variate polynomials, we can set up a sort of DOT protocol in which conditions (3) and (5) of Definition 2.2 hold against a subset of servers X, such that $|X| = t < k-1$ servers, and in which condition (6) of Definition 2.3 is satisfied with respect to a coalition among the Receiver and a subset of servers X, such that $|X| = \ell < k - t$. We refer to such a protocol as to a t-private one-round (k, m)-DOT-$\binom{n}{1}$. In Table 5 we describe the sub-protocol that can be used to set up a t-private one-round (k, m)-DOT-$\binom{n}{1}$, satisfying the mentioned properties.

Correctness. Definition 4.1 is satisfied. Indeed, denoting as before by $Z_{y_j}(x) = \sum_{r=0}^{k-1} s_{y_j}^r x^r$ the polynomials generated by $\mathcal{R}$, 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 as

$$\sum_{j=0}^{d_x} \sum_{\ell_1=0}^{d_y} \cdots \sum_{\ell_{n-1}=0}^{d_y} a_{j,\ell_1,\ldots,\ell_{n-1}} x^j (s_{y_1}^0 + \sum_{j=1}^{d_z} s_{y_1}^j x^j)^{\ell_1} \cdots (s_{y_{n-1}}^0 + \sum_{j=1}^{d_z} s_{y_{n-1}}^j x^j)^{\ell_{n-1}}$$

A sub-protocol for t-private one-round (k,m)-DOT-$\binom{n}{1}$.
Let $s_0, s_1, \ldots, s_{n-1} \in Z_p$ be $\mathcal{S}$'s secrets, and let $i \in \{0, \ldots, n-1\}$ be $\mathcal{R}$'s index.

Set-up Phase.

- $\mathcal{S}$ generates an n-variate polynomial with values in Z_p

$$Q(x, y_1, \ldots, y_{n-1}) = \sum_{j=0}^{d_x} \sum_{\ell_1=0}^{d_y} \cdots \sum_{\ell_{n-1}=0}^{d_y} a_j x^j y_1^{\ell_1} \cdots y_{n-1}^{\ell_{n-1}}$$

  where $a_{0\ldots0} = s_0$ and, for $t = i, \ldots, n-1$, $\sum_{\ell_i=0}^{d_y} a_{0\ldots\ell_i\ldots0} = s_i$. Notice that, $Q(0,0,\ldots,0) = s_0, Q(0,1,0,\ldots,0) = s_1, \ldots, Q(0,0,\ldots,1) = s_{n-1}$.

- Then, for $j = 1, \ldots, m$, $\mathcal{S}$ sends the $(n-1)$-variate polynomial $Q(j, y_1, \ldots, y_{n-1})$ to server $\mathcal{S}_j$.

Oblivious Transfer Phase.

- $\mathcal{R}$ chooses $n-1$ random polynomials $Z_{y_1}(x) \ldots, Z_{y_{n-1}}(x)$ of degree d_z such that $(Z_{y_1}(0), \ldots, Z_{y_{n-1}}(0))$ is an $(n-1)$-tuple of zeroes if $\mathcal{R}$ is interested in s_0 or an $(n-1)$-tuple of zeroes and a single 1 in position i, if $\mathcal{R}$ is interested in s_i. The degree d_x, d_y of the polynomial $Q(x, y_1, \ldots, y_{n-1})$, and d_z of the polynomials $Z_{y_j}(x)$, must be such that $d_x + d_z d_y(n-1) = k-1$.

- Then, $\mathcal{R}$ sends server $\mathcal{S}_j$ the values $Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j)$ and receives the value $V(j) = Q(j, Z_{y_1}(j), \ldots, Z_{y_{n-1}}(j))$.

- After receiving k values, say $V(j_1), \ldots V(j_k)$, the receiver $\mathcal{R}$ interpolates a univariate polynomial $V(x) = Q(x, Z_{y_1}(x), \ldots, Z_{y_{n-1}}(x))$ of degree $k-1$, and computes $V(0)$.

Table 5: A sub-protocol for one-round t-private (k,m)-DOT-$\binom{n}{1}$.

Therefore, $V(0)$ is exactly

$$\sum_{j=0}^{d_x} \sum_{\ell_1=0}^{d_y} \cdots \sum_{\ell_{n-1}=0}^{d_y} a_{j,\ell_1,\ldots,\ell_{n-1}} (s_{y_1}^0)^{\ell_1} \cdots (s_{y_{n-1}}^0)^{\ell_{n-1}} = \sum_{\ell_i=0}^{d_y} a_{0,0,\ldots,\ell_i\ldots,0}$$

Privacy. Along the same line of the proof given for the sub-protocol described in Table 2, we can show that, *with respect to a threshold* $t < k-1$, conditions (3) and (5) of Definition 2.2 are satisfied, and condition (6) of Definition 2.3 is also satisfied, with respect to a coalition of $\ell \leq k-t$ servers and the Receiver. Notice that the thresholds t and ℓ depend on the particular choices of d_x, d_y, d_z and k. However, the protocol given in Table 5 does not satisfies condition (4) of Definition 2.2, but it guarantees that the Receiver can learn at most a linear combination of the secrets. The same strategy applied in Tables 3 and 4 can be used to set up a t-private one-round (k,m)-DOT-$\binom{n}{1}$. Actually notice that, when the number of secrets n is equal to 2, the sub-protocols given in Table 2 and Table 5 become *exactly* the sub-protocols given in [34], and the techniques we have used to develop the general case are an extension of the ones therein presented.

6 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. 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 4.4. However, they are not so efficient in terms of server memory storage and communication complexity.

6.1 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. In order to provide an intuition of the ideas underlying the following protocols, we start by looking at an example of a one-round $(2,2)$-DOT-$\binom{3}{1}$.

Set-up Phase. Assume that the three secrets held by the Sender are s_0, s_1 and s_2. The Sender constructs a 2×3^2 matrix A with values chosen in Z_p

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

in such a way that the sum of the values $\bmod p$ of every column is one of the secrets. Therefore, we say that every column of A *hides* a secret. The rule used to hide a secret by means of a column is depicted in Table 6.

Index column	0	1	2	3	4	5	6	7	8
Representation in base 3	00	01	02	10	11	12	20	21	22
Corresponding secret index	0	1	2	1	2	0	2	0	1

Table 6: Correspondence secret-column

The indices of the columns of A, i.e., $j \in \{0,1,\ldots,8\}$, represented in base 10 are given in the first row. The representations of such indices in base 3 are given in the second. The third row contains the sum $\bmod 3$ of the digits of the representations in base 3. For example, column 0 has representation in base 3 equal to 00. Hence, $0+0 \bmod 3 = 0$, which is the value reported in the third row. Column 1 has representation 01. Hence, $0+1 \bmod 3 = 1$. Column 5 has representation 12. Hence, $1+2 \bmod 3 = 0$, and so on. In general, column $j \in \{0,1,\ldots,8\}$, whose representation in base 3 is given by $c_1^j c_2^j$, hides the secret s_i, for $i \in \{0,1,2\}$, *if and only if* $c_1^j + c_2^j \bmod 3 = i$. Therefore, it is easy to check that columns $0, 5$ and 7 hide s_0, columns $1, 3$ and 8 hide s_1, and columns $2, 4$ and 6 hide s_2. Once the matrix A has been set up, the Sender sends the first row of A to S_1 and the second to S_2.

Oblivious Transfer Phase. To recover a secret among s_0, s_1, s_2, let us say s_1, the Receiver chooses one of the columns which hides s_1, for example column 3, and sends $c_1^3 = 1$ to S_1 and $c_2^3 = 0$ to S_2. Every server sends to the Receiver a subset of the values of his own row. More precisely, S_1 compares 1 with the first digit c_1^j of the representation in base 3 of column j, for $j \in \{0,1,\ldots,8\}$. If they are equal, then S_1 sends the value $a_{1,j}$. Server S_2 does the same by comparing 0 with the second digit c_2^j of the representation in base 3 of column

j, for $j \in \{0, 1, \ldots, 8\}$. Therefore, the Receiver gets the values $a_{1,3}, a_{1,4}, a_{1,5}$ from S_1 and $a_{2,0}, a_{2,3}, a_{2,6}$ from S_2, and computes $s_1 = a_{1,3} + a_{2,3} \bmod p$. ■

The reader can check by hand that the above example yields a one-round $(2,2)$-DOT-$\binom{3}{1}$. The protocol for the general case is given in Table 7:

A One-Round (k,k)-DOT-$\binom{n}{1}$ Construction.
Let $s_0, s_1, \ldots, s_{n-1} \in Z_p$ be $\mathcal{S}$'s secrets and let $i \in \{1, \ldots, n\}$ be $\mathcal{R}$'s index.

Set-up Phase

- $\mathcal{S}$ sets up a $k \times n^k$ matrix A of random values in Z_p as follows: for $j \in \{0, \ldots, n^k - 1\}$, the sum of the values of column $A[\cdot, j]$ is equal to s_h if, denoting by $c_1^j \cdots c_k^j$ the representation in base n of j, then $\sum_{\ell=1}^{k} c_\ell^j \bmod n = h$.
- $\mathcal{S}$, for $q = 1, \ldots, k$, sends the q-th row $A[q, \cdot]$ to the server S_q.

Oblivious Transfer Phase

- $\mathcal{R}$ chooses a value $j \in \{0, \ldots, n^k - 1\}$ such that $\sum_{\ell=1}^{k} c_\ell^j \bmod n = i$ and, for $q = 1, \ldots, k$, she sends the digit c_q^j to server S_q.
- Server S_q, for $\ell = 0, \ldots, n^k - 1$, sends to the Receiver the value $A[q, \ell]$ if and only if the q-th digit of the n-ary representation of ℓ is equal to c_q^j.
- $\mathcal{R}$ sums up the values $A[1, j], \ldots, A[k, j]$, recovering the secret, that is $s_i = \sum_{i=1}^{k} A[i, j] \bmod p$.

Table 7: A One-Round (k,k)-DOT-$\binom{n}{1}$ Construction

We show that the above protocol implements a One-Round (k,k)-DOT-$\binom{n}{1}$.

Correctness. Definition 4.1 is satisfied since, once $\mathcal{R}$ has chosen column j, whose n-ary representation is $c_1^j \ldots c_k^j$ and has sent, for $q = 1, \ldots, k$ the digit c_q^j to server S_q, among other values, certainly she receives back $A[1, j], \ldots, A[k, j]$. Hence, $\mathcal{R}$ can compute s_i as $\sum_{i=1}^{k} A[i, j] \bmod p$.

Privacy. The privacy property, stated by Definition 2.2, can be shown as follows:

- Condition (3) is satisfied: a coalition of $k-1$ servers, say $S_1, \ldots, S_{k-1}$, contacted by $\mathcal{R}$, cannot infer in which secret she is interested. Indeed, assume that column j, chosen by $\mathcal{R}$ to recover the secret, has n-ary representation $c_1^j \ldots c_k^j$. Then $S_1, \ldots, S_{k-1}$, receive *only* $c_1^j \ldots c_{k-1}^j$ from $\mathcal{R}$. Since the index i of the secret s_i, hidden by column j, is given by $\sum_{\ell=1}^{k} c_\ell^j \bmod n$, for any index $i \in \{0, \ldots, n-1\}$, there is exactly one value c_k^j such that $\sum_{\ell=1}^{k-1} c_\ell^j + c_j^k \bmod n = i$. Hence, the coalition has absolutely no information about the index i.
- Condition (4) holds because the Receiver $\mathcal{R}$ can recover at most one secret. Indeed, assume that $\mathcal{R}$ recovers two secrets, say s_1 and s_2 from columns j_1 and j_2. Then, she must have received, for $q = 1, \ldots, k$, the values $A[q, j_1]$ and $A[q, j_2]$ from server S_q. But this means that, denoting by $c_1^{j_1} \ldots c_k^{j_1}$ and $c_1^{j_2} \ldots c_k^{j_2}$ the n-ary representations of j_1 and

j_2, for each $\ell = 1, \ldots, k$, must be $c_\ell^{j_1} = c_\ell^{j_2}$. Since each column has a different n-ary representation, we have a contradiction. Moreover, since all the values of a column are needed to decrypt a secret, and each value is essential to determine the secret, easily follows that $\mathcal{R}$ does not get *any* information at all about the others.

- Condition (5) basically holds for the same reason we have seen discussing Condition (3): The Receiver does not get any information about the secrets during the set-up phase, and a coalition of $k-1$ servers, say again $S_1, \ldots, S_{k-1}$, cannot compute any secret. Indeed, for $j = 0, \ldots, n^k - 1$, they hold $A[1,j], \ldots, A[k-1,j]$. Since for any $i = 0, \ldots, n-1$, secret s_i is hidden by $\sum_{q=1}^{k} A[q,j] \bmod p$ for certain columns $j \in \{0, \ldots, n^k - 1\}$, then for any secret s_i, there is exactly one value $A[k,j]$ such that $\sum_{q=1}^{k-1} A[q,j] + A[q,k] \bmod n = s_i$. Therefore, the coalition has absolutely no information about any secret.

Using some well-known combinatorial structures, we can in a certain way generalize the above construction, in order to set up a (k,m)-DOT-$\binom{n}{1}$. More precisely, let t, q, r and λ be integers 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 of r symbols, say $\{0, 1, \ldots, r-1\}$, such that within any t columns, every possible t-tuple of symbols occurs in exactly λ rows. Using an orthogonal array and threshold secret sharing schemes[7], we can set up a (k,m)-DOT-$\binom{n}{1}$ (see Table 8).

Example. We present a one-round $(2,3)$-DOT-$\binom{3}{1}$ using the protocol described in Table 8.

Set-up Phase. The Sender constructs and publishes the following matrix obtained by transposing an $OA_1(2,4,3)$:

$$I = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 \end{bmatrix}$$

Then, he shares the secret associated with each column by means of an independent copy of a $(2,3)$-threshold secret sharing scheme. Finally, he sends the shares associated to row j to S_j for $j \in \{1, \ldots, 3\}$.

Oblivious Transfer Phase. Let $X = \{2,3\}$ be a 2-subset of $\{1,2,3\}$, denoting servers S_2, S_3. Suppose that $\mathcal{R}$ wishes to recover the secret s_1. Hence, she chooses one of the columns $3, 4, 5$, say $c = 5$, and sends 0 to S_2 and 1 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 $(1, sh_{3,1}), (5, sh_{3,5})$ and $(6, sh_{3,6})$.

Therefore, the Receiver can recover s_1 using $(5, sh_{2,5})$ and $(5, sh_{3,5})$. ■

CORRECTNESS. The protocol satisfies Definition 4.1 since the Receiver, for any secret and for any column she has chosen to retrieve the secret, gets from the contacted servers a sufficient

[7] A (k,m)-threshold secret sharing scheme is a method by means of which a dealer shares a secret among a set of m participants in such a way that: 1) any subset of participants of size greater than or equal to k reconstructs the secret; 2) any subset of size less than k does not get any information about the secret [40].

A One-Round (k, m)-DOT-$\binom{n}{1}$ Construction
Let $s_0, s_1, \ldots, s_{n-1} \in Z_p$ be $\mathcal{S}$'s secrets, and let $i \in \{1, \ldots, n\}$ be $\mathcal{R}$'s index.

Set-up Phase.

- The Sender $\mathcal{S}$ sets up an orthogonal array $OA_1(k, m+1, n)$. We denote the transpose of such an $OA_1(k, m+1, n)$ by I and we assume that it is public. The first row $I[0, \cdot]$ of I establishes "which column hides which secret".
- Then $\mathcal{S}$, for each $c \in \{0, \ldots, n^k - 1\}$, shares the secret $s_{I[0,c]}$ using an (k, m)-threshold secret sharing scheme. Let us denote the shares by $sh_{1,c}, \ldots, sh_{m,c}$. For each column is used a different scheme, i.e., threshold secret sharing schemes are independent.
- Finally, for $j = 1, \ldots, m$, $\mathcal{S}$ sends $sh_{j,0}, \ldots, sh_{j,n^k-1}$ to server S_j.

Oblivious Transfer Phase

- Let $X = \{p_1, \ldots, p_k\}$ be a k-subset of $\{1, \ldots, m\}$. $\mathcal{R}$ chooses a random column c of the matrix I such that $I[0, c] = i$, and, for $1 \in \{1, \ldots, k$, sends the value $y_j = I[p_j, c]$ 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 $I[p_j, d] = y_j$. $\mathcal{R}$ gets n shares from each of the k servers.
- Finally, $\mathcal{R}$ applies the reconstruction function of the threshold secret sharing scheme to $sh_{p_1,c}, \ldots, sh_{p_k,c}$, and she reconstructs the secret s_i.

Table 8: A One-Round (k, m)-DOT-$\binom{n}{1}$ Construction

number of shares. Indeed, assume that to recover s_i she has chosen column c and, without loss of generality, has sent for $j = 1, \ldots, k$, the value $y_j = I[j, c]$ to server S_j. Then, she has received, among others, certainly the k pairs $(1, sh_{1,c}), \ldots, (k, sh_{k,c})$, enabling her to recover s_i.

Privacy. The privacy property, stated by Definition 2.2, can be shown as follows:

- Condition (3) is satisfied: Indeed, assume that $k-1$ servers contacted by $\mathcal{R}$, say $S_1, \ldots, S_{k-1}$, collude in order to figure out the index i. If $\mathcal{R}$ has chosen column c to recover s_i, they have received the values $y_1 = I[1, c], \ldots, y_{k-1} = I[k-1, c]$. It is not difficult to see that, due to the structure of an $OA_1(k, m+1, n)$, this $(k-1)$-tuple appears along *any* k-restriction of the matrix exactly k times, and for each instance, the corresponding value of the k-th row is different. In particular, let us consider the k-restriction defined by rows $0, 1, \ldots, k-1$. Since each instance of the $(k-1)$-tuple $y_1 \ldots y_{k-1}$ is completed *exactly once* with a different value of y_0 in $\{0, \ldots, n-1\}$ (which represents an index of a possible key), the coalition has absolutely no information on the index i.

- Condition (4) holds because the Receiver $\mathcal{R}$ can retrieve at most one secret, due to the structure of an orthogonal array $OA_1(k, m+1, n)$. Indeed, without loss of generality, assume that $\mathcal{R}$ recovers two secrets say s_{i_1} and s_{i_2}, contacting servers $S_1, \ldots, S_k$, from the k-set of shares corresponding to columns c_1 and c_2. This implies that, for $j = 1, \ldots, k$, the value $I[j, c_1] = I[j, c_2]$. Hence, there is a k-restriction of the orthogonal array where a k-tuple appears twice! But, according to the definition of an $OA_1(k, m+1, n)$, this is a contradiction. Furthermore, since less than k shares do not give *any* information on the secret, from the shares corresponding to other columns $\mathcal{R}$ does not get any information on the associated secrets.

- Condition (5) holds because the Receiver does not get any information, and a coalition of $k-1$ servers, say $S_1, \ldots, S_{k-1}$, cannot compute any secret. Indeed, each secret is shared according to a (k, m) threshold secret sharing scheme, and they hold only $k-1$ shares for each of them.

Server memory storage and communication complexity of the combinatorial schemes are quite heavy. The following technique enables us to reduce both resources. Indeed, looking at the protocol described in Table 7 in the particular case of 2 secrets, notice that it is not necessary that *all* the $k \times 2^k$ values $a_{i,j}$ are independent. For example, they can be choosen in such a way that the following relation holds: let j and j' be two different columns of the matrix A whose binary representations are $b_1^j \ldots b_k^j$ and $b_1^{j'} \ldots b_k^{j'}$. Then, for $i = 1, ..., k$, let

$$A[i,j] = A[i,j'] \text{ if and only if } b_1^j = b_1^{j'}, \ldots, b_i^j = b_i^{j'}.$$

Let us consider an example assuming that $k = 3$. For 3 servers, we have:

000	001	010	011	100	101	110	111
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$

The values $a_{i,j}$ can be grouped as

$$a_{1,0} = a_{1,1} = a_{1,2} = a_{1,3}, \text{ and } a_{1,4} = a_{1,5} = a_{1,6} = a_{1,7}$$

for the first row,

$$a_{2,0} = a_{2,1},\ a_{2,2} = a_{2,3},\ a_{2,4} = a_{2,5}, \text{ and } a_{2,6} = a_{2,7}$$

for the second row, and

$$a_{3,0},\ a_{3,1},\ a_{3,2},\ a_{3,3},\ a_{3,4},\ a_{3,5},\ a_{3,6},\ a_{3,7}$$

for the third row. In this example, server S_1 gets 2 values, server S_2 gets 4 values and server S_3 gets 8 values. Hence, memory server storage and communication complexity are reduced. An interesting open problem is to find more efficient representations for the matrix of values.

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[8]. We can pursue the same idea in order to support general access structures.

Let us start by formalizing the notion of a general access structure.

[8] Indeed, notice that also the constructions given in Table 7, can be re-phrased along the same line of the protocol described in Table 8. In this case the orthogonal array used is an $OA_1(k, k+1, n)$.

Definition 6.1 *Let $S = \{S_1, \ldots, S_m\}$ be a set of servers. An access structure $\mathcal{A}$ on S is a collection of subsets $\mathcal{A} \subseteq 2^S \backslash \{\emptyset\}$ such that any $A \in \mathcal{A}$ enables the receiver $\mathcal{R}$ to recover one of the secret.*

To explain the protocol and how to construct the *indexing structure*, let us consider a simple case. Let $S = \{S_1, S_2, S_3, S_4\}$ be a set of 4 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 S. This access structure is well-known in the secret sharing scheme theory and its information rate ρ, which is the maximum ratio between the size of the secret and the size of the share given to the user[9], is equal to $\frac{2}{3}$. Assume that the secret is a pair of values (k_1, k_2) belonging to $Z_{p'} \times Z_{p'}$. The secret can be shared among $\mathcal{P}_3$ as shown in Table 9.

S_1	x	z	
S_2	$k_1 + x \bmod p'$	$k_2 + z \bmod p'$	w
S_3	$k_1 + w \bmod p'$	$k_2 + y \bmod p'$	z
S_4	w	y	

Table 9: A secret sharing scheme for $\mathcal{P}_3$.

The values x, y, z, and w are random values chosen in $Z_{p'}$. The dealer computes the shares given in Table 9 and sends row i to server S_i.

We can construct a $\mathcal{P}_3$-DOT-$\binom{n}{1}$ using this secret sharing scheme as 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 setup using as "rule" to fill in the entries of each column 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 $(i, j) \in Z_3 \times Z_3$. An indexing matrix I can be set up considering 3^4 sharings for each key (i.e., the number of possible choices for x, y, z, and w when seen as elements belonging to Z_3). For example, the restriction of the indexing matrix I to the key (k_1, k_2) indexed by $(1, 2)$ is reported in Table 10:

0	$(\mathbf{1}, \mathbf{2})$			$\cdots$	$(\mathbf{1}, \mathbf{2})$		
1	0	0		$\cdots$	2	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	0		$\cdots$	2	2	

Table 10: Partial view of the indexing matrix I, corresponding to the 3^4 sharings of the secret (k_1, k_2).

In Table 10, each of the 3^4 columns indexed by $(1, 2)$ represents a sharing of $(k_1, k_2) \in Z_{p'} \times Z_{p'}$. We assume that the $3^2 \cdot 3^4$ sharings for the 3^2 secrets are mantained in another (corresponding) matrix A: more precisely, each column of A contains a sharing of a certain key (k_i, k_j).

The Receiver can choose one of the column of I and can ask a subset $B \in \mathcal{P}_3$ to receive the shares in A, whose indices match the entries of the column of the matrix I, in correspondence

[9]The reader is referred to [42] for background on secret sharing schemes.

of the servers in B. In our example, the Receiver, to retrieve (k_1, k_2), can choose the first column of I and can send $(1,2,0)$ to S_3 and $(0,0)$ to S_4, receiving from S_3 all the shares (belonging to the third row of the matrix A) whose indices are $(1,2,0)$ (and, among these, surely $sh^{(1,2)}_{(1,2,0)}$), and from S_4 all the shares (belonging to the fourth row of the matrix A) whose indices are $(0,0)$ (and, among these, surely $sh^{(1,2)}_{(0,0)}$).

It is not difficult to see that the construction is correct, due to the reconstruction property of the secret sharing scheme. In our example $sh^{(1,2)}_{(1,2,0)}$ and $sh^{(1,2)}_{(0,0)}$ enable the Receiver to recover (k_1, k_2). Moreover the scheme is private 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 indexing 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[10]. 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 compute information about any secret, due to the security property of the secret sharing scheme.

Notice that, if we have $n = p^2$ secrets, the construction seen before requires p^4 sharing for each secret, and an indexing matrix I with p^6 columns.

At this point it is not difficult to figure out how the same strategy can be applied to any access structure. We would like just to point out the use of the secret sharing for the construction of *both* the indexing structure I and the sharing of the secrets. Perhaps this design technique can be applied successfully also to other cryptographic protocols. The protocol can be generalized to *arbitrary* access structures $\mathcal{A}$ on the set of servers.

Let $n = p^r$ and, for $i = 1, \ldots, n$, let $s_i \in (Z_{p'})^r$ be $\mathcal{S}$'s secrets. Let each index i be represented by $(i_1, \ldots, i_r) \in (Z_p)^r$. Moreover, let $\mathcal{A}$ be an access structure on the set of the m servers, and let Σ be a secret sharing scheme for $\mathcal{A}$ with information rate $\rho = \frac{r}{\omega}$ represented, as before, in tabular form. Finally, assume that Σ uses, to share a secret, t random values belonging to $Z_{p'}$. The protocol is described in Tables 11 and 12.

In order to show the correctness and the privacy of the protocol it is enough to prove that the indexing structure, given by the matrix I, satisfies the *same properties* of an orthogonal array, the combinatorial structure we have used in the threshold case. Indeed, notice that, when the general access structure considered in the above construction is a threshold structure, a secret sharing scheme with information rate $\rho = 1$ (called *ideal* [42]), i.e., where each share has the same size of the secret, realizing the access structure does exist. As shown by K. Martin [32] and, independently by Dawson et al. [20], if we represent such a secret sharing scheme by means of a distribution table, this table is exactly an orthogonal array. For the general case, a proof that the indexing structure satisfies the same properties of an orthogonal array can be achieved arguing by contradiction: for any fixed set of rows of the matrix I, corresponding to a qualified subset of servers, if two different columns j and j' have the same t-tuples, row by row, then $j = j'$, due to the matrix generation rule. The correctness and privacy properties of the DOT construction easily follow from the same observations we have done analysing the threshold case.

[10] 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 it must be $i = i'$ and $j = j'$.

A One-Round $\mathcal{A}$-DOT-$\binom{n}{1}$ Protocol.

Set-up Phase

- $\mathcal{S}$ sets up a public indexing matrix I, of order $(m+1) \times p^{r+t}$, which *represents*, for each of the p^r secrets, p^t different sharings according to Σ. The matrix I is filled in as follows: the first row says which secret the column hides. More precisely, for $i = 1, \ldots, p^r$ and $j = 1, \ldots, p^t$, the value $I[0, j + (i-1)p^t] = i = (i_1, \ldots, i_r)$. Then, for $q = 1, \ldots, m$, the entry $I[q, j + (i-1)p^t]$ represents the index of the share given to server S_q for the j-th sharing of the i-th secret, and is equal to $(v^j_{q,1}, \ldots, v^j_{q,\ell_i}) \in Z_p^{\ell_i}$ where $\ell_i \leq \omega$, These values are computed using Σ as "rule", and by considering $(i_1, \ldots, i_r)$ (i.e., the representation of i in Z_p^r) *as the secret*, and all possible sequences $(d_1, \ldots, d_t) \in Z_p^t$. These sequences correspond to the p^t random values belonging to $Z_{p'}$, used by Σ to generate the p^t sharings for s_i.
- Once the indexing matrix has been set up, for each secret $s_i \in (Z_{p'})^r$, $\mathcal{S}$ computes the p^t sharings. Let us denote, for the j-th sharing of the secret, by

$$sh^i_{(v^j_{q,1}, \ldots, v^j_{q,\ell_i})}$$

  the share for server S_q according to Σ. We assume that a matrix A, of order $m \times p^{r+t}$, contains in each column one of such sharings.
- For $i = 1, \ldots, n$, for $q = 1, \ldots, m$, and for $j = 1, \ldots, p^t$, $\mathcal{S}$ sends to server S_q the share $sh^i_{v^j_{q,1}, \ldots, v^j_{q,\ell_i}}$, i.e., all the shares belonging to the q-th row of the matrix A.

Table 11: A One-Round $\mathcal{A}$-DOT-$\binom{n}{1}$ Protocol: Set up Phase

6.2 Two-Round Constructions

It is possible to gain in terms of privacy and complexity if we allow *one more round* of interaction between the Receiver and the servers. A simple protocol is described in Table 13. Correctness. The Receiver, once she has received all the values of a column, computes the secret by means of a simple sum.

Privacy. The Privacy property, stated by Definition 2.2, can be shown as follows:

- Condition (3) of Definition 2.2 is satisfied because a coalition of $k-1$ servers cannot infer which secret $\mathcal{R}$ wishes to recover, since they do not know which secret is hidden by which column.
- Condition (4) holds because the Receiver $\mathcal{R}$ can retrieve at most one secret: Indeed, all the values of a single column are needed for computing one of the secrets. Hence, if she gets all values of a column, then she gets no information about the secret hidden by the other column. On the other hand, if she gets values belonging to the two different columns from different servers, then she gets no information on *both* secrets.
- Condition (5) holds because a coalition of $k-1$ servers, say $S_1, \ldots, S_{k-1}$, cannot compute any secret. Indeed, each secret is actually shared according to a (k, k) threshold secret

A One-Round $\mathcal{A}$-DOT-$\binom{n}{1}$ Protocol.

Oblivious Transfer Phase

- $\mathcal{R}$, to recover s_i, chooses a column of I say the g-th one, such that $I[0,g] = (i_1, \dots i_r)$, chooses a subset of servers $B \in \mathcal{A}$ and sends, to each $S_q \in B$, the tuple $y^q = I[q,g]$.
- Each server $S_q \in B$ sends to $\mathcal{R}$, for any column z such that $I[q,z] = y^q$, the pair column-share $(z, A[q,z])$.
- $\mathcal{R}$ reconstructs the secret by using the pairs $(g, A[q,g])$ sent by servers $S_q \in B$.

Table 12: A One-Round $\mathcal{A}$-DOT-$\binom{n}{1}$ Protocol: Oblivious Transfer Phase

sharing scheme, and they hold only $k-1$ shares for each of them. The Receiver cannot help in any way.

It is worthwhile to point out that the two-round construction above described enjoys condition (6) of Definition 2.3, i.e., the further privacy 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 secrets, still cannot compute the other without the help of the last server, due to the sharing of secrets by means of a (k,k) secret sharing scheme.

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 $v_0[j]$ and $v_1[j]$, for $j = 1, \dots, k$. In such a case, one of the servers, say S_j, chooses uniformly at random which of the two values $v_0[j], v_1[j]$ to send $\mathcal{R}$.

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 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_1 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 secrets 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 $r_i \in \{0, \dots, n-1\}$, and, instead of two vectors sharing s_0 and s_1, there are exactly n vectors $v_0, \dots, v_{n-1}$, sharing the secrets $s_0, \dots, s_{n-1}$, respectively. The value $r = \sum r_j \bmod n$ establishes the correspondence between the 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 also this is correct and private.

A Strong (k,k)-DOT-$\binom{2}{1}$ with one addressing bit
Let $s_0, s_1 \in Z_p$ be $\mathcal{S}$'s secrets.

Set-up Phase.

- $\mathcal{S}$ chooses k random bits r_j, and computes the bit r, xoring the r_j's, i.e., $r = \bigotimes_{j=1}^{k} r_j$.
- Moreover, $\mathcal{S}$ sets up two vectors with k entries in Z_p, v_0 and v_1, choosing the first $k-1$ entries at random and computing

$$v_0[k] = s_r - \sum_{j=1}^{k-1} v_0[j] \bmod p, \text{ and } v_1[k] = s_{1-r} - \sum_{j=1}^{k-1} v_1[j] \bmod p.$$

- Then, for $j = 1, \ldots, k$, he sends the bit r_j and the values $v_0[j]$ and $v_1[j]$ to server S_j.

Oblivious Transfer Phase.

- In a first round of communication, $\mathcal{R}$ asks to each server S_j the bit r_j, and computes r. Then, for $j = 1, \ldots, k$, if $\mathcal{R}$ is interested in s_0 and $r = 0$, asks server S_j the value $v_0[j]$; otherwise, if $r = 1$, asks $v_1[j]$. Symmetrically, to recover s_1, if $r = 1$, she asks $v_0[j]$, while if $r = 0$, she asks $v_1[j]$.
- Finally, $\mathcal{R}$ sums up mod$p$ the received values.

Table 13: A Two-Round (k,k)-DOT-$\binom{2}{1}$

7 Information to the Receiver

In this section we slightly modify the model for DOT. We assume that during the setup phase, the Sender $\mathcal{S}$ sends information not only to the m servers but *also to the Receiver* $\mathcal{R}$. Intuitively, by giving information to the Receiver we should be able to achieve a stronger privacy condition. The two-round protocol described in Table 13 (and all its generalizations) can be transformed in a one-round protocol for the new model. Indeed, notice that the random bit each server transmits to the Receiver during the first round, can be eliminated if the sender, during the set up, says to the Receiver which column which secret hides (see Table 14).

Notice that, $k-1$ servers do not have any information about the secrets. At the same time, the Receiver is still not able to gain extra-information about other secrets, apart the one that she recovers honestly. Actually the above protocol is very simple: each secret is shared according to a (k,k) threshold scheme and *only* the Receiver knows which shares correspond to which secret. The generalization of the above protocols to the case of a general access structure on the set of servers and to r secrets can be done along the same line of the two-round protocol without information in set up phase to the Receiver.

A Strong (k,k)-DOT-$\binom{2}{1}$ with information to the Receiver
Let $s_0, s_1 \in Z_p$ be $\mathcal{S}$'s secrets.

Set-up Phase

- $\mathcal{S}$ chooses a random bit, say r.
- Then, $\mathcal{S}$ sets up two vectors with entries in Z_p, v_0 and v_1, choosing the $k-1$ entries at random and computing

$$v_0[k] = s_r - \sum_{j=1}^{k} v_0[j] \bmod q, \text{ and } v_1[k] = s_{1-r} - \sum_{j=1}^{k} v_1[j] \bmod q.$$

- Finally, for $j = 1, \ldots, k$, $\mathcal{S}$ sends the values $v_0[j]$ and $v_1[j]$ to server S_i and the bit r to $\mathcal{R}$.

Oblivious Transfer Phase

- If $\mathcal{R}$ is interested in s_0 and $r = 0$, then, for $j = 1, \ldots, k$, asks to server S_j the value $v_0[j]$; otherwise, if $r = 1$, asks $v_1[j]$. Symmetrically, to recover s_1, if $r = 0$, she asks $v_1[j]$, while if $r = 1$, she asks $v_0[j]$.
- Then, $\mathcal{R}$ sums up mod$q$ the received values v_i with r_i, recovering the secret.

Table 14: One-Round (k,k)-DOT-$\binom{2}{1}$

8 Applications

The protocols described before have several interesting applications and connections with other cryptographic protocols. Let us quickly describe some of them.

Privacy Preserving Auctions and Mechanism Design [36]. The notion of DOT, introduced in [34], was introduced to improve the protocol of [36]. More precisely, in that protocol, there are three parties: an auctioneer, many bidders, and an agency supporting the auction. The auctioneer advertises the auction and its rules. The bidders submit their bids in "encrypted form" to the auctioneer, and the auctioneer, with the help of the agency, can compute the winner of the auction in such a manner that the privacy of the bidders (i.e., non-essential information about their own bids) is preserved. The weak point of the protocol is that if *the auctioneer and the agency collude*, then the privacy of the bids is lost. In order to strengthen the protocol, the agency can be split in two parts: a central agency, that operates only in a set up phase, and m servers, with which the auctioneer communicates in order to compute the auction. In this case the auctioneer needs to collude with k out of the m servers in order to violate the privacy of the bidders. The impossibility result for one-round (k,m) protocols private against a coalition of $k-1$ servers and the Receiver we have shown in Section 4, in this setting means that the highest degree of privacy aimed in [36] with this approach cannot be achieved. On the positive side, the two-round protocols described in Section 6 can be applied to this framework but the *communication pattern* changes and some more details must be taken into account.

Symmetric Private Information Retrieval. Distributed Oblivious Transfer protocols have relations with symmetric private information retrieval schemes [26]. A PIR scheme [13] 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 [26], 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. Therefore, a SPIR Scheme can be seen as a *communication-efficient* 1-out-of-n oblivious transfer scheme. The main differences between the model we have considered 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 $k \leq m$. This property is useful since it guarantees a sort of *Robustness* for the SPIR scheme, in the sense that even if some server crashes, the item can still be retrieved by the user by means of the other available ones. Hence, *a communication-efficient threshold DOT scheme realizes a robust SPIR scheme.* Another important difference is that in information theoretic PIR and SPIR schemes the database is *replicated* among the servers. Hence, every server knows the content. In our model only a k-subset of servers can reconstruct the database.

Another interesting relation of the DOT model we have studied is with information theoretic PIR schemes with preprocessing [2]. The set up phase performed by the dealer can be seen as the preprocessing stage performed by the database owner in [2]. The combinatorial constructions we have shown are communication-inefficient but they require trivial computation for the servers, once the scheme has been set up.

Just to emphasyze the connection, notice that, using the DOT constructions presented in Section 7, we can set up a robust unconditionally secure symmetric private retrieval scheme. The database $\mathcal{D}$ is simply distributed by the owner among m servers, according to the (k, m)-DOT scheme for n secrets of Section 7.

9 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 concept, 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 privacy level in this model can be achieved with, at the same time, a suitable reduction of resources (randomness, memory storage and communication complexity). 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. Several questions and interesting open problems come up from this study. Among others:

- The design of a one-round weak DOT protocol meeting all the bounds given by the information theoretic analysis.
- Techniques to improve the communication complexity of some of the presented schemes with application to SPIR with preprocessing.

- Identification of applicative settings which can benefit from this distributed implementation of the oblivious transfer.

10 Acknowledgement

D. R. Stinson's research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN 203114 – 02.

References

[1] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway, *Locally Random Reductions: Improvements and Applications*, Journal of Cryptology, vol. 10, No.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, LNCS, Springer-Verlag, vol. 1880, pp. 55-73, 2000.

[3] M. Bellare and S. Micali, *Non-interactive Oblivious Transfer and Applications*, Advances in Cryptology: Proceedings of Crypto 1989, LNCS, Springer-Verlag, vol. 435, pp. 547-559, 1990.

[4] G.R. Blakley, *Safeguarding Cryptographic Keys*, Proceedings of AFIPS 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, A. De Santis, and U. Vaccaro, *Randomness in Distribution Protocols*, Information and Computation, vol. 131, no. 2, pp. 111–139, 1996.

[7] C. Blundo, A. De Santis, and U. Vaccaro, *On Secret Sharing Schemes*, Information Processing Letters, No. 65, pp. 25–32, 1998.

[8] 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.

[9] 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.

[10] G. Brassard, C. Crepéau, and J.-M. Roberts, *All-or-Nothing Disclosure of Secrets*, Advances in Cryptology: Proceedings of Crypto 1986, LNCS, Springer-Verlag, vol. 263, pp. 234-238, 1987.

[11] 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.

[12] R.M. Capocelli, A. De Santis, L. Gargano and U. Vaccaro, *On the Size of the Shares in Secret Sharing Schemes*, Advances in Cryptology: Proceedings of Crypto 1991, LNCS, vol. 576, pp. 101–113, 1992.

[13] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, *Private Information Retrieval*, Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS), pp. 41–50, 1995.

[14] T. M. Cover and J. A. Thomas, **Elements of Information Theory**, John Wiley & Sons, 1991.

[15] C. Crepéau, *Equivalence between to flavors of oblivious transfers*, Advances in Cryptology: Proceedings of Crypto 1987, LNCS, Springer Verlag, vol. 293, pp. 350-354, 1988.

[16] 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 1986, LNCS, Springer-Verlag, vol. 263, pp. 239-247, 1987.

[17] G. Di Crescenzo, T. Malkin, and R. Ostrovsky, *Single Database Private Information Retrieval Implies Oblivious Transfer*, Advances in Cryptology: Proceedings of Eurocrypt 2000, LNCS, Springer-Verlag, vol. 1807, pp. 122-138, 2000.

[18] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky, *Universal Service-Providers for Database private Information Retrieval*, Proceedings of 17th Annual ACM Symposium on Principles of Distributed Computing (PODC), 1998.

[19] P. D'Arco and D.R. Stinson, *Generalized Zig-zag Functions and Oblivious Transfer Reductions*, Selected Areas in Cryptography SAC 2001, LNCS, Springer Verlag, vol. 2259, pp. 87-103, 2001.

[20] E. Dawson, E. S. Mahmoodian and A. Rahilly, *Orthogonal arrays and ordered threshold schemes*, Australasian Journal of Combinatorics, No. 8, pp. 27–44, 1993.

[21] A. De Santis and G. Persiano, *Public-Randomness in Public Key Cryptography*, Advances in Cryptology: Proceedings of Eurocrypt 1990, LNCS, Springer-Verlag, vol. 437, pp. 46–62, 1990.

[22] A. De Santis, G. Di Crescenzo, and G. Persiano, *Zero-Knowledge Arguments and Public Key Cryptography*, Information and Computation, vol. 121, no. 1, pp. 23-40, 1995.

[23] Y. Dodis and S. Micali, *Lower Bounds for Oblivious Transfer Reduction*, Advances in Cryptology: Proceedings of Eurocrypt 1999, LNCS, Springer Verlag, vol. 1592, pp. 42-54, 1999.

[24] S. Even, O. Goldreich, and A. Lempel, *A Randomized Protocol for Signing Contracts*, Communications of the ACM, Vol. 28, pp. 637-647, 1985.

[25] 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.

[26] 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), pp. 151-160, 1998.

[27] 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*, Proceedings of RANDOM 1998, LNCS, Springer Verlag, vol. 1518, pp. 200-217, 1998.

[28] 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.

[29] 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.

[30] J. Kilian, *Founding Cryptography on Oblivious Transfer*, Proceedings of the 20th Annual Symposium on Theory of Computing, pp. 20-31, 1988.

[31] Knuth D. E. and Yao A. C., *The Complexity of Nonuniform Random Number Generation*, Algorithms and Complexity, Academic Press, pp. 357–428, 1976.

[32] K. Martin, *Discrete Structures in the Theory of Secret Sharing*, PhD Thesis, University of London, 1991.

[33] M. Naor and B. Pinkas, *Computationally Secure Oblivious Transfer*, Advances in Cryptology: Proceedings of Crypto 1999, LNCS, Springer-Verlag, vol. 1666, pp. 205-219, 2000.

[34] M. Naor and B. Pinkas, *Distributed Oblivious Transfer*, Advances in Cryptology: Proceedings of Asiacrypt 2000, LNCS, Springer-Verlag, vol. 1976, pp. 205-219, 2000.

[35] M. Naor and B. Pinkas, *Efficient Oblivious Transfer Protocols*, Proceedings of SODA 2001, pp. 448-457, 2001.

[36] M. Naor, B. Pinkas, and R. Sumner, *Privacy Preserving Auctions and Mechanism Design*, proceedings of ACM Conference on Electronic Commerce, pp. 129–139, 1999.

[37] V. Nikov, S. Nikova, B. Preneel, and J. Vandewalle, *On Unconditionally Secure Distributed Oblivious Transfer*, Progress in Cryptology: Proceedings of Indocrypt 2002, LNCS, Springer-Verlag, vol. 2551, pp. 395–408, 2002.

[38] M. Rabin, *How to Exchange Secrets by Oblivious Transfer*, Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[39] 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

[40] A. Shamir, *How to Share a Secret*, Communications of ACM, vol. 22, n. 11, pp. 612–613, 1979.

[41] D.R. Stinson and R. Wei, *Bibliography on Secret Sharing Schemes.* http://www.cacr.math.uwaterloo.ca/~dstinson/ssbib.html.

[42] D.R. Stinson, *An Explication of Secret Sharing Schemes*, Designs, Codes, and Cryptography, Vol. 2, pp. 357–390, 1992.

[43] W. Tzeng, *Efficient 1-out-of-n Oblivious Transfer Schemes*, Proceedings of PKC 2002, LNCS, Springer Verlag, Vol. 2274, pp. 159–171, 2002.

[44] S. Wiesner, *Conjugate Coding*, SIGACT News 15, pp. 78-88, 1983.

A Information Theory Elements

In this appendix we briefly recall some concepts of information theory. The reader is referred to [14] for details.

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 "distance" of the random variable from a random variable uniformly distributed
- 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

$$H(\mathbf{X}|\mathbf{Y}) \geq 0. \tag{14}$$

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,

$$I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X}) \text{ and } I(\mathbf{X};\mathbf{Y}) \geq 0, \tag{15}$$

it is easy to see that

$$H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y}), \tag{16}$$

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, $\mathbf{X}_1 \ldots \mathbf{X}_n \mathbf{Y}$, the entropy of $\mathbf{X}_1 \ldots \mathbf{X}_n$ given $\mathbf{Y}$ can be written as

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n | \mathbf{Y}) = H(\mathbf{X}_1 | \mathbf{Y}) + H(\mathbf{X}_2 | \mathbf{X}_1 \mathbf{Y}) + \cdots + H(\mathbf{X}_n | \mathbf{X}_1 \ldots \mathbf{X}_{n-1} \mathbf{Y}). \tag{17}$$

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

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) = \sum_{i=1}^{n} H(\mathbf{X}_i | \mathbf{X}_1 \ldots \mathbf{X}_{i-1}) \leq \sum_{i=1}^{n} H(\mathbf{X}_i). \tag{18}$$

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

$$H(\mathbf{X}_1 \ldots \mathbf{X}_n) \geq H(\mathbf{X}_1 \ldots \mathbf{X}_k). \tag{19}$$

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

$$I(\mathbf{X}; \mathbf{Y} | \mathbf{Z}) = H(\mathbf{X} | \mathbf{Z}) - H(\mathbf{X} | \mathbf{Z}\ \mathbf{Y}) = H(\mathbf{Y} | \mathbf{Z}) - H(\mathbf{Y} | \mathbf{Z}\ \mathbf{X}) = I(\mathbf{Y}; \mathbf{X} | \mathbf{Z}). \tag{20}$$

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

$$H(\mathbf{X} | \mathbf{Z}) \geq H(\mathbf{X} | \mathbf{Z}\ \mathbf{Y}). \tag{21}$$

Finally, it is not difficult to show that, given three random variable $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$, such that $H(\mathbf{Z} | \mathbf{Y}) = 0$, then

$$H(\mathbf{X} | \mathbf{Y}) = H(\mathbf{X} | \mathbf{Z}, \mathbf{Y}). \tag{22}$$