

# On Unconditionally Secure Distributed Oblivious Transfer

Ventzislav Nikov<sup>1</sup>, Svetla Nikova<sup>2</sup> \*, Bart Preneel<sup>2</sup>, and Joos Vandewalle<sup>2</sup>

<sup>1</sup> Department of Mathematics and Computing Science  
Eindhoven University of Technology  
P.O. Box 513, 5600 MB, Eindhoven, the Netherlands  
[vnikov@mail.com](mailto:vnikov@mail.com)

<sup>2</sup> Department Electrical Engineering, ESAT/COSIC,  
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10,  
B-3001 Heverlee-Leuven, Belgium

[svetla.nikova](mailto:svetla.nikova), [bart.preneel](mailto:bart.preneel), [joos.vandewalle@esat.kuleuven.ac.be](mailto:joos.vandewalle@esat.kuleuven.ac.be)

**Abstract.** This work is about distributed protocols for oblivious transfer, proposed by Naor and Pinkas, and recently generalized by Blundo et. al. In this settings a Sender has  $n$  secrets and a Receiver is interested in one of them. The Sender distributes the information about the secrets to  $m$  servers, and a Receiver must contact a threshold of the servers in order to compute the secret. These distributed oblivious transfer protocols provide information theoretic security. We present impossibility result and lower bound for existence of one-round threshold distributed oblivious transfer protocols, generalizing the results of Blundo et. al. A threshold based construction implementing 1-out-of- $n$  distributed oblivious transfer achieving the proved lower bound for existence is proposed. A condition for existence of general access structure distributed oblivious transfer scheme is proven. We also present a general access structure protocol implementing 1-out-of- $n$  distributed oblivious transfer.

## 1 Introduction

*Oblivious Transfer (OT)* refers to several types of two-party protocols where at the beginning of the protocol one party, the *Sender*, has an input, and at the end of the protocol the other party, the *Receiver* (sometimes called the chooser), learns some information about this input in a way that does not allow the Sender to figure out what the Receiver has learned. Introduced by M. Rabin in [22], and subsequently defined in different forms in [15, 5], the oblivious transfer has found many applications in cryptographic studies and protocol design. 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 *OT* with other cryptographic primitives (e.g. see [1, 3, 6, 12, 13, 19]).

---

\* The author was partially supported by NATO research fellowship and Concerted Research Action GOA-MEFISTO-666 of the Flemish Government.

The Private Information Retrieval (PIR) and Symmetric Private Information Retrieval (SPIR) Schemes, introduced in [7, 16] represent another very close area. In PIR and SPIR schemes the emphasis is placed on the communication complexity of the interaction between user and servers. Other interesting PIR papers for the distribute OT scenario are [2, 11, 17].

Rivest’s model given in [23] utilizes a trusted initializer, who participates only in an initial setup phase. The setting of the scheme is close to the one described in [20] and considered in this paper. In the very recent paper [24] the author deals with distributed oblivious transfer implementations, close to the settings in [20], but not unconditionally secure.

In this paper we are concerned with *unconditionally secure distributed oblivious transfer protocols*, introduced by Naor and Pinkas in [20] and recently generalized by Blundo et. al. in [4]. Distributed oblivious transfer (*DOT*) protocols distribute the task of the Sender between several servers. Security is ensured as long as a limited number of these servers collude. We present an analysis of the threshold and general access structure model for *DOT*. We prove an impossibility result and a lower bound for existence of one-round threshold distributed oblivious transfer protocols, generalizing the result of Blundo et. al. A threshold based construction implementing 1-out-of- $n$  *DOT* achieving the proved lower bound for existence is proposed as well.

Since in many natural scenarios the assumption that trust is “uniformly distributed” over the players does not model the reality well, and moreover, in more realistic model no threshold solution will work, we want to protect against general adversary structures. We give a condition for existence of general access structure *DOT* scheme. Finally, a General Access Structure protocol implementing 1-out-of- $n$  *DOT* is presented.

The paper is organized as follows. In the next section we give the basic definitions for the distributed oblivious transfer and a formal model. In Section 3 using some Information Theory tools we prove an impossibility result for certain parameters and consequently we derive a lower bound for the existence of *DOT* with the same parameters. Protocol Implementing  $(r, m) - DOT - \binom{n}{1}$  is presented in Section 4. In sections 5, 6 and 7 a General Access Structure model for *DOT* –  $\binom{n}{1}$  is analyzed and the corresponding protocol is constructed.

## 2 The Distributed Model

### 2.1 Definitions

A distributed  $r$ -out-of- $m$  *OT* –  $\binom{n}{1}$  protocol involves three types of parties:

- A **Sender**  $\mathcal{S}$  which has  $n$  inputs (secrets)  $s_0, s_1, \dots, s_{n-1}$ . It is convenient to assume that these inputs are elements in a finite field  $\mathbb{F}$ .
- A **Receiver**  $\mathcal{R}$  that has an input  $\sigma \in \{0, 1, \dots, n - 1\}$ .
- Additional  $m$  **servers**,  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m$ .

We assume that the Sender holds  $n$  secrets and the Receiver is interested in one of them. 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.

The protocol is composed of the following functional steps:

- **Initialization Phase.** Let  $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_m$  be  $m$  servers. The Sender  $\mathcal{S}$  generates  $m$  programs  $P_1, P_2, \dots, P_m$  and, for  $i = 1, \dots, m$  sends in a *secure way*, program  $P_i$  to the server  $\mathcal{S}_i$ . Each program  $P_i$  depends on the secrets  $s_0, s_1, \dots, s_{n-1}$  and on some random data.
- **Oblivious Transfer Phase.** The Receiver  $\mathcal{R}$  holds a program  $R$  which enables her to interact with a subset  $\{\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_r}\}$  of  $r$  servers at her choice. She sends to the server  $\mathcal{S}_i$  a query  $q_i$  which is a function of  $\sigma$  and  $i$ , and of some random data. The server answers the query with  $a_i$ . Using the answers the Receiver  $\mathcal{R}$  is collected, she is able to recover the secret in which is interested, receiving no information about the other secrets. At the same time, any subset of  $t - 1$  servers, say  $\{\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_{t-1}}\} \subseteq \{\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_r}\}$ , does not gain any information about the secret she has recovered.

More precisely, a distributed  $(r, m) - DOT - \binom{n}{1}$  must guarantee the following properties:

- **Reconstruction.** If the Receiver gets information from  $r$  out of the  $m$  servers, she can compute the secret  $s_\sigma$ .
- **Sender's Privacy.** Given any  $r$  values, the Receiver must gain information about a single secret, and no information about the others.
- **Receiver's Privacy.** No coalition of less than  $t$  servers gains information about which secret the Receiver has recovered.
- **Receiver-servers collusion.** A coalition of the Receiver with  $l$  corrupt servers cannot learn about the  $n$  secrets more than can be learned by the Receiver herself.

## 2.2 A Formal Model

In this section we will follow the notations and the formal model given by Blundo et. al. in [4]. Assume that  $\mathcal{S}$  holds a program  $S$  to generate  $m$  programs  $P_1, \dots, P_m$  enabling  $\mathcal{S}_1, \dots, \mathcal{S}_m$  and  $\mathcal{R}$  to perform  $(r, m) - DOT - \binom{n}{1}$  oblivious transfer of his  $n$  secrets.  $\mathcal{R}$  holds an associated program  $R$  for interacting with the servers. The  $m + 1$  programs  $P_1, \dots, P_m$  and  $R$ , specify the computations to be performed to achieve  $(r, m) - DOT - \binom{n}{1}$ . In order to model dishonest behavior, where a coalition of at most  $t - 1$  servers tries to figure out which secret  $\mathcal{R}$  has recovered from the transfer, we assume that cheating servers  $\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_{t-1}}$  hold a modified version of the programs, denoted by  $\bar{P}_{i_1}, \dots, \bar{P}_{i_{t-1}}$ . These programs could have been generated either by a dishonest  $\mathcal{S}$ , who holds a cheating program  $\bar{S}$ , or 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  $\bar{R}$ . These programs can be described by random variables, which will be denoted by the same letters in bold.

An execution of the protocol can be described by using the following additional random variables: for  $j = 1, \dots, m$  let  $C_j$  be the transcript of the communication between  $\mathcal{R}$  and  $\mathcal{S}_j$ . Moreover, let  $W$  be the set of all length  $n$  sequences of secrets, and, for any  $w \in W$ , let  $w_i$  be the  $i$ -th secret of the sequence. Denoting by  $\mathbf{W}$  the random variable that represents the choice of an element in  $W$  and by  $\mathbf{T}$  the random variable representing the choice of an index  $i$  in  $T = \{0, 1, \dots, n-1\}$ , one can define (as in [4]) the conditions that  $(r, m) - DOT - \binom{n}{1}$  oblivious transfer protocol must satisfy as follows:

**Definition 1.** [4] *The sequence of programs  $[S, P_1, \dots, P_m, R]$  is correct for  $(r, m) - DOT - \binom{n}{1}$  if for any  $i \in T$  and  $j = 1, \dots, m$*

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

and, for any  $w \in W$  and for any  $\{i_1, \dots, i_r\} \subseteq \{1, \dots, m\}$

$$H(\mathbf{W}_T | \mathbf{C}_{i_1} \dots \mathbf{C}_{i_r}) = 0. \quad (2)$$

The definition means that the transcript of the communication is completely determined by the program of the server  $\mathcal{S}_j$  and the program of the Receiver and her choices. Moreover, after interacting with  $r$  servers, an honest Receiver always recovers the secret in which 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  $(r, m) - DOT - \binom{n}{1}$  can be defined as follows:

**Definition 2.** [4] *The sequence of programs  $[S, P_1, \dots, P_m, R]$  is private for  $(r, m) - DOT - \binom{n}{1}$  if*

– for any set of indices  $\{i_1, \dots, i_{t-1}\} \subseteq \{1, \dots, m\}$ ,

$$H(\mathbf{T} | \overline{\mathbf{P}}_{i_1} \dots \overline{\mathbf{P}}_{i_{t-1}} \mathbf{C}_{i_1} \dots \mathbf{C}_{i_{t-1}}) = H(\mathbf{T}). \quad (3)$$

– for any program  $\overline{\mathbf{R}}$ , for any  $i \in T$  and for any set of indices  $\{i_1, \dots, i_r\} \subseteq \{1, \dots, m\}$ ,

$$H(\mathbf{W} \setminus \mathbf{W}_T | \mathbf{T} \overline{\mathbf{R}} \mathbf{C}_{i_1} \dots \mathbf{C}_{i_r} \mathbf{W}_T) = H(\mathbf{W} \setminus \mathbf{W}_T). \quad (4)$$

– for any set of indices  $\{i_1, \dots, i_l\} \subseteq \{1, \dots, m\}$  for any  $i \in T$  and for any  $\overline{\mathbf{R}}$ ,

$$H(\mathbf{W} | \mathbf{T} \overline{\mathbf{R}} \mathbf{C}_{i_1} \dots \mathbf{C}_{i_l} \overline{\mathbf{P}}_{i_1} \dots \overline{\mathbf{P}}_{i_l}) = H(\mathbf{W}). \quad (5)$$

– for any pair of sets of indices  $\{i_1, \dots, i_l\} \subseteq \{1, \dots, m\}$  and  $\{j_1, \dots, j_r\} \subseteq \{1, \dots, m\}$ , for any  $i \in T$  and for any  $\overline{\mathbf{R}}$ ,

$$H(\mathbf{W} \setminus \mathbf{W}_T | \mathbf{T} \overline{\mathbf{R}} \overline{\mathbf{P}}_{i_1} \dots \overline{\mathbf{P}}_{i_l} \mathbf{C}_{j_1} \dots \mathbf{C}_{j_r} \mathbf{W}_T) = H(\mathbf{W} \setminus \mathbf{W}_T). \quad (6)$$

These first two conditions ensure that a dishonest coalition of  $t - 1$  servers does not gain information about  $\mathcal{R}$ 's index: a dishonest  $\mathcal{R}$  infers at most one secret among the ones held by  $\mathcal{S}_1, \dots, \mathcal{S}_m$ . Condition (5) takes into account the possibility of an attack against  $\mathcal{S}$  performed either by at most  $l$  servers alone or with the cooperation of  $\mathcal{R}$ . The condition states that such kind of coalitions do not gain any information about the secrets held by  $\mathcal{S}$ . Finally, conditions (6) states that a coalition of  $l$  servers and the Receiver cannot compute any information about the others, once the Receiver has obtained a secret.

### 3 Impossibility Result and Lower Bound for Existence

Using some Information Theory tools and the ideas in [4] we can show that with one round *DOT* protocol an impossibility result holds for the parameters  $r$ ,  $t$ , and  $l$ . And consequently a lower bound for the existence of *DOT* with parameters  $r$ ,  $t$ , and  $l$  will be proved.

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

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

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

**Theorem 1.** *In any  $(r, m) - \text{DOT} - \binom{n}{1}$  scheme with parameters  $t$ , and  $l$  such that  $r < t + l$ , once the Receiver has legally recovered a secret, a coalition of  $l$  corrupt servers and the Receiver can recover all the others.*

*Proof.* Let  $r = l + t - 1$  i.e.  $l = r - t + 1$ . Let  $q_1, \dots, q_r$  be the queries sent by the Receiver when  $T = i$ , and let  $a_1, \dots, a_r$  be the answers that  $\mathcal{S}_1, \dots, \mathcal{S}_r$  send back to  $\mathcal{R}$ . The Receiver's security property (3) with respect to  $t - 1$  servers, say  $\mathcal{S}_{l+1}, \dots, \mathcal{S}_r$ , implies that there exist queries  $q_1^s, \dots, q_l^s$  and answers  $a_1^s, \dots, a_l^s$  for any  $s \neq i$ , such that if

$$H(\mathbf{W}_i | \mathbf{Q}_1 = q_1 \dots \mathbf{Q}_r = q_r, \mathbf{A}_1 = a_1 \dots \mathbf{A}_r = a_r) = 0$$

then

$$\begin{aligned} H(\mathbf{W}_s | \mathbf{Q}_1 = q_1^s \dots \mathbf{Q}_l = q_l^s \mathbf{Q}_{l+1} = q_{l+1} \dots \mathbf{Q}_r = q_r, \\ \mathbf{A}_1 = a_1^s \dots \mathbf{A}_l = a_l^s \mathbf{A}_{l+1} = a_{l+1} \dots \mathbf{A}_r = a_r) = 0. \end{aligned}$$

Since the answers given by  $\mathcal{S}_1, \dots, \mathcal{S}_l$  depend only on their own programs  $P_1, \dots, P_l$  and on the received queries (i.e.  $H(\mathbf{A}_j | \mathbf{Q}_j \mathbf{P}_j) = 0$  for  $j = 1, \dots, l$ ) it holds that

$$H(\mathbf{W} | \mathbf{P}_1 \dots \mathbf{P}_l \mathbf{A}_{l+1} \dots \mathbf{A}_r, \mathbf{Q}_{l+1} \dots \mathbf{Q}_r \mathbf{R}) = 0.$$

Indeed

$$\begin{aligned} & H(\mathbf{W} | \mathbf{P}_1 \dots \mathbf{P}_l \mathbf{A}_{l+1} \dots \mathbf{A}_r, \mathbf{Q}_{l+1} \dots \mathbf{Q}_r \mathbf{R}) \\ & \leq \sum_{t \in T} H(\mathbf{W}_t | \mathbf{P}_1 \dots \mathbf{P}_l \mathbf{A}_{l+1} \dots \mathbf{A}_r, \mathbf{Q}_{l+1} \dots \mathbf{Q}_r \mathbf{R}, \mathbf{T} = t) \end{aligned}$$

and

$$\begin{aligned}
& H(\mathbf{W}_t | \mathbf{P}_1 \dots \mathbf{P}_l \mathbf{A}_{l+1} \dots \mathbf{A}_r, \mathbf{Q}_{l+1} \dots \mathbf{Q}_r, \mathbf{R}, \mathbf{T} = t) \\
& \leq H(\mathbf{W}_t | \mathbf{P}_1 \dots \mathbf{P}_l \mathbf{A}_{l+1} \dots \mathbf{A}_r, \mathbf{Q}_1 \dots \mathbf{Q}_r) \\
& \leq H(\mathbf{W}_t | \mathbf{A}_1 \dots \mathbf{A}_r, \mathbf{Q}_1 \dots \mathbf{Q}_r) = 0.
\end{aligned}$$

Therefore the Receiver and a coalition of  $l$  servers can recover all the secrets and the result holds.  $\square$

The last theorem is a natural extension of Theorem 3.5 in [4], where the case  $r = k, t = k, l = 1$  is considered.

A consequence of this impossibility result for one-round protocols is the lower bound for existence of *DOT* with parameters  $r, t$ , and  $l$ .

**Corollary 1.** *A necessary and sufficient condition for existence of  $(r, m) - DOT - \binom{n}{1}$  scheme with parameters  $t$  and  $l$  is*

$$r \geq t + l.$$

*Proof.* The necessity follows directly from Theorem 1. In the next section the protocol implementing  $(r, m) - DOT - \binom{n}{1}$  scheme with parameters  $t, l$  and satisfying  $r = t + l$  will be presented, which prove the sufficient condition.  $\square$

Note that two-round protocols, as for example the one proposed in [4], satisfy the same bound, because two times contacting  $k$  servers can be viewed as once contacting  $2k$  servers. Hence  $r = 2k, t = l = k$  are suitable parameters for existence of *DOT*.

## 4 Protocol Implementing $(r, m) - DOT - \binom{n}{1}$

Two protocols for  $(r, m) - DOT - \binom{2}{1}$  have been proposed by Naor and Pinkas in [20]. Recently Blundo et. al. in [4] generalized the idea of Naor and Pinkas and proposed several protocols for  $(r, m) - DOT - \binom{n}{1}$ . The protocols proposed by Naor and Pinkas and two of the protocols in [4] are based on polynomial interpolation. Combinatorial constructions are presented in [4] as well.

In this section we propose a protocol, based on polynomial interpolation. It is a generalization of the protocols of Naor and Pinkas and Blundo et. al. The protocol is described as follows:

**Initialization Phase.** Let  $s_0, s_1, \dots, s_{n-1} \in \mathbb{F}$  ( $\mathbb{F}$  - finite field) be the Sender's  $\mathcal{S}$  secrets.

1.  $\mathcal{S}$  generates  $n - 1$  random polynomials  $B_1(x), \dots, B_{n-1}(x)$  of degree  $l$  and one random polynomial  $B_0(x)$  of degree  $r - 1 \geq l + t - 1$  with values in  $\mathbb{F}$  such that  $B_0(0) = s_0$  and, for  $i = 1, \dots, n - 1$ ,  $s_i = B_0(0) + B_i(0)$ .
2. Then,  $\mathcal{S}$  constructs an  $n$ -variate polynomial  $Q(x, y_1, \dots, y_{n-1})$  with values in  $\mathbb{F}$  such that  $Q(0, 0, \dots, 0) = s_0$ ,  $Q(0, 1, 0, \dots, 0) = s_1, \dots, Q(0, 0, \dots, 1) = s_{n-1}$ . More precisely,

$$Q(x, y_1, \dots, y_{n-1}) = B_0(x) + \sum_{j=1}^{n-1} B_j(x)y_j$$

3. Finally, for  $i = 1, \dots, m$ , he sends the  $n - 1$  variate polynomial  $Q(i, y_1, \dots, y_{n-1})$  to the server  $\mathcal{S}_i$ .

**Oblivious Transfer Phase.** Let  $\sigma \in \{0, 1, \dots, n - 1\}$  be the Receiver's  $\mathcal{R}$  index.

1.  $\mathcal{R}$  generates  $n - 1$  random polynomials  $D_1(x), \dots, D_{n-1}(x)$  of degree  $t - 1$  such that  $(D_1(0), \dots, D_{n-1}(0))$  is an  $(n - 1)$ -tuple of zeroes with at most a 1 in position  $\sigma$ , the position corresponding to the secret in which she is interested. Define a univariate polynomial  $V$  to be  $V(x) = Q(x, D_1(x), \dots, D_{n-1}(x))$ .
2. Then, she asks  $r$  servers  $\mathcal{S}_{i_j}$  for  $j = 1, \dots, r$ , sending a query of the form  $(D_1(i_j), \dots, D_{n-1}(i_j))$ .
3. The server  $\mathcal{S}_{i_j}$  calculates the value  $Q(i_j, D_1(i_j), \dots, D_{n-1}(i_j)) = V(i_j)$  and sends it back to  $\mathcal{R}$ .
4. After receiving  $r$  values of  $V$ , say  $V(i_1), \dots, V(i_r)$ ,  $\mathcal{R}$  interpolates  $V(x)$  and computes  $V(0)$ .

#### 4.1 Correctness and Security

The correctness of the proposed protocol: The degree of the polynomial  $V(x)$  is  $r - 1$ , hence receiving  $r$  values in step 3. the Receiver is able to recover correctly  $V(x)$  and calculate  $V(0)$ . On the other hand assuming that  $(D_1(0), \dots, D_{n-1}(0)) = (0, \dots, 0, 1, 0, \dots, 0)$  (i.e. at most a 1 in position  $\sigma$ ), then

$$V(0) = Q(0, D_1(0), \dots, D_{n-1}(0)) = Q(0, 0, \dots, 0, 1, 0, \dots, 0) = s_\sigma.$$

Now we will see that the proposed protocol for  $(r, m) - DOT - \binom{n}{1}$  satisfy the four properties described in the definition.

About the **Reconstruction** as we have already checked our protocol is correct. The **Receiver's Privacy** is guaranteed against coalitions of at most  $t - 1$  servers, because  $\mathcal{R}$  herself chooses polynomials  $D_1(x), \dots, D_{n-1}(x)$  to have degree  $t - 1$ . Again using the proof for correctness of the proposed protocol it follows

that **Sender's Privacy** is guaranteed. And finally the **Receiver-servers collusion**, assuming that the Receiver has already calculated one secret and that a coalition of at most  $l$  corrupt servers helps her to discover others. Because the Sender  $\mathcal{S}$  chooses the polynomials  $B_1(x), \dots, B_{n-1}(x)$  of degree  $l$  and a polynomial  $B_0(x)$  of degree  $r - 1 \geq l + t - 1$ , the information these  $l$  corrupt servers possess (i.e.  $B_0(i_j), B_1(i_j), \dots, B_{n-1}(i_j)$  for  $j = 1, \dots, l$ ) is insufficient to recover any of the polynomials  $B_0(x), B_1(x), \dots, B_{n-1}(x)$ , hence it is insufficient to find any of the values  $B_0(0), B_1(0), \dots, B_{n-1}(0)$ .

*Remark:* The proposed protocol satisfy  $r = l + t$ , which prove the “sufficient” part in the proof of the Corollary 1.

## 4.2 Efficiency

Comparing our scheme with the polynomial scheme of Blundo et. al. they are equal in respect of the following parameters: the memory storage of servers, the complexity of each interaction, the randomness to set up the scheme and the randomness of the whole communication. The proposed here scheme achieves the bounds of Theorems 3.1, 3.2, 3.3, 3.4 in [4]. But the memory storage for the Sender and the Receiver is higher in our scheme, because it provides better security.

One of the questions that Naor and Pinkas arose is how the scheme will ensure that a Receiver does not obtain more than  $r$  shares. It is clear that in our scheme the Sender can choose  $m = r$ , and solve this problem providing the desired security.

## 5 General Access Structure model for $DOT - \binom{n}{1}$

Threshold-based schemes make sense only in environment where one assumes that any player subset of a certain cardinality is equally likely (or unlikely) to cheat (or to be corrupted). In many natural scenarios this assumption does not model the reality well, thus we need to protect against general adversary structures. The well known drawback of using general access structure approach than the threshold one is that the memory storage and the complexity of each interaction will be not optimal. In this section we will apply a general access structure method for building a  $DOT - \binom{n}{1}$ .

### 5.1 Definitions

A Distributed General Access Structure  $OT - \binom{n}{1}$  protocol involves the same three types of parties as in the threshold case: Sender, Receiver and servers.

The protocol now is composed in nearly the same way with a few changes in the **Oblivious Transfer Phase**: The Receiver  $\mathcal{R}$  holds a program  $R$  which enables her to interact with a subset of qualified servers  $\{\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_r}\} \in \Gamma$  at her choice. At the same time, any subset  $\{\mathcal{S}_{i_1}, \dots, \mathcal{S}_{i_{t-1}}\} \in \Delta_1$  of forbidden servers, does not gain any information about the secret she has recovered.

More precisely, a Distributed General Access Structure  $DOT - \binom{n}{1}$  must guarantee the following properties:

- **Reconstruction.** If the Receiver gets information from a set of qualified servers  $G \in \Gamma$ , she can compute the secret  $s_\sigma$ .
- **Sender’s Privacy.** Given any set of qualified servers  $G \in \Gamma$  values, the Receiver must gain information about a single secret, and no information about the others.
- **Receiver’s Privacy.** No coalition of set of forbidden servers  $G_1 \in \Delta_1$  gains information about which secret the Receiver has recovered.
- **Receiver-servers collusion.** A coalition of the Receiver with a set of corrupt servers  $G_2 \in \Delta_2$  cannot learn about the  $n$  secrets more than can be learned by the Receiver herself.

The set of  $m$  servers is divided in three sets of subsets  $\Gamma, \Delta_1, \Delta_2$  of qualified, forbidden and corrupt servers, resp. The set  $\Gamma$  is monotone increasing and the sets  $\Delta_1, \Delta_2$  are monotone decreasing.

## 6 Condition for Existence

First we will give the following definition:

**Definition 3.** [21] We define the operation  $*$  for any monotone decreasing sets  $\Delta_1, \Delta_2$  as follows:  $\Delta_1 * \Delta_2 = \{A = A_1 \cup A_2; A_1 \in \Delta_1, A_2 \in \Delta_2\}$ .

It is easy to check that  $\Delta_1 * \Delta_2$  is also monotone decreasing.

The same operation for monotone structures is defined by Fehr and Maurer in [14], which they call element-wise union, in order to give necessary and sufficient conditions for robust VSS and Distributed Commitments.

Using some Information Theory tools we can show, in the same way as in the threshold case (see Theorem 1), that there is a condition for existence of one-round General Access Structure  $DOT$  protocol.

**Theorem 2.** In any General Access Structure  $DOT - \binom{n}{1}$  scheme with set of qualified, forbidden and corrupt servers  $\Gamma, \Delta_1, \Delta_2$ , and such that  $\Gamma \cap (\Delta_1 * \Delta_2) \neq \emptyset$ , once the Receiver has legally recovered a secret, a coalition of corrupt servers from  $\Delta_2$  and the Receiver can recover all the others.

A consequence of this existence condition for one-round protocols is the following Corollary.

**Corollary 2.** A necessary condition for existence of General Access Structure  $DOT - \binom{n}{1}$  scheme with set of qualified, forbidden and corrupt servers  $\Gamma, \Delta_1, \Delta_2$ , is the tuple  $(\Gamma, \Delta_1 * \Delta_2)$  to be access structure.

Denote by  $U = \{\mathcal{S}_1, \dots, \mathcal{S}_m\}$  the set of servers and by  $2^U$  the set of all subsets of  $U$ . Denote  $\Gamma_1 = \Delta_1^c$  to be the complement of  $\Delta_1$  to the  $2^U$  and  $\Gamma_2 = \Delta_2^c$  to be the complement of  $\Delta_2$  to the  $2^U$ . Correspondingly we have  $\Delta_1 = \Gamma_1^c$  and  $\Delta_2 = \Gamma_2^c$ . It is well known that  $\Gamma_1, \Gamma_2$  are monotone increasing. Now we can consider three separate access structures  $\Gamma, \Gamma_1, \Gamma_2$ .

**Definition 4.** [21] We define the operation  $*$  for any monotone **increasing** sets  $\Gamma_1, \Gamma_2$  as follows:  $\Gamma_1 * \Gamma_2 = (\Delta_1 * \Delta_2)^c$ .

## 7 General Access Structure protocol for *DOT* – $\binom{n}{1}$

In this section we propose a protocol for General Access Structure *DOT* –  $\binom{n}{1}$ . Most proposed SSS are *linear*, but the concept of an LSSS was first considered in its full generality by Karchmer and Wigderson in [18], who introduced the equivalent notion of *Monotone Span Program* (MSP), which we describe later. Each linear SSS can be viewed as derived from a monotone span program  $\mathcal{M}$  computing its access structure. On the other hand, each monotone span program gives rise to an LSSS. Hence, one can identify an LSSS with its underlying monotone span program. Such an MSP always exists because MSP’s can compute any monotone function. Note that the size of  $\mathcal{M}$  is also the size of the corresponding LSSS. Now we will consider any access structure, as long as it admits a linear secret sharing scheme.

We will use the definitions and results by Cramer et. al. in [9] about General Secure Multi-Party Computation.

**Definition 5.** [9, 8] A *Monotone Span Program*  $\mathcal{M}$  is a quadruple  $(\mathbb{F}, M, \varepsilon, \psi)$ , where  $\mathbb{F}$  is a finite field,  $M$  is a matrix (with  $e$  rows and  $d \leq e$  columns) over  $\mathbb{F}$ ,  $\psi : \{1, \dots, e\} \rightarrow \{1, \dots, m\}$  is a surjective function and  $\varepsilon$  is a fixed vector, called *target vector*, e.g. column vector  $(1, 0, \dots, 0) \in \mathbb{F}^d$ . The size of  $\mathcal{M}$  is the number of rows ( $e$ ).

Thus,  $\psi$  labels each row with a number from  $[1, \dots, e]$  corresponding to a fixed player, hence we can think of each player as being the “owner” of one or more rows. For every player we consider a function  $\varphi$  which gives the set of rows owned by the player, i.e.  $\varphi$  is (in some sense) inverse of  $\psi$ .

It is known (e.g. see [10, Remark 2]) that the number of columns  $d$  can be increased, without changing the access structure that is computed by a MSP. The space generated by the 2nd up to the  $d$ -th column of  $M$  does not contain even a non-zero multiple of the first column. Without changing the access structure that is computed, we can always replace the 2nd up to the  $d$ -th column of  $M$  by any set of vectors that generates the same space.

MSP is said to compute an access structure  $\Gamma$  when  $\varepsilon \in \text{Im}(M_G^T)$  if and only if  $G$  is a member of  $\Gamma$ . So, the players can reconstruct the secret precisely if the rows they own contain in their linear span the target vector of  $\mathcal{M}$ , and otherwise they get no information about the secret, i.e. there exists a so called *recombination vector*  $\mathbf{r}$  such that  $\langle \mathbf{r}, M_G(s, \rho) \rangle = s$  and  $M_G^T \mathbf{r} = \varepsilon$  for any secret  $s$  and any  $\rho$ .

Let  $f_1$  and  $f_2$  be monotone boolean functions, computed by MSP’s  $\mathcal{M}_1 = (\mathbb{F}, M_1, \varepsilon, \psi)$  and  $\mathcal{M}_2 = (\mathbb{F}, M_2, \varepsilon, \psi)$ . Given two  $d$ -vectors  $\mathbf{x}$  and  $\mathbf{y}$ , Cramer et. al. in [9, 8] denote  $\mathbf{x} \diamond \mathbf{y}$  to be the vector containing all entries of form  $x_i y_j$ , where  $\psi(i) = \psi(j)$ . Thus, if  $d_i = |\varphi(i)|$  is the number of rows owned by a player  $i$ , then  $\mathbf{x} \diamond \mathbf{y}$  has  $\bar{d} = \sum_i d_i^2$  entries. So, if  $\mathbf{x}, \mathbf{y}$  contain shares resulting from sharing

two secrets using  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , then the vector  $\mathbf{x} \diamond \mathbf{y}$  can be computed using only local computation by the players, i.e. each component of the vector can be computed by one player.

**Definition 6.** [9, 8] *A multiplicative MSPs are the MSPs  $\mathcal{M}_1$  and  $\mathcal{M}_2$  for which there exists an  $\bar{d}$ -vector  $\mathbf{r}$  called a **recombination vector**, such that for any two secrets  $s'$  and  $s''$  and any  $\rho'$  and  $\rho''$ , it holds that*

$$s' s'' = \langle \mathbf{r}, M_1(s', \rho') \diamond M_2(s'', \rho'') \rangle$$

It means that one can construct a multiplicative MSP computing  $f_1 \vee f_2$ . We will call it *multiplicative result MSP*.

**Definition 7.** [9, 8] *If  $A$  is a player subset,  $\mathcal{M}_A$  is the MSP obtained by  $\mathcal{M}$  by keeping only the rows owned by players in  $A$ . We say that  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are **strongly multiplicative** if for any player subset  $A$  that is qualified by both  $\mathcal{M}_1$  and  $\mathcal{M}_2$ ,  $(\mathcal{M}_1)_A$  and  $(\mathcal{M}_2)_A$  are multiplicative.*

It means that one can construct a strongly multiplicative MSP computing  $f_1 \vee f_2$ , this MSP we will call *strongly multiplicative result MSP*.

We are now ready to describe the protocol for General Access Structure  $DOT - \binom{n}{1}$  scheme with set of qualified, forbidden and corrupt servers  $\Gamma, \Delta_1, \Delta_2$ , resp., and the corresponding three access structures  $\Gamma, \Gamma_1, \Gamma_2$ .

Let  $\Gamma_1, \Gamma_2$  be the access structures with the MSPs  $\mathcal{M}_1$  and  $\mathcal{M}_2$ , which possess strongly multiplicative property. Denote by  $\bar{\Gamma}$  the access structure corresponding to the strongly multiplicative result MSP  $\bar{\mathcal{M}}$  (see Definition 7).

**Definition 8.** *We say that MSPs  $\mathcal{M}, \mathcal{M}_1$  and  $\mathcal{M}_2$  are **DOT MSPs** if there exists a  $\bar{d}$ -vector  $\mathbf{r}$  called a **recombination vector**, such that for any three secrets  $s, s'$  and  $s''$  and any  $\rho, \rho'$  and  $\rho''$ , it holds that*

$$s + s' s'' = \langle \mathbf{r}, M(s, \rho) + M_1(s', \rho') \diamond M_2(s'', \rho'') \rangle$$

**Lemma 1.** *A necessary condition for existence of DOT MSPs  $\mathcal{M}, \mathcal{M}_1$  and  $\mathcal{M}_2$  is that  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are strongly multiplicative, and their strongly multiplicative result MSP  $\bar{\mathcal{M}} = \mathcal{M}$  (i.e.  $\Gamma = \bar{\Gamma}$ ).*

Thus, a necessary condition for existence of General Access Structure  $DOT - \binom{n}{1}$  scheme, which turns out to be also a sufficient condition, is the following.

**Theorem 3.** *A necessary and sufficient condition for existence of General Access Structure  $DOT - \binom{n}{1}$  scheme with set of qualified, forbidden and corrupt servers  $\Gamma, \Delta_1, \Delta_2$ , and the corresponding to them three access structures  $\Gamma, \Gamma_1, \Gamma_2$  is that their MSPs  $\mathcal{M}, \mathcal{M}_1$  and  $\mathcal{M}_2$  are DOT MSPs.*

Now we are ready to present the following protocol for General Access Structure  $DOT - \binom{n}{1}$  scheme.

**Initialization Phase.** Let  $s_0, s_1, \dots, s_{n-1} \in \mathbb{F}$  be the Sender's  $\mathcal{S}$  secrets. There are three access structures  $\Gamma, \Gamma_1, \Gamma_2$  and corresponding to them three MSPs  $\mathcal{M}_1 = (\mathbb{F}, M_1, \varepsilon, \psi)$ ,  $\mathcal{M}_2 = (\mathbb{F}, M_2, \varepsilon, \psi)$  and  $\mathcal{M} = (\mathbb{F}, M, \varepsilon, \tilde{\psi})$  as well as the “reverse” functions  $\tilde{\varphi}$  and  $\varphi$ . In our construction we require  $\Gamma = \bar{\Gamma}$ , i.e.  $\tilde{\varphi} = \overline{\varphi}$  and  $\tilde{\psi} = \overline{\psi}$ . As we noted before the number of columns in the MSP can be increased without changing the access structure that is computed by a MSP. Therefore we can assume that the number of columns in the MSPs  $\mathcal{M}, \mathcal{M}_1$  and  $\mathcal{M}_2$  is equal to  $d$ .

1.  $\mathcal{S}$  generates  $n$  random vectors  $B_0, B_1, \dots, B_{n-1} \in \mathbb{F}^d$ , such that  $\langle B_0, \varepsilon \rangle = s_0$  and, for  $i = 1, \dots, n-1$ ;  $s_i = \langle B_0 + B_i, \varepsilon \rangle$ .
2. Then, for  $i = 1, \dots, m$ , he sends the  $n$  packets of shares  $(v_j)_{\varphi(i)}$ , for  $j = 1, \dots, n-1$  and  $(v_0)_{\tilde{\varphi}(i)}$  to the server  $\mathcal{S}_i$ . Where  $(v_0)_{\tilde{\varphi}(i)} = M_{\tilde{\varphi}(i)} B_0$  and for  $j = 1, \dots, n-1$ ,  $(v_j)_{\varphi(i)} = (M_2)_{\varphi(i)} B_j$ .

**Oblivious Transfer Phase.** Let  $\sigma \in \{0, 1, \dots, n-1\}$  be the Receiver's  $\mathcal{R}$  index.

1.  $\mathcal{R}$  generates  $n-1$  random vectors  $D_1, \dots, D_{n-1} \in \mathbb{F}^d$  such that  $(\langle D_1, \varepsilon \rangle, \dots, \langle D_{n-1}, \varepsilon \rangle)$  is an  $(n-1)$ -tuple of zeroes with at most a 1 in position  $\sigma$ , the position corresponding to the secret in which she is interested.
2. Then she asks a set of qualified servers  $\mathcal{S}_i$ , sending a query of  $n-1$  packets of temporary shares  $(v_j^R)_{\varphi(i)}$ , for  $j = 1, \dots, n-1$ . Where  $(v_j^R)_{\varphi(i)} = (M_1)_{\varphi(i)} D_j$ .
3. The server  $\mathcal{S}_i$  calculates the values

$$(v^S)_{\tilde{\varphi}(i)} = (v_0)_{\tilde{\varphi}(i)} + \sum_{j=1}^{n-1} (v_j^R)_{\varphi(i)} \diamond (v_j)_{\varphi(i)}$$

and sends it back to  $\mathcal{R}$ .

4. After receiving values  $(v^S)_{\tilde{\varphi}(i)}$  for a set of qualified servers (i.e.  $i \in G$  and  $G \in \Gamma$ ) the Receiver is able to recover the secret  $s_\sigma$ . First she computes  $\mathbf{r}$ , such that  $M_{\tilde{\varphi}(G)}^T \mathbf{r} = \varepsilon$  and then she computes  $s_\sigma = \langle (v^S)_{\tilde{\varphi}(G)}, \mathbf{r} \rangle$ .

## 7.1 Correctness and Security

The correctness of the proposed protocol: We have  $\langle B_0, \varepsilon \rangle = s_0$  and,  $s_j - s_0 = \langle B_j, \varepsilon \rangle$  for  $j = 1, \dots, n-1$ . Denote by  $(d_1, \dots, d_{n-1}) = (\langle D_1, \varepsilon \rangle, \dots, \langle D_{n-1}, \varepsilon \rangle)$ . So,  $(v_j^R)_{\varphi(i)} \diamond (v_j)_{\varphi(i)}$  is the share of  $\mathcal{S}_i$  which corresponds to the share of strongly multiplicative result MSP computing  $f_1 \vee f_2$ , i.e. the share for the secret  $d_j(s_j - s_0)$  shared with access structure  $\bar{\Gamma} = \Gamma$ . Hence the share  $(v^S)_{\tilde{\varphi}(i)}$  corresponds to the share of the same strongly multiplicative MSP with shared secret  $s_0 + \sum_{j=1}^{n-1} d_j(s_j - s_0)$ . Since  $(d_1, \dots, d_{n-1}) = (0, \dots, 0, 1, 0, \dots, 0)$  is  $(n-1)$ -tuple of zeroes with at most a 1 in position  $\sigma$ , the position corresponding to the secret

in which  $\mathcal{R}$  is interested. We have  $s_\sigma = s_0 + \sum_{j=1}^{n-1} d_j(s_j - s_0)$ . Using the well known calculations for MSP (i.e.  $\langle (v^S)_{\bar{\varphi}(G)}, \mathbf{r} \rangle$ ) the Receiver recovers the secret, which is exactly  $s_\sigma$ .

Now we will see that the proposed General Access Structure protocol for  $DOT - \binom{n}{1}$  satisfy the four properties described in the extended definition.

About the **Reconstruction** as we have already checked our protocol is correct. The **Receiver's Privacy** is guaranteed against coalitions  $\Delta_1$  of forbidden servers, because  $\mathcal{R}$  herself chooses vectors  $D_1, \dots, D_{n-1}$  with values  $d_1, \dots, d_{n-1}$ . Again using the proof for correctness of the proposed protocol it follows that **Sender's Privacy** is guaranteed. And finally the **Receiver-servers collusion**, assuming that the Receiver has already calculated one secret and that a coalition of  $\Delta_2$  corrupt servers helps her to discover others. Because the Sender  $\mathcal{S}$  chooses the vectors  $B_0, B_1, \dots, B_{n-1}$  the information these  $\Delta_2$  corrupt servers posses (i.e. their collected shares) is insufficient to recover any of the secrets  $s_0, s_1 - s_0, \dots, s_{n-1} - s_0$ .

## 8 Conclusions

In this paper we have studied unconditionally secure distributed oblivious transfer protocols. We have presented an analysis of the threshold and general access structure model and some new results: impossibility result and lower bound for existence of one-round threshold  $DOT$  protocols, generalizing the result of Blundo et. al.; a threshold base construction implementing 1-out-of- $n$   $DOT$  achieving the proved lower bound for existence; a condition for existence of general access structure  $DOT$  scheme; a general access structure protocol implementing 1-out-of- $n$   $DOT$ .

## References

1. **D. Beaver, J. Feigenbaum, J. Kilian, P. Rogaway**, Locally Random Reductions: Improvements and Applications, *Journal of Cryptology* 10 (1), 1997, pp. 17-36.
2. **A. Beimel, Y. Ishai, T. Malkin**, Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing, *CRYPTO'2000, LNCS 1880*, 2000, pp. 55-73.
3. **M. Bellare, S. Micali**, Non-interactive Oblivious Transfer and Applications, *Advances in Cryptology: Crypto '89*, Springer-Verlag, 1990, pp. 547-559
4. **C. Blundo, P. D'Arco, A. De Santis, D. R. Stinson**, New Results on Unconditionally Secure Distributed Oblivious Transfer, to appear in *SAC'02*, 2002.
5. **G. Brassard, C. Crepeau, J.-M. Roberts**, All-or-Nothing Disclosure of Secrets, *CRYPTO'86, LNCS 263*, 1987, pp. 234-238.
6. **G. Brassard, C. Crepeau, M. Santha**, Oblivious Transfer and Intersecting Codes, *IEEE Trans. on Inf. Th.*, special issue in coding and complexity, Vol. 42, No. 6, 1996, pp. 1769-1780.
7. **B. Chor, O. Goldreich, E. Kushilevitz, M. Sudan** Private Information Retrieval, *Proc. 36th IEEE Symposium on Foundations of Computer Sciences (FOCS)*, 1995, pp. 41-50.

8. **R. Cramer**, Introduction to Secure Computation, *Lectures on Data Security - Modern Cryptology in Theory and Practice*, LNCS 1561, 1999, pp. 16-62.
9. **R. Cramer, I. Damgard, U. Maurer**, General Secure Multi-Party Computation from any linear secret sharing scheme, *EUROCRYPT'00*, LNCS 1807, 2000, pp. 316-335.
10. **R. Cramer, S. Fehr**, Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups, *Proc. CRYPTO 2002*, Springer Verlag LNCS 2442, pp.272-287.
11. **G. Di Crescenzo, Y. Ishai, R. Ostrovski**, Universal Service-Providers for Database Private Information Retrieval, *Proc. 17th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, 1998.
12. **P. D'Arco, D. R. Stinson**, Generalized Zig-zag Functions and Oblivious Transfer Reductions, *SAC 2001*, LNCS 2259, 2001, pp. 87-103.
13. **Y. Dodis, S. Micali**, Lower bounds for Oblivious Transfer Reduction, *EUROCRYPT'99*, LNCS 1592, 1999, pp. 42-54.
14. **S. Fehr, U. Maurer**, Linear VSS and Distributed Commitments Based on Secret Sharing and Pirwise Checks, *Proc. CRYPTO 2002*, Springer Verlag LNCS 2442, pp.565-580.
15. **S. Even, O. Goldreich, A. Lempel**, A Randomized Protocol for Signing Contracts, *Communications of the ACM* 28, 1985, pp. 637-647.
16. **Y. Gertner, Y. Ishai, E. Kushilevitz, T. Malkin**, Protecting Data Privacy in Private Information Retrieval Schemes, *Proc. 30th Annual ACM Symposium on Theory of Computing (STOC)*, 1998, pp. 151-160.
17. **Y. Gertner, S. Goldwasser, T. Malkin**, A Random Server Model for Private Information Retrieval or How to Achieve Information Theoretic PIR Avoiding Database Replication, *RANDOM* 1998, LNCS 1518, 1998, pp. 200-217.
18. **M. Karchmer, A. Wigderson**, On Span Programs, *Proc. of 8-th Annual Structure in Complexity Theory Conference*, San Diego, California, 18-21 May 1993. IEEE Computer Society Press, pp. 102-111.
19. **M. Naor, B. Pinkas, R. Sumner**, Privacy Preserving Auctions and Mechanism Design, ACM Conference on Electronic Commerce, 1999, available at <http://www.wisdom.weizmann.ac.il/naor/onpub.html>.
20. **M. Naor, B. Pinkas**, Distributed Oblivious Transfer, *ASIACRYPT'00*, 2000, pp. 205-219.
21. **V. Nikov, S. Nikova, B. Preneel, J. Vandewalle**, Applying General Access Structure to Proactive Secret Sharing Schemes, *Proc. of the 23rd Symposium on Information Theory in the Benelux*, May 29-31, 2002, Universite Catolique de Lovain (UCL), Lovain-la-Neuve, Belgium, pp.197-206, *Cryptology ePrint Archive*: Report 2002/141.
22. **M. Rabin**, How to Exchange Secrets by Oblivious Transfer, *Technical Memo TR-81*, Aiken Computation Laboratory, Harvard University, 1981.
23. **R. Rivest**, Unconditionally Secure Commitment and Oblivious Transfer Schemes Using Private Channels and a Trusted Initializer, manuscript, available at: <http://theory.lcs.mit.edu/rivest/publications.html>.
24. **W. Tzeng**, Efficient 1-out-of-n Oblivious Transfer Schemes, *Proc. PKC2002*, LNCS 2274, 2002, pp. 159-171.