Resource-efficient OT combiners with active security

Ignacio Cascudo[1], Ivan Damgård[2], Oriol Farràs[3], and Samuel Ranellucci[4]

[1]Aalborg University, ignacio@math.aau.dk
[2]Aarhus University, ivan@cs.au.dk
[3]Universitat Rovira i Virgili, oriol.farras@urv.cat
[4]University of Maryland and George Mason University, samuel@umd.edu

Abstract

An OT-combiner takes n implementations of the oblivious transfer (OT) functionality, some of which may be faulty, and produces a secure instance of oblivious transfer as long as a large enough number of the candidates are secure. More specifically, an OT-combiner is a 2-party protocol between Alice and Bob which can make several black-box calls to each of the n OT candidates. An adversary can corrupt one of the players and certain number of OT candidates, obtaining their inputs and (in the active case) full control of their outputs and we want the resulting protocol to be secure against such adversary.

In this work we consider perfectly (unconditionally, zero-error) secure OT-combiners and we focus on *minimizing the number of calls* to the candidate OTs.

First, we extend a result from Ishai et. al (ISIT 2014), constructing a perfectly secure single-use (one call per OT candidate) OT-combiner which is secure against active adversaries corrupting one player and at most a tenth of the OT candidates. Ishai et. al obtained the same result for passive adversaries.

Second, we consider a general asymmetric corruption model where an adversary can corrupt different sets of OT candidates depending on whether it is Alice or Bob who is corrupted. We give sufficient and necessary conditions on the existence of an OT combiner with a given number of calls to each server in terms of the existence of secret sharing schemes with certain access structures and share-lengths. This allows us for example to reduce the number of calls needed by known OT combiners, and in fact to determine the optimal number of calls, in some concrete situations even in the symmetric case, e.g. when there are three OT candidates and one of them is corrupted.

1 Introduction

1-out-of-2 bit oblivious transfer [EGL82] (OT) is a well-known cryptographic primitive between two parties, a sender and a receiver, in which the sender has two messages and the receiver chooses to learn one of them; in addition, two other guarantees hold, namely the sender does know which of her two messages was chosen by the receiver and the receiver obtains no information about the message that he did not choose to learn.

OT is a fundamental primitive for secure multiparty computation. In fact it is known that secure multiparty computation protocols can be entirely based on OT [Kil88, IPS08]. However, unconditionally secure two-party computation is not possible in the plain model, even if we only assume that one of the players may be passively corrupted. Hence, OT is likewise impossible to attain unless we assume the existence of some additional resource. Examples of such resources can be physical assumptions such as the existence of a noisy channel between the sender and the receiver [CK88], hardware tokens [GIS$^+$10], or the assumption that one of the parties have bounded memory [CCM98]. However, arguably the most studied resource for oblivious transfer is the assumption that the parties are computationally bounded. In this vein, oblivious transfer protocols have been proved secure assuming the computational hardness of different problems, for example hardness of factoring [Rab81], the DDH assumption [BM89, AIR01], hardness of decoding [DvdGMN08], the quadratic residuosity assumption, and worst-case lattice assumptions [PVW08].

Basing oblivious transfer on a computational assumption however forces each party to rely on the fact that the particular hardness assumption used has not been broken so far by the other party. This motivates the notion of an OT combiner which is a protocol between Alice and Bob that makes black-box calls to n candidate implementations of OT, and produces a single instance of OT which is secure as long as a certain number of the candidates were secure to start with. This way we can for example use candidate implementations which are secure under different hardness assumptions and the combiner will produce a secure OT as long as not too many of these hardness assumptions are broken.

An OT combiner can also be seen as a *server-aided* oblivious transfer protocol, where Alice and Bob have access to n servers S_1, S_2, …, S_n, each of them only implementing one of the candidates for the OT functionality. Alice and Bob can call each of the servers several times, where for each execution a server receives two bits from Alice and one bit from Bob, and outputs the resulting bit to Bob. Therefore, in particular, these servers do not communicate to (or even need to be aware of) each other. We adopt this view of OT combiners in what follows.

OT combiners were introduced in [HKN$^+$05] and further studied in [HIKN08, PW08, IMSW14]. In this paper we are interested in minimizing the number of calls to each of the servers. We take as starting point [IMSW14], where the authors focus on *single-use* OT combiners, in which each OT server is used only once. In their work, they consider an adversary that may corrupt Alice and up to t_A servers or Bob and up to t_B servers, thereby obtaining all information seen during the protocol by the corrupted servers and party. We will call this adversary a (t_A, t_B)-adversary. It is shown that for large enough n, there exists a single-use OT combiner which is perfectly secure against a *passive* (t_A, t_B)-adversary where $t_A = t_B = \Omega(n)$. More precisely this holds for $t_A = t_B = 0.11n$. Furthermore, they show that the existence of single-use OT combiners implies the existence of a certain secret sharing scheme whose privacy and reconstruction thresholds are related to t_A and t_B and where the shares are of constant size. By applying certain bounds on secret sharing over small alphabets [CCX13], they conclude among other facts that unconditionally secure single-use OT-combiners cannot exist when $t_A + t_B = n - O(1)$ (it is easy to show that perfectly secure OT combiners, single-use or not, cannot exist if $t_A + t_B \geq n$).

In this work, we first show a construction of single-use OT-combiners which are perfectly secure against an *active* adversary corrupting the same sets as in [IMSW14], namely:

Theorem 1.1. *For any large enough n, there exists an n-server single-use OT-combiner which is perfectly secure against an active $(0.11n, 0.11n)$-adversary.*

In fact this theorem is a special case of a more general result, that represents a much tighter link between secret sharing schemes and OT combiners.

In order to explain this result, we first need to consider a slightly more general adversary that can corrupt either Alice and a set $A \in \mathcal{A}$ of servers, or Bob and a set $B \in \mathcal{B}$ of servers. Here $\mathcal{A}$ and $\mathcal{B}$ are two adversary structures[1] in $\{1, \dots, n\}$. Now we say that $(\mathcal{A}, \mathcal{B})$ is an $\mathcal{R}_2$ pair of structures if there is at least one server $i \in \{1, \dots, n\}$ such that $i \notin A \cup B$. Our result is then as follows.

Theorem 1.2. *Let $\mathcal{A}$, $\mathcal{B}$ be adversary structures on the set of servers $\{S_1, \dots, S_n\}$. Suppose that the following conditions are true:*

- *$(\mathcal{A}, \mathcal{B})$ is an R2 pair of structures.*
- *There exists a secret sharing scheme $\mathcal{S}$ for n players with the following properties:*
  1. *It is a linear secret sharing scheme.*
  2. *The domain of secrets is $\{0, 1\}$ and the domain of the i-th share is $\{0, 1\}^{\ell_i}$, for $i = 1, \dots, n$.*
  3. *Every set $A \in \mathcal{A}$ is unqualified in $\mathcal{S}$ and for every set $B \in \mathcal{B}$, its complement $\overline{B}$ is qualified in $\mathcal{S}$.*

Then there exists a OT-combiner which is perfectly secure against any active $(\mathcal{A}, \mathcal{B})$-adversary and uses server S_i exactly ℓ_i times.

Therefore we can see that a single-use OT combiner will exist in the cases where an *ideal* (i.e. every share is one bit long) secret sharing scheme $\mathcal{S}$ exists satisfying all properties. Theorem 1.1 is then obtained by using secret sharing schemes coming from families of binary linear codes such that both them and their duals are on the Gilbert-Varshamov bound [CCG$^+$07] (see Section 5.3 for more details)

An interesting fact about Theorem 1.2 is that it is close to give a tight characterization of unconditionally secure OT combiners in terms of secret sharing schemes, since one can extend the arguments in [IMSW13] to prove the following result.

Theorem 1.3. *Let $\mathcal{A}$, $\mathcal{B}$ be adversary structures on the set of servers $\{S_1, \dots, S_n\}$. If there exists a perfectly secure OT-combiner which is secure against any active $(\mathcal{A}, \mathcal{B})$-adversary and uses server S_i exactly ℓ_i times, then:*

- *$(\mathcal{A}, \mathcal{B})$ is an R2 pair of structures.*
- *There exists a secret sharing scheme for n players with the following properties:*
  1. *The domain of secrets is $\{0, 1\}$ and the domain of the i-th share is $\{0, 1\}^{\ell_i}$, for $i = 1, \dots, n$.*
  2. *Every set $A \in \mathcal{A}$ is unqualified in $\mathcal{S}$ and for every set $B \in \mathcal{B}$, its complement $\overline{B}$ is qualified in $\mathcal{S}$.*

If we compare both Theorems 1.2 and 1.3 we see there is just one gap regarding sufficient and necessary conditions, namely that our construction from Theorem 1.2 requires a linear secret sharing scheme.

[1] An adversary (or anti-monotone) structure $\mathcal{A}$ is a family of subsets of in $\{1, \dots, n\}$ such that if $A \in \mathcal{A}$ and $A' \subseteq A$, then $A' \in \mathcal{A}$

1.1 Details and techniques

Our construction of an OT combiner showing Theorem 1.2 relies on the combination of two secret sharing schemes. The first one is the secret sharing scheme $\mathcal{S}$ assumed by the theorem, which is used by Bob in order to secret share his input among the servers. The other secret sharing scheme is a multi-secret sharing scheme Σ with some unusual properties, whose construction may be of independent interest. This will be used by Alice in order to secret share her inputs among the servers.

Such secret sharing scheme takes a 2-bit secret (m_0, m_1) and, in the simplified "single-use" case of our theorem (which is enough to show Theorem 1.1), splits it into $2n$ shares, indexed by pairs (i, j), where $i = 1, \dots, n$, and $j = 0, 1$. The secret sharing scheme is such that a set of participants of the form $\{(1, v_1), (2, v_2), \dots, (n, v_n)\}$ (where $v_i \in \{0, 1\}$) can reconstruct the message m_0 if and only if the bit-string $(v_1, \dots, v_n)$ belongs to some given vector space V, while it can reconstruct m_1 if and only if $(v_1, \dots, v_n)$ belongs to some affine space $\mathbf{t} + V$ for some given vector $\mathbf{t}$. Further, these sets are the only minimally qualified sets for each of the messages.

If that were the only requirements, the existence of such a secret sharing scheme would be guaranteed by known general results in secret sharing (where each coordinate m_0 and m_1 would then be independently shared with a secret sharing scheme with potentially exponentially long shares). But what makes the problem interesting is that for our problem we need the additional condition that *every share is one bit long*. Moreover, it is also necessary to exact some conditions preventing certain sets of shares from leaking correlations between m_0 and m_1 even if they give no information about either individual message. We show that we can achieve all these properties by a relatively simple construction.

With all these elements in hand, it is now easy to explain how our OT combiner works. Alice will use a secret sharing scheme as specified above where V is the set of all possible sharings of 0 in the scheme $\mathcal{S}$ used by Bob, and $\mathbf{t}$ is a sharing of 1 in $\mathcal{S}$. In this situation $\mathbf{t} + V$ is the set of all sharings of 1 in $\mathcal{S}$ by linearity of $\mathcal{S}$. She then sends the $(i, 0)$ and $(i, 1)$-th shares to the i-th server. If Bob has used $b_i, i = 1, ..., n$ as input for the servers, he will receive the shares of (m_0, m_1) with indices $(1, b_1), ..., (n, b_n)$. By the properties of the scheme Σ given that set of shares he can now reconstruct m_0 if $(b_1, \dots, b_n)$ was a sharing of 0 with $\mathcal{S}$, and m_1 if $(b_1, \dots, b_n)$ was a sharing of 1 with $\mathcal{S}$.

Of course this only shows the correctness of the protocol when Alice and Bob are honest. In particular, we need to take into account that Bob can corrupt a set $B \in \mathcal{B}$ of servers, obtaining both of Alice's shares corresponding to those servers. Furthermore, in the active case, he can also submit values that do not correspond to a valid sharing of a bit with $\mathcal{S}$. However, we show that even using both strategies simultaneously will not give him information about more than one of Alice's messages.

1.2 Related work

[HKN$^+$05] introduced the notion of OT combiners. Several different flavours are introduced; the notion we are considering in this paper corresponds to that they call third-party black-box combiners. They consider threshold security with $t_A = t_B = t$, and show that passively unconditionally secure OT combiners cannot exist for $n = 2$, $t = 1$. On the other hand, they give a concrete construction of a secure OT combiner for $n = 3$, $t = 1$ which makes 2 calls to each OT-candidate. In Section 8, we show how our construction can improve the number of server calls of this result, since we can construct an OT combiner which makes two calls to two of the servers, but only one to the other server. Furthermore, we can show that this is optimal.

In [HIKN08] OT-combiners are constructed from secure multiparty computation protocols. Again the threshold case with $t_A = t_B = t$ is considered. They show how to construct passively statistically secure OT combiners with $t = \Omega(n)$ which make $O(1)$ calls to each server. Furthermore they achieve constant production rate, meaning that their construction allows to produce $\Theta(n)$ instances of OT.

In [PW08] an oblivious linear function evaluation (OLFE) combiner is constructed where each server executes a single instance of OLFE and the construction achieves perfect security whenever $t_A + t_B < n$. OLFE is a functionality where Alice has as input two values a, b in a finite field $\mathbb{F}_q$ of q elements, Bob has as input $x \in \mathbb{F}_q$ and receives $ax + b$ as output. Even though OLFE is a generalization of OT (OT is equivalent to OLFE over $\mathbb{F}_2$), the construction in [PW08] requires $q > n$, since it uses Shamir secret sharing in order to share the players' inputs among the servers.

Finally, it is interesting to point out that [BI01] and [VV15] consider, in different contexts, secret sharing schemes with access structures that are somewhat related to the ones we need. Given a language $L \subseteq \{0,1\}^n$, their secret sharing schemes for $2n$ players have as minimally qualified subsets all those of the form $\{(1, v_1), (2, v_2), \ldots, (n, v_n)\}$ where $(v_1, v_2, \ldots, v_n) \in L$. However, both works also include the sets of the form $\{(i, 0), (i, 1)\}$ as minimally qualified.

1.3 Overview

Section 2 contains preliminaries on secret sharing and adversary structures, although we also introduce the notion of $\mathcal{R}_2$ pair. Section 3 describes our model. Section 4 gives a construction of a multi-secret sharing scheme with certain properties regarding its access structure; this will be the secret sharing scheme used by Alice in our construction. In Section 5 we show Theorem 1.2 in the case where $\mathcal{S}$ can be taken to be an ideal secret sharing scheme (i.e. every share is a bit long). This is enough to show Theorem 1.1. In Section 6 we show Theorem 1.2 in the general case. In Section 7 we show Theorem 1.3. In Section 8 we show how Theorems 1.2 and 1.3 can be used to determine the exact minimal number of calls that we need for a perfectly secure OT combiner in the case where we have 3 servers and one can be corrupted.

2 Preliminaries

2.1 $\mathcal{Q}_2$ structures and $\mathcal{R}_2$ pairs of structures

We denote by $\mathcal{P}_n$ the set $\{1, 2, \ldots, n\}$. Furthermore, $2^{\mathcal{P}_n}$ is the family of all subsets of $\mathcal{P}_n$. An adversary (or antimonotone) structure $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ is a family of subsets of $\mathcal{P}_n$ such that $\emptyset \in \mathcal{A}$ and for every $A \in \mathcal{A}$ and $B \subseteq A$ we have $B \in \mathcal{A}$.

Definition 2.1. *We say that an adversary structure $\mathcal{A}$ is $\mathcal{Q}_2$ if for all $A, B \in \mathcal{A}$, we have $A \cup B \neq \mathcal{P}_n$.*

Definition 2.2. *We say that a pair $(\mathcal{A}, \mathcal{B})$ of adversary structures is $\mathcal{R}_2$ if for all $A \in \mathcal{A}$, $B \in \mathcal{B}$, we have $A \cup B \neq \mathcal{P}_n$.*

$\mathcal{R}_2$ is a generalization of $\mathcal{Q}_2$. More precisely, the pair of adversary structures $(\mathcal{A}, \mathcal{A})$ is $\mathcal{R}_2$ if and only if $\mathcal{A}$ is $\mathcal{Q}_2$. However, there exist adversary structures $\mathcal{A}, \mathcal{B}$ such that neither $\mathcal{A}$ nor $\mathcal{B}$ are $\mathcal{Q}_2$, while the pair $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$. For example: $n = 4$, and $\mathcal{A}$ and $\mathcal{B}$ are the adversary structures with maximal sets $\{1, 2\}, \{3, 4\}$ in the case of $\mathcal{A}$, and $\{1, 3\}, \{2, 4\}$ in the case of $\mathcal{B}$.

Definition 2.3. *For an adversary structure $\mathcal{A}$, the dual adversary structure $\mathcal{A}^*$ is defined as follows: $A \in \mathcal{A}^*$ if and only if $\bar{A} \notin \mathcal{A}$, where $\bar{A} = \mathcal{P}_n \setminus A$.*

Lemma 2.4. *If $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$, then $\mathcal{B} \subseteq \mathcal{A}^*$ (consequently also $\mathcal{A} \subseteq \mathcal{B}^*$).*

Indeed, if $B \in \mathcal{B}$, then $\bar{B} \notin \mathcal{A}$ by $\mathcal{R}_2$, and then $B \in \mathcal{A}^*$ by definition of the dual adversary structure.

2.2 Secret sharing

Our protocols rely heavily on secret sharing, a well-known cryptographic primitive introduced by Shamir [Sha79] and, independently, Blakley [Bla79]. We recall some terminology and results which will be needed later.

A secret sharing scheme for the set of players $\mathcal{P}_n$ is given by a probabilistic algorithm $\texttt{Share}_{\mathcal{S}}$ that takes as input a secret s and outputs values $a_1, a_2, \ldots, a_n$ known as shares. The vector $(a'_1, a'_2, \ldots, a'_n)$ is called a sharing of s if on input s $\texttt{Share}_{\mathcal{S}}$ outputs the values a'_i as shares with non-zero probability.

We say that a set $A \subseteq \mathcal{P}_n$ is unqualified if for any secret s and any sharing $(a_1, a_2, \ldots, a_n)$ for it, the vector $(a_i)_{i \in A}$ gives no information about the secret, i.e., the conditional probability that $\texttt{Share}_{\mathcal{S}}$ outputs the values $(a_i)_{i \in A}$ conditioned to the secret being s is the same as the probability of the same event conditioned to the secret being s'. Note that the family $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ of all unqualified sets of $\mathcal{S}$ is an adversary structure. We say that a set $A \subseteq \mathcal{P}_n$ is qualified if for any secret s and any sharing $(a_1, a_2, \ldots, a_n)$ for it, the the vector $(a_i)_{i \in A}$ uniquely determines the secret, i.e. there is a unique secret s for which $\texttt{Share}_{\mathcal{S}}$ can output those values. The family of all qualified sets is called the access structure of $\mathcal{S}$. We say that a secret sharing scheme is perfect if every set $A \subseteq \mathcal{P}_n$ is either qualified or unqualified (there are no sets of shares which give partial information about the secret).

We also define $\texttt{Reconstruct}_{\mathcal{S}}$, an algorithm that takes as input a set of pairs $\{(i, a_i) : i \in A\}$ where $A \subseteq \mathcal{P}_n$ and outputs s if A is a qualified set for $\mathcal{S}$ and the values $\{a_i : i \in A\}$ are part of a valid sharing of the secret s, and $\perp$ otherwise.

Let $\mathbb{F}$ be a finite field. A linear secret sharing scheme $\mathcal{S}$ (over $\mathbb{F}$), LSSS for short, is a secret sharing scheme where the space of secrets is a vector space $\mathbb{F}^{\ell_0}$, the space of the i-th shares is $\mathbb{F}^{\ell_i}$ for $i = 1, \ldots, n$, and there exists as integer e and a map $M : \mathbb{F}^{\ell_0 + e} \to \mathbb{F}^{\ell_1} \times \cdots \times \mathbb{F}^{\ell_n}$ such that $\texttt{Share}_{\mathcal{S}}$ consists in choosing a uniformly random vector $\mathbf{u} \in \mathbb{F}^e$ and outputting $M(s, \mathbf{u})$ as shares. We denote by $[s, \mathbf{u}]_{\mathcal{S}} \in \mathbb{F}^{\ell}$ this sharing, where $\ell = \sum_{i=1}^{n} \ell_i$. Given a set $A \subseteq \mathcal{P}_n$ we use $[s, \mathbf{u}]_{\mathcal{S}}^{(A)}$ to denote the vector consisting only of the shares corresponding to A. When we do not need to make the randomness explicit, then we write $[s]_{\mathcal{S}}$ and $[s]_{\mathcal{S}}^{(A)}$. Moreover, we say that ℓ is the complexity of $\mathcal{S}$. We note that $\texttt{Share}_{\mathcal{S}}$ runs in polynomial time in ℓ. It is also satisfied that the set of possible sharings is a vector space and that given $\lambda_1 [s_1, \mathbf{u_1}]_{\mathcal{S}} + \lambda_2 [s_2, \mathbf{u_2}]_{\mathcal{S}} = [\lambda_1 s_1 + \lambda_2 s_2, \lambda_1 \mathbf{u_1} + \lambda_2 \mathbf{u_2}]_{\mathcal{S}}$, i.e. a linear combination of sharings is a sharing for the same linear combination applied to the secrets. It is easy to see that this implies that $\texttt{Reconstruct}_{\mathcal{S}}$, on input a qualified set A and a set of shares for it, acts by applying a linear function to these shares.

We need a few facts about when sets are qualified and unqualified in a linear secret sharing scheme. First, consider the case $\ell_0 = 1$, where the secret is just an element in $\mathbb{F}$. In that case a LSSS is perfect, and we have:

Lemma 2.5. *Let $\mathcal{S}$ be a LSSS with secrets in $\mathbb{F}$. A set $A \subseteq \mathcal{P}_n$ is unqualified if and only if there exists a vector $\mathbf{u}$, such that $[1, \mathbf{u}]_{\mathcal{S}}^{(A)} = \mathbf{0}$, i.e., if we share the secret 1 using randomness $\mathbf{u}$, the shares corresponding to A are all zero. It is qualified otherwise.*

This can be easily derived by taking into account that, if the condition above is satisfied, then$[s, \mathbf{t}]_{\mathcal{S}}$ and $[s', \mathbf{t}']_{\mathcal{S}} = [s, \mathbf{t}]_{\mathcal{S}} + (s' - s)[1, \mathbf{u}]_{\mathcal{S}}$ are sharings of s, s' such that all the shares in A coincide.

Now suppose that in addition $\mathbb{F} = \mathbb{F}_2$, so we are dealing with binary LSSS and that every share is one bit long, i.e., $\ell_i = 1$. Since given a qualified set A, the reconstruction algorithm in a LSSS consists of applying a linear function on the corresponding shares, under the conditions above it must hold that the secret equals the sum of the shares of a fixed subset $A' \subseteq A$. Therefore we can characterize the minimally qualified set (those qualified sets such that none of their subsets are qualified) as follows

Lemma 2.6. *Let $\mathcal{S}$ be a LSSS with secrets in $\mathbb{F}_2$ and shares in $\mathbb{F}_2$. A set A is minimally qualified if and only if for any secret $s \in \mathbb{F}_2$ and any sharing $(a_1, a_2, \ldots, a_n) = [s]_{\mathcal{S}}$, we have that $s = \sum_{i \in A} a_i$.*

In this work it will also be essential to understand LSSSs where $\ell_0 = 2$ and $\mathbb{F}$ is the binary field $\mathbb{F}_2$. In general, if $\ell_0 > 1$, the situation is more complicated than in the case $\ell_0 = 1$ since there may be sets $A \subseteq \mathcal{P}_n$ which can obtain partial information about the secret. The generalization of Lemma 2.5 is as follows. Let $T_A \subseteq \mathbb{F}^{\ell_0}$ be the set of secrets s such that there exists $\mathbf{u}$ with $[s, \mathbf{u}]_{\mathcal{S}}^{(A)} = \mathbf{0}$. Then for any secret m, when given $[m]_{\mathcal{S}}^{(A)}$, any element in $m + T_A$ has the same probability of being the secret and any element not in $m + T_A$ can be ruled out. Furthermore, T_A is always a vector space. In the case $\ell_0 = 2$, $\mathbb{F} = \mathbb{F}_2$, this means that a set A can be either qualified, unqualified or learn one bit of information, about the secret $m = (m_0, m_1)$ and this partial information can be of three types: either it learns one coordinate m_0 and has no information about the other m_1, or viceversa, or it learns $m_0 + m_1$ and nothing else. A LSSS Σ with secrets (m_0, m_1) in $\mathbb{F}_2^2$ induces a perfect LSSS Σ_0 for the secret m_0 (by considering m_1 as randomness) and similarly, perfect LSSSs Σ_1 and Σ_2 for m_1 and $m_0 + m_1$ respectively. Therefore we can talk about qualified sets and unqualified sets for m_0 (resp. m_1, $m_0 + m_1$) and we will use Lemma 2.5 and Lemma 2.6 for these individual secrets later on. We are therefore seeing Σ as a multi-secret sharing scheme (in a multi-secret sharing scheme[JMO93] several secret values are distributed among a set of users, and each secret may have different qualified subsets). Moreover, we can clearly define a reconstruction algorithm for the individual secrets m_0 and m_1, which we call $\texttt{Reconstruct}_{\Sigma}^{0}$ and $\texttt{Reconstruct}_{\Sigma}^{1}$ respectively.

As for the existence of LSSS, it is well known [ISN87] that every adversary structure is the adversary structure of a LSSS.

Theorem 2.7. *For every finite field $\mathbb{F}$ and integer $\ell_0 \geq 1$ and for every adversary structure $\mathcal{A}$ there exists a perfect LSSS $\mathcal{S}$ with secrets in $\mathbb{F}^{\ell_0}$ and adversary structure $\mathcal{A}$.*

In general the complexity of the LSSS $\mathcal{S}$ constructed with the methods used in [ISN87] is exponential in n. We say that a LSSS is ideal if $\ell_0 = 1$ and $\ell_i = 1$ for all i. The complexity of an ideal LSSS is n, which is smallest possible. Given a field $\mathbb{F}$ and an adversary structure $\mathcal{A}$, it is not necessarily true that there exists an ideal LSSS over $\mathbb{F}$ with $\mathcal{A}$ as its adversary structure.

3 OT-combiners

We describe our model in more detail. Alice has a pair of inputs $m_0, m_1 \in \{0, 1\}$ and Bob has an input a selection bit $b \in \{0, 1\}$. They execute a protocol π whose goal is to implement functionality $\mathcal{F}_{OT}$ securely (in the presence of an adversary which we specify below) on those inputs. The protocol π consists only of local computations by each of the players and oracle calls to servers $S_1, \ldots, S_n$ (in particular, we do not need a direct communication channel between Alice and Bob). If the server S_i is not corrupted, then it executes a copy of the functionality $\mathcal{F}_{OT}$ and may be called several times. Each time a server is called, it receives a new pair of inputs $x_0, x_1 \in \{0, 1\}$ from Alice and c from Bob, and executes the functionality $\mathcal{F}_{OT}$ on these inputs, therefore outputting the message x_c towards Bob.

We consider a static adversary Adv characterized by a pair of adversary structures $(\mathcal{A}, \mathcal{B})$ each contained in $2^{\{S_1, \ldots, S_n\}}$, which we call an $(\mathcal{A}, \mathcal{B})$-adversary. Such adversary can corrupt, before the protocol starts, either Alice and a set of servers $A \in \mathcal{A}$ or Bob and a set of servers $B \in \mathcal{B}$. If the adversary is passive, then it obtains all information seen by the corrupted party and servers during the protocol, but cannot make them deviate from the protocol. If the adversary is active, it can in addition make the corrupted player and servers deviate arbitrarily from the protocol.

Functionality $\mathcal{F}_{OT}$

1. On input $(\texttt{transfer}, b)$ from Bob, send $(\texttt{ready})$ to Alice.
2. On input $(\texttt{send}, m_0, m_1)$ from Alice, if $(\texttt{transfer}, b)$ has been received previously from Bob, send $(\texttt{sent}, m_b)$ to Bob.

Figure 1: Functionality $\mathcal{F}_{OT}$

In this conditions, we say that the protocol π is an n-server OT-combiner secure against Adv if it securely implements the functionality $\mathcal{F}_{OT}$ in the presence of this adversary. In this paper we will prove security using the Universal Composability framework [Can01], see [CDN15] for more information.

Let $1 \leq t_A, t_B \leq n$. If there exist $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A}$ contains all subsets of size t_A of $\{1, \dots, n\}$ and $\mathcal{B}$ contains all subsets of size t_B of $\{1, \dots, n\}$ and if π is an n-server OT-combiner secure against any $(\mathcal{A}, \mathcal{B})$-adversary, then we say that π is an n-server OT-combiner secure against a (t_A, t_B)-adversary.

4 A multi-secret sharing scheme

As we mentioned in Section 1.1, our OT combiners rely on the combination of two linear secret sharing schemes $\mathcal{S}$ and Σ. $\mathcal{S}$ is given by the statement of Theorem 1.2 and is used by Bob. The secret sharing scheme Σ, used by Alice, is a multi-secret sharing scheme satisfying a number of properties that we need in order to achieve security of our combiner.

In this section, we abstract the properties that we will need for Σ, and we give a construction achieving these properties. How this will play a role in our OT-combiners will become apparent in the next sections.

Proposition 4.1. *Let ℓ be an integer, $V \subseteq \mathbb{F}_2^\ell$ a vector subspace, $\mathbf{t} \in \mathbb{F}_2^\ell$ be a vector such that $\mathbf{t} \notin V$ and let W be the affine space $W = \mathbf{t} + V$. Finally for $I \subseteq \{1, \dots, \ell\}$ let $\mathbf{e}_I \in \mathbb{F}_2^\ell$ denote the vector with 1's in the I-coordinates and 0's in the rest.*

Then the linear secret sharing scheme Σ for 2ℓ players (indexed by pairs (i, j)) with secrets in $\{0,1\}^2$ and shares in $\{0,1\}$, given in Figure 2, is such that the following properties hold:

1. *The minimally qualified sets A for reconstructing the first coordinate m_0 of the secret are exactly the sets of the form $A = \{(i, a_i) : i = 1, \dots, n, (a_1, \dots, a_n) \in V\}$.*
2. *The minimally qualified sets A' for reconstructing the second coordinate m_1 of the secret are exactly the sets of the form $A' = \{(i, a_i) : i = 1, \dots, n, (a_1, \dots, a_n) \in W\}$.*
3. *The minimally qualified sets A'' for reconstructing the sum $m_0 + m_1$ are those of the form $A'' = \{(i, c) : i \in H, c = 0, 1\}$ where H is such that $\mathbf{e}_H \in W$ and $\mathbf{e}_{H'} \notin W$ for $H' \subsetneq H$.*

Before starting with the proof, we need some definitions.

Let U be the vector space spanned by the set $V \cup \{\mathbf{t}\}$. Note $U = V + W$. We define

$$Z_0 = U^\perp = \{\mathbf{h} \in \mathbb{F}_2^\ell : \mathbf{h} \in V^\perp, < \mathbf{t}, \mathbf{h} >= 0\}$$

and

$$Z_1 = \{\mathbf{h} \in \mathbb{F}_2^\ell : \mathbf{h} \in V^\perp, < \mathbf{t}, \mathbf{h} >= 1\}.$$

Note since $\mathbf{b} \notin V$, then Z_1 is non-empty and $Z_1 = Z_0 + \mathbf{g}$ for some $\mathbf{g}$ such that $< \mathbf{t}, \mathbf{g} >= 1$.

We also need the following lemma, which is a basic fact of linear algebra.

The multi-secret sharing scheme Σ

Let $V^{\perp}$ be the orthogonal space to V, i.e.,

$$V^{\perp} = \{\mathbf{h} \in \mathbb{F}_2^{\ell} : \langle \mathbf{v}, \mathbf{h} \rangle = 0 \text{ for all } \mathbf{v} \in V\}.$$

To share $(m_0, m_1) \in \mathbb{F}_2^2$.

- Sample uniformly at random $r_1, \ldots, r_{\ell-1} \in \mathbb{F}_2$ and let $r_\ell = m_0 - \sum_{i=1}^{\ell-1} r_i$.
- Sample $\mathbf{h} = (h_1, h_2, \ldots, h_\ell)$ uniformly at random in the space
$$\{\mathbf{h} \in \mathbb{F}_2^{\ell} : \mathbf{h} \in V^{\perp}, < \mathbf{t}, \mathbf{h} >= m_0 + m_1\}.$$
- Send $a_{(i,j)} = r_i + j h_i \in \mathbb{F}_2$ to participant (i, j)

Figure 2: The multi-secret sharing scheme Σ

Lemma 4.2. *For every $\mathbf{u} \notin U$, the random variable $< \mathbf{u}, \mathbf{h} >$, where $\mathbf{h}$ is chosen uniformly at random in Z_0 (resp. Z_1), is uniformly distributed in $\mathbb{F}_2$.*

Now we can proceed with the proof of Proposition 4.1

Proof of Proposition 4.1. Clearly Σ is linear, since a fixed linear combination of the sharings is a sharing for the same linear combination applied to the secrets. Nevertheless we can also make the linearity of the construction more explicit by showing how the shares are constructed as a linear function of the secret (m_0, m_1) and a uniform random vector in some space $\mathbb{F}_2^e$, as follows. Note that $V^{\perp}$ is a vector subspace. The set Z_0 is also a vector subspace which will have a basis $\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(s)}\}$.

A uniformly random element in $\{\mathbf{h} \in \mathbb{F}_2^{\ell} : \mathbf{h} \in V^{\perp}, < \mathbf{t}, \mathbf{h} >= m_0 + m_1\}$ can be then sampled by sampling independent uniform random elements $d_1, \ldots, d_s \in \mathbb{F}_2$ and outputting $d_1\mathbf{z}^{(1)} + \cdots + d_s\mathbf{z}^{(s)} + (m_0 + m_1)\mathbf{g}$. The elements h_i in our construction are simply the coordinates $d_1 z_i^{(1)} + \cdots + d_s z_i^{(s)} + (m_0 + m_1) g_i$. Therefore, the shares can be written as a linear combination of uniformly random elements $r_1, \ldots, r_{\ell-1}, d_1, \ldots, d_s \in \mathbb{F}_2$ and the values m_0, m_1.

Now we need to argue about the access structure of the secret sharing schemes for the different pieces of information m_0, m_1 and $m_0 + m_1$.

By Lemma 2.6, in the conditions of these scheme (linear, binary, every share is a bit) a set is minimally qualified for m_0 (resp. m_1, $m_0 + m_1$) if and only if the corresponding shares always sum up to m_0 (resp. m_1, $m_0 + m_1$) and there is no stricty smaller subset satisfying the same.

Fix $A \subseteq \{1, 2, \ldots, \ell\} \times \{0, 1\}$ a set of indices. We define two sets $I_1, I_2 \subseteq \{1, 2, \ldots, \ell\}$ as follows:

$$I_1 = \{i : \text{exactly one of } (i, 0) \text{ and } (i, 1) \text{ is in } A\}$$

and

$$I_2 = \{i : (i, 1) \in A\}.$$

Then

$$\sum_{(i,j) \in A} a_{(i,j)} = \sum_{i \in I_1} r_i + \sum_{i \in I_2} h_i = \sum_{i \in I_1} r_i + < \mathbf{e}_{I_2}, \mathbf{h} >$$

where $\mathbf{e}_{I_2}$ is the vector with 1's in the positions of I_2 and 0's in the rest.

Note that if $I_1 \neq \emptyset, \{1, \ldots, \ell\}$, then $\sum_{i \in I_1} r_i$ is uniformly distributed in $\mathbb{F}_2$ over the choice of the r_i's. Furthermore, $\sum_{i \in I_1} r_i$ is clearly independent from $< \mathbf{e}_{I_2}, \mathbf{h} >$. Hence the sum $\sum_{(i,j) \in A} a_{(i,j)}$ is uniformly distributed in $\mathbb{F}_2$.

Likewise if $\mathbf{e}_{I_2} \notin U = V \cup W$ then $< \mathbf{e}_{I_2}, \mathbf{h} >$ is uniformly distributed in $\mathbb{F}_2$ by Lemma 4.2 (regardless of whether $m_0 + m_1 = 0$ or $m_0 + m_1 = 1$). Therefore, the only cases where A can be minimally qualified for either m_0, m_1, $m_0 + m_1$ are the following:

- $I_1 = \{1, \ldots, \ell\}$, $\mathbf{e}_{I_2} \in V$. This case corresponds to $A = \{(1, b_1), (2, b_2), \ldots, (n, b_n)\}$ where $(b_1, b_2, \ldots, b_n) = \mathbf{e}_{I_2} \in V$. Moreover $\sum_{(i,j)\in A} a_{(i,j)} = m_0 + < \mathbf{h}, \mathbf{e}_{I_2} >= m_0$, so this set is minimally qualified for m_0, since clearly there cannot be smaller subsets satisfying the same property.

- $I_1 = \{1, \ldots, \ell\}$, $\mathbf{e}_{I_2} \in W$. This case corresponds to $A = \{(1, b_1), (2, b_2), \ldots, (n, b_n)\}$ where $(b_1, b_2, \ldots, b_n) = \mathbf{e}_{I_2} \in W$. Moreover $\sum_{(i,j)\in A} a_{(i,j)} = m_0 + < \mathbf{h}, \mathbf{e}_{I_2} >= m_1$, so this set is minimally qualified for m_1, since clearly there cannot be smaller subsets satisfying the same property.

- $I_1 = \emptyset$, $\mathbf{e}_{I_2} \in V$: in this case, $A = \{(i, 0) : i \in I_2\} \cup \{(i, 1) : i \in I_2\}$. However $\sum_{(i,j)\in A} a_{(i,j)} =< \mathbf{h}, \mathbf{e}_{I_2} >= 0$, so this set is not minimally qualified for any of the secrets.

- $I_1 = \emptyset$, $\mathbf{e}_{I_2} \in W$: in this case, again $A = \{(i, 0) : i \in I_2\} \cup \{(i, 1) : i \in I_2\}$. Now $\sum_{(i,j)\in A} a_{(i,j)} =< \mathbf{h}, \mathbf{e}_{I_2} >= m_0 + m_1$, so this set is minimally qualified for $m_0 + m_1$ unless there is a smaller subset $I_2' \subseteq I_2$ such that $e_{I_2'} \in W$.

5 Construction of OT-combiners when $\mathcal{S}$ is ideal

In this section we will show Theorem 1.2, under the additional assumption that the secret sharing scheme $\mathcal{S}$ is also ideal. That is, we show:

Theorem 1.2, case $\mathcal{S}$ ideal. *Let $\mathcal{A}$, $\mathcal{B} \subseteq 2^{\mathcal{P}}_n$ be adversary structures such that $(\mathcal{A}, \mathcal{B})$ is a R2 pair. Suppose there exists a linear secret sharing scheme $\mathcal{S}$ for n players where the secret is in $\{0, 1\}$ and* every share is in $\{0, 1\}$*, and such that every set $A \in \mathcal{A}$ is unqualified in $\mathcal{S}$ and the complement $\overline{B}$ of every set $B \in \mathcal{B}$ is qualified in $\mathcal{S}$.*

Then there exists a single-use *OT combiner which is perfectly secure against any active $(\mathcal{A}, \mathcal{B})$-adversary.*

This result is enough to show Theorem 1.1, which is proven at the end of this section.

5.1 The protocol

Our protocol works as follows: Bob computes a secret sharing of his input b with the ideal linear secret sharing scheme $\mathcal{S}$ promised above, therefore creating n shares b_i, each of which is a bit since the scheme is ideal. On the other hand, Alice will secret share her input (m_0, m_1) with a secret sharing scheme Σ that is defined as follows: Σ is the secret sharing scheme given by Proposition 4.1 where $\ell = n$, V is the set of all possible sharings $[0, \mathbf{u}]_{\mathcal{S}}$ of 0 with $\mathcal{S}$ (which is a vector space because $\mathcal{S}$ is linear) and $\mathbf{t}$ will be one sharing of 1 with $\mathcal{S}$ (for example $\mathbf{t} = [1, \mathbf{0}]_{\mathcal{S}}$). By linearity W is the set of all possible sharings of 1.

Now Alice an Bob call each OT server once, the inputs to the i-th server being $a_{(i,0)}$ and $a_{(i,1)}$, in this order, on Alice's side, and b_i on Bob's side. Assuming there is no active corruption, Bob will receive $a_{(i,b_i)}$ from the servers. By definition of Σ he has enough information to reconstruct m_b by running the corresponding reconstruction algorithm (if the reconstruction fails, because Alice's shares were malformed, Bob outputs 0 by default).

Proposition 5.1. *If Alice and Bob follow the protocol semi-honestly, then π_{OT} implements OT with perfect correctness.*

Oblivious transfer protocol π_{OT}

Let (m_0, m_1) be Alice's input and b be Bob's input.

1. Local computation:
   Alice creates a sharing $[(m_0, m_1)]_\Sigma = (a_{(i,j)})_{(i,j)\in\mathcal{P}_{n,2}}$ of her input.
   Bob creates a sharing $[b]_\mathcal{S} = (b_1, \dots, b_n)$ of his input. Note that each $b_i \in \{0, 1\}$ because $\mathcal{S}$ is ideal.
2. Use of the OT servers:
   For $i \in \{1, \dots, n\}$, Alice and Bob use server S_i to execute an OT with inputs $(a_{i,0}, a_{i,1})$ for Alice and b_i for Bob. Let y_i denote the output of Bob.
3. Local computation: If $b = 0$, Bob constructs m'_0 by applying
   $$\texttt{Reconstruct}^0_\Sigma(\{((i, b_i), y_i) : i \in \mathcal{P}_n\}).$$
   Similarly, if $b = 1$, Bob constructs m'_1 by applying
   $$\texttt{Reconstruct}^1_\Sigma(\{(i, b_i), y_i) : i \in \mathcal{P}_n\}).$$
   In any of the cases, if the reconstruction fails, output 0. Otherwise output the reconstructed m'_b.

Figure 3: Protocol π_{OT}

Proof. If Alice and Bob follow the protocol (semi-)honestly, at the end of the protocol Bob will have received all values $m_b^{(i,b_i)}, i = 1, \dots, n$, for some sharing $[b]_\mathcal{S} = (b_1, \dots, b_n)$. By Proposition 4.1 $\{(1, b_1), \dots, (n, b_n)\}$ is qualified for reconstructing m_b (because $(b_1, \dots, b_n) \in V$ if $b = 0$ and $(b_1, \dots, b_n) \in W$). □

5.2 Security

In order to guarantee the privacy of Alice's input, the first thing that we need to observe is that Bob does not learn m_b from $a_{(i,b_i)}$ if $(b_1, \dots, b_n)$ is not a valid sharing of b with $\mathcal{S}$, since in that case $\{(1, b_1), \dots, (n, b_n)\}$ is not qualified for m_b by Proposition 4.1. However, this only guarantees privacy against a very weak semi-honest adversary corrupting Bob and no servers. Note that, first of all, the adversary can corrupt some set $B \in \mathcal{B}$ of servers, thereby obtaining both $a_{(i,0)}$ and $a_{(i,1)}$ for all $i \in B$. Moreover, if the adversary is malicious, it can also make Bob submit values b_i such that $(b_1, \dots, b_n)$ is not a valid sharing $[b]_\mathcal{S}$. Finally, remember that in Section 2.2 we argued that given an ideal LSSS with secrets in $\mathbb{F}_2$, like it is the case with Σ, it may in principle happen that some sets of shares allow to reconstruct $m_0 + m_1$ even if they do not get any information about the individual m_0 and m_1. Therefore we also need to ensure that these cases will not happen in our problem.

We show how the properties we have guaranteed in Proposition 4.1 take care of all these and prevent the potentially malicious Bob from learning other information than he should.

Proposition 5.2. *Suppose $(\mathcal{A}, \mathcal{B})$ is an $\mathcal{R}_2$ pair of adversary structures and $\mathcal{S}$ and Σ are defined as above. Let (m_0, m_1) be shared with Σ. Fix $B \in \mathcal{B}$ and $(b'_1, \dots, b'_n) \in \mathbb{F}_2^n$, and define the set of indices*

$$\mathcal{H} = \{(i, b'_i) : i \in \overline{B}\} \cup \{(i, j) : i \in B, j \in \{0, 1\}\}.$$

Then:

- *If the set $\{b'_i : i \in \overline{B}\}$ is not part of any sharing $[c]_{\mathcal{S}}$ for any $c \in \{0,1\}$ then the values $a_{(i,j)}, (i,j) \in I'$ give no information about the pair (m_0, m_1).*
- *If the set $\{b'_i : i \in \overline{B}\}$ is a part of a sharing $[c]_{\mathcal{S}}$ of some $c \in \{0,1\}$ then the values $a_{(i,j)}, (i,j) \in I'$ give full information about m_c but no information about m_{1-c}.*

Proof. By the considerations in Section 2.2, we know that in principle a set of shares could either be unqualified (give no information about (m_0, m_1)), qualified (give full information) or give partial information, which in turn can be of three types: either it gives information about one of the coordinates m_d and no information about m_{1-d} or it could give information about $m_0 + m_1$ and nothing else. On the other hand, Proposition 4.1 describes the minimally qualified sets for m_0, m_1 and $m_0 + m_1$.

We show first that the set $\mathcal{H}$ is not qualified for $m_0 + m_1$ in any case. If that were the case, then there would exist a set $I \subseteq \mathcal{P}_n$ such that $\mathcal{H}$ would contain all indices of the form $(i, 0)$, $(i, 1)$ with $i \in I$ and such that $\mathbf{e}_I \in \mathbb{F}_2^n$ is a sharing of 1 with $\mathcal{S}$. $\mathcal{H}$ contains both $(i, 0)$ and $(i, 1)$ exactly for those $i \in B$. But assume there existed a $I \subseteq B$ such that $\mathbf{e}_I \in \mathbb{F}_2^n$ were a sharing of 1. Now we get a contradiction as follows: from the assumptions, $\overline{B}$ is qualified in $\mathcal{S}$. Therefore by linearity of $\mathcal{S}$ there cannot be a sharing of 1, $[1]_{\mathcal{S}}$, such that $[1]_{\mathcal{S}}^{\overline{B}} = \mathbf{0}$. But on the other hand $\mathbf{e}_I \in \mathbb{F}_2^n$ is a sharing of 1 which satisfies that $[1]_{\mathcal{S}}^{\overline{I}}$ is zero, and since $\overline{B} \subseteq \overline{I}$ both statements are contradictory.

Now note that the minimally qualified sets for m_0 (resp. m_1) are those of the form $\{(1, b_1), \ldots, (n, b_n)\} \subseteq \mathcal{P}_{n,2}$ where $(b_1, \ldots, b_n)$ is a sharing of 0 (resp. 1) with $\mathcal{S}$. However under the assumptions, $\mathcal{H}$ cannot contain any such set. Indeed if that were the case then necessarily $b_i = b'_i$ for $i \in \overline{B}$ but then $\{b'_i : i \in \overline{B}\}$ would be part of a sharing $[0]_{\mathcal{S}}$ (respectively $[0]_{\mathcal{S}}$), contradicting the assumption.

□

These elements are enough to formally show the security of our construction.

THEOREM 5.3. *The protocol π_{OT} UC-implements the functionality $\mathcal{F}_{OT}$ in the presence of an $(\mathcal{A}, \mathcal{B})$-adversary.*

Proof. **Alice honest, Bob malicious:** The first thing to notice is that the values b'_i that malicious sends to the servers do not need to be a correct sharing in $\mathcal{S}$ of any bit. The idea of the proof is to have the simulator extract the environment's input by applying the reconstruction procedure of $\mathcal{S}$ to the b'_i. By the $\mathcal{R}_2$ property, these shares can be consistent with at most one b. If indeed they are a valid sharing of some b, this is sent to the functionality and m_b is received; the simulator then sets m_{1-b} at random and generates shares for both messages. Otherwise, the simulator generates shares for random m_0, m_1. After that the simulator generates the remaining information as it would happen in the protocol. Proposition 5.2 will be used to guarantee that indeed this is possible.

More precisely, the simulator is constructed as follows. We will suppose without loss of generality that corrupted servers act as a dummy adversary. Let B denote the set of corrupted servers.

First, Sim awaits $(\texttt{ready}, i)$ for $i \in B$ and that the environment has sent b'_i for each $i \in \overline{B}$. Then it executes $\texttt{Reconstruct}_{\mathcal{S}}(\{(i, b'_i) : i \in \overline{B}\})$. If the reconstruction fails then Sim chooses random messages $\tilde{m}_0$, $\tilde{m}_1$. If the reconstruction succeeds, let b be its output; then Sim sends the command ($\texttt{transfer}$, b) to $\mathcal{F}_{OT}$, receives message $(\texttt{sent}, m_b)$ and sets $\tilde{m}_b := m_b$; it selects a random message $\tilde{m}_{1-b} \in \mathcal{M}$.

In any case, Sim generates a sharing $(a_{(i,j)})_{(i,j) \in \mathcal{P}_{n,2}} = [(\tilde{m}_0, \tilde{m}_1)]_{\Sigma}$.

Finally, in parallel Sim sends the following to the environment: for each $i \in \overline{B}$, it sends $a_{(i,b'_i)}$, and for each $i \in B$, it sends the entire vectors $a_{(i,0)}$, $a_{(i,1)}$.

We need to prove now that the distribution of these values is indistinguishable from the ones obtained in the interaction with the actual protocol. We should first note that since the set $\overline{B}$ is qualified for $\mathcal{S}$, the values $\{b'_i : i \in \overline{B}\}$ cannot be part of both a sharing $[0]_{\mathcal{S}}$ and a sharing $[1]_{\mathcal{S}}$. Using Proposition 5.2, this implies that the distribution of the set of shares $(\tilde{m}_0)_{(i,j)}$, $(\tilde{m}_1)_{(i,j)}$, for $i \in B$ and $j \in \{0,1\}$ and $(\tilde{m}_0)_{(i,b'_i)}), (\tilde{m}_1)_{(i,b'_i)})$ for $i \in \overline{B}$ obtained in the simulation is the same as the corresponding distribution in the actual protocol.

Alice malicious, Bob honest: The simulation in this case is slightly tricky, since a potential problem of the protocol is that Alice can generate inconsistent shares which make Bob's output dependent on his selections (that is, on the random sharing of his input). We can show, perhaps surprisingly, that this does not affect the security of the protocol. Essentially, the simulator will generate one sharing for $b = 0$ and one for $b = 1$ such that the shares corresponding to the corrupted servers coincide. The simulator will then construct the value that a receiver would construct for each of these two sharings and will send these values to the functionality. This results in a view in the ideal world which is perfectly indistinguishable from the real world, due to the privacy for the set of corrupted servers.

Let $A \in \mathcal{A}$ be the set of corrupted servers. The simulator works as follows: Upon receiving (`ready`) from the ideal functionality $\mathcal{F}_{OT}$, Sim generates uniformly random sharings of $b = 0$ and $b' = 1$ in $\mathcal{S}$ subject to the only condition that if $i \in A$, then $b_i = b'_i$. Note that this is possible since A is unqualified for $\mathcal{S}$. Then, in parallel Sim sends b_i to the environment for each $i \in A$. Sim now awaits that for each $i \in \overline{A}$, the environment sends $a_{(i,0)}$ and $a_{(i,1)}$ and that for each $i \in A$ the environment sends $a_{(i,b_i)}$.

For $k = 0, 1$, if m_k is not already set to 0 then Sim computes

$$m_k = \texttt{Reconstruct}^k_{\Sigma}(\{((i, b_i), a_{(i,b_i)}) : i \in \mathcal{P}_n\})$$

If the reconstruction of m_k fails, Sim sets $m_k = 0$. Finally, it sends $(\texttt{send}, m_0, m_1)$ to $\mathcal{F}_{OT}$.

By construction, the shares b_i corresponding to the set A of corrupt servers that the environment receives are indistinguishable from the A-shares in a uniformly random sharing of b, regardless of whether $b = 0$ or $b = 1$. Hence these b_i do not allow the receiver to distinguish the real and ideal world. Now, since after that step there is no further interaction, it suffices to show that the messages sent to Bob are indistinguishable from the ones sent in the real world.

This is the case since the shares have been chosen with the distribution Bob would use and since the simulator reconstructs the messages m_0 and m_1 in exactly the same way as Bob would reconstruct m_b in the real protocol, if b is his input. Therefore the real and ideal world are indistinguishable. □

We note that the simulators in the proof above run in polynomial time.

5.3 Proof of Theorem 1.1

We know from [Mas93] (see also [CCG+07, Theorem 1]) that a linear code (over a field $\mathbb{F}_q$) with length $n+1$, minimum distance d and whose dual code has minimum distance $d^{\perp}$ yields a linear secret sharing scheme for n players, with secret and shares in the same field $\mathbb{F}_q$ and such that any set of $d^{\perp} - 2$ players is unqualified and any set of $n - d + 2$ players is qualified. One can use then Gilbert-Varshamov (see below) bound to show that

Theorem 5.4. *For large enough n, there exists an ideal binary LSSS such that any set of $0.11n$ players is unqualified and any set of $0.89n$ players is qualified.*

Plugging this into Theorem 1.2 (in the case we have already proved) shows Theorem 1.1.

We now give a detailed argument on how to derive Theorem 5.4 from the Gilbert-Varshamov bound. This essentially follows the steps from [CCG+07].

Theorem 5.5 (Gilbert-Varshamov). *For every $0 \leq \delta < 1/2$ and any $0 < \epsilon < 1 - h_2(\delta)$ (where $h(\cdot)$ denotes the binary entropy function), if a linear code is chosen uniformly at random among all linear codes over $\mathbb{F}_2$ of length $n+1$ and dimension $k = \lceil (1 - h_2(\delta) - \epsilon)(n+1) \rceil$, then with probability $1 - 2^{-\Omega(n)}$ the code has minimum distance at least $\delta(n+1)$.*

Choosing $\delta = 0.11$ (which guarantees $h_2(\delta) < 1/2$), and $\epsilon = 1/2 - h_2(\delta)$ the theorem states that for large n, a uniformly random binary linear code of dimension $(n+1)/2$ has minimum distance $\delta(n+1)$ with very large probability. Now the dual of a code of dimension $(n+1)/2$ also has dimension $(n+1)/2$. So one can use a union bound argument and the observations above to conclude that choosing a random code of of dimension $(n+1)/2$ yields an ideal binary linear secret sharing scheme for n players such that any set of $\delta(n+1) - 2$ players is unqualified and any set of $(1-\delta)(n+1)+2$ players is qualified. Now the probabilistic method guarantees that for all large enough n there exists a binary ideal LSSS with $0.11n$-privacy and $0.89n$ reconstruction.

We should point out that Theorem 1.1 is merely an existence result, since explicit constructions of codes attaining the Gilbert-Varshamov bound are not known. Following [IMSW14], since randomly selected codes attain the bound with large probability, this result can be turned into an explicit construction of an *statistically* secure OT combiner, where Alice and Bob first agree on a random binary linear code by means of a coin tossing protocol; in this case we need a direct communication channel between Alice and Bob. Explicit constructions of perfectly secure OT-combiners against an active $(\Omega(n), \Omega(n))$-adversary can be obtained from algebraic geometric codes, but the underlying constant is worse than 0.11.

Note that one can also give non-asymptotic statements, at the cost of a small loss in the constant 0.11. Indeed [CCG+07, Corollary 2] (see also Definition 5 in the same paper) guarantees that for $n \geq 21$, there exists a binary linear code with both $d, d^{\perp} \geq \lfloor 0.1n \rfloor$.

Finally, for small values of n one can also obtain *explicit* constructions of ideal binary LSSS with relatively good privacy and reconstruction thresholds. One possibility is to use self-dual codes (i.e. codes that are their own duals), since in that case the minimum distance of the code and its dual is the same. Tables of such codes are available at [Gab]. These tables show for instance the existence of a binary self-dual code of length 8 and minimum distance 4, which would yield a single-use 7-server OT-combiner with perfect security against an active $(2,2)$-adversary.

6 Construction of OT-combiners in the general case

We show the general version of the protocol π_{OT} from the previous Section 5, when the adversary structure $\mathcal{A}$ is not necessarily the adversary structure of an ideal LSSS over $\mathbb{F}_2$. Note that many interesting access structures, for example most threshold structures, do not admit an ideal LSSS over $\mathbb{F}_2$. We show:

Theorem 1.2. *Let $\mathcal{A}$, $\mathcal{B} \subseteq 2^{\mathcal{P}}_n$ be adversary structures such that $(\mathcal{A}, \mathcal{B})$ is a R2 pair. Suppose there exists a linear secret sharing scheme $\mathcal{S}$ for n players where the secret is in $\{0,1\}$ and the i-th share is in $\{0,1\}^{\ell_i}$, and such that every set $A \in \mathcal{A}$ is unqualified in $\mathcal{S}$ and the complement $\overline{B}$ of every set $B \in \mathcal{B}$ is qualified in $\mathcal{S}$.*

Then there exists an OT combiner which calls the i-th server ℓ_i times and is perfectly secure against any active $(\mathcal{A}, \mathcal{B})$-adversary.

Let $\mathcal{S}$ be a possibly non-ideal perfect secret sharing scheme with adversary structure $\mathcal{A}$. For $i = 1, \ldots, n$ the i-th share of $\mathcal{S}$ belongs to some vector space $U_i = \{0,1\}^{\ell_i}$ for some integer $\ell_i \geq 1$. Let $\ell = \sum_{i=1}^{n} \ell_i$ be the complexity of $\mathcal{S}$.

The idea of the generalization is simple. The i-th server is split in ℓ_i subservers, each of which will receive one different bit of the i-th share of Bob's input. These subservers will now

Oblivious transfer protocol π_{OT} (non-ideal $\mathcal{S}$ case)

We use the index $i \in \{1, \ldots, n\}$ for the servers, $k_i \in \{1, \ldots, \ell_i\}$ to index the bits of the i-th share of $\mathcal{S}$ and $j \in \{0, 1\}$ to index the bits in Alice's input to each instance of OT.

1. Local computation:
   Bob creates a sharing $[b]_{\mathcal{S}} = (b_i)_{i \in \{1,\ldots,n\}}$, where each $b_i \in \{0,1\}^{\ell_i}$ is parsed as $(b_{i,1}, b_{i,2}, \ldots, b_{i,\ell_i})$ with $b_{i,k} \in \{0,1\}$.
   Alice creates a sharing

   $$[(m_0, m_1)]_\Sigma = (a_{(i,k,j)})_{i \in \{1,\ldots,n\}, k \in \{1,\ldots,\ell_i\}, j \in \{0,1\}}.$$

2. Use of the OT servers:
   For $i \in \{1, \ldots, n\}$ and for each $k \in \{1, \ldots, \ell_i\}$, Alice and Bob use server S_i to execute an OT with inputs $(a_{i,k,0}, a_{i,k,1})$ for Alice and $b_{i,k}$ for Bob. Let $y_{i,k}$ denote the output of Bob in instance (i, k).

3. Local computation:
   If $b = 0$, Bob constructs m'_0 by applying

   $$\texttt{Reconstruct}^0_\Sigma(\{((i,k,b_{i,k}), y_{i,k}) : i \in \mathcal{P}_n, k \in \{1, \ldots, \ell_i\}\}).$$

   Similarly, if $b = 1$, Bob constructs m'_1 by applying

   $$\texttt{Reconstruct}^1_\Sigma(\{((i,k,b_{i,k}), y_{i,k}) : i \in \mathcal{P}_n, k \in \{1, \ldots, \ell_i\}\}).$$

   In any of the cases, if the reconstruction fails, output $\mathbf{0}$. Otherwise output the reconstructed m'_b.

Figure 4: Protocol π_{OT}

work as the servers did in the protocol from Section 5 (we remark however that the adversaries corrupt full servers and not individual subservers). For that we need to modify the secret sharing scheme Σ used by Alice accordingly. More precisely, let $V, W \subseteq U_1 \times \cdots \times U_n$ be the sets of all possible sharings of 0 and 1 respectively. We can think of the elements of V and W as ℓ-bit strings, and we index their coordinates by pairs (i, k) where the (i, k)-th coordinate of a sharing is the k-th bit of the i-th share. Now we can define Σ as in Proposition 4.1 for these V and W (and setting $\mathbf{t}$ to be some sharing $[1]_{\mathcal{S}}$). Everything works therefore the same as in Section 5.1 except that Σ will now have 2ℓ shares. The set of shares will be indexed by $\mathcal{P}_{\ell,2} := \{(i,k,j) : i = 1, \ldots, n,\ k = 1, \ldots, \ell_i,\ j = 0, 1\}$. The general protocol is given in Figure 4. The security proofs work essentially as in the case presented in Section 5.

7 Necessary conditions for the existence of OT combiners

In this section we show Theorem 1.3,

Theorem 1.3. *Let $\mathcal{A}$, $\mathcal{B}$ be adversary structures on the set of servers $\{S_1, \ldots, S_n\}$. If there exists a perfectly secure OT-combiner which is secure against any passive $(\mathcal{A}, \mathcal{B})$-adversary and uses server S_i exactly ℓ_i times, then $(\mathcal{A}, \mathcal{B})$ is an R2 pair of structures and there exists a secret sharing scheme for n players with secret in $\{0, 1\}$, the i-th share in $\{0, 1\}^{\ell_i}$, for $i = 1, \ldots, n$ and such that every set $A \in \mathcal{A}$ is unqualified in $\mathcal{S}$ and the complement $\overline{B}$ of any set every set $B \in \mathcal{B}$ is qualified in $\mathcal{S}$.*

First we show that if $(\mathcal{A}, \mathcal{B})$ were not $\mathcal{R}_2$ then the existence of an unconditionally secure OT combiner would imply the existence of a 2-party unconditionally secure OT protocol. Indeed if $(\mathcal{A}, \mathcal{B})$ is not $\mathcal{R}_2$, then there exists $A \in \mathcal{A}$ and $B \in \mathcal{B}$ such that $A \cup B$ is the set of all servers. Then the entire protocol can be emulated by two players: Alice', who plays the joint role of Alice and all the servers in A and Bob' who plays for Bob and all servers in B. This is then a two-party protocol in the plain model which is unconditionally secure against a semi-honest adversary who can corrupt either of the players Alice' and Bob'. This is known to be impossible.

Next, we prove the existence of a secret sharing scheme with the properties mentioned in the theorem. In fact, we simply reproduce the arguments from [IMSW13] in our setting. Assume we have an OT combiner which is perfectly secure against an $(\mathcal{A}, \mathcal{B})$-adversary and where the i-th server is used ℓ_i times. Then Bob's inputs to the OT servers must have been computed from his global input to the OT combiner by some probabilistic algorithm `AlgBob`. We now consider a secret sharing scheme $\mathcal{S}$ whose sharing algorithm is `AlgBob` (understanding that the i-th share is the bit-string containing all ℓ_i inputs bits to the i-th OT server produced by `AlgBob`). Since the OT combiner is secure against and adversary corrupting Alice and a set $A \in \mathcal{A}$, this means that every $A \in \mathcal{A}$ must be unqualified in $\mathcal{S}$. Next we show that for every $B \in \mathcal{B}$, its complement $\overline{B}$ must be a reconstructing set for $\mathcal{S}$. Consider a player Alice' who plays the role of Alice and the servers in $\overline{B}$ in the OT-combiner and a player Bob', who plays the role of Bob and the servers in B. Assume that the inputs of Alice and Bob are independent. We then have a protocol between Alice' and Bob' in the plain model, which correctly implements the OT functionality and in which, by security of the OT combiner and since $B \in \mathcal{B}$, Bob' obtains no information about the input (m_0, m_1) of Alice' after the protocol has been executed. In these conditions, it follows from standard arguments about the impossibility of two party computation in the plain model (see e.g. [CDN15]) that Alice' not only obtains information about Bob's input, but in fact she recovers it with probability 1. Given that all the information that Alice' has learned during the execution of the protocol is the input bits to the servers in $\overline{B}$, we conclude that $\overline{B}$ is a reconstructing set for $\mathcal{S}$.

8 2-out-of-3 OT-combiners

As an application of Theorems 1.2 and 1.3 we determine the minimal number of calls for a perfectly secure OT combiner where we have 3 servers, and 2 of them are secure. In other words, we want perfect security against an $(1,1)$-adversary, i.e. $\mathcal{A} = \mathcal{B} = \{\{1\}, \{2\}, \{3\}\}$. By Theorem 1.2, we are then interested in finding a linear secret sharing scheme over $\mathbb{F}_2$ for 3 players such that it has 1-privacy (every single player is unqualified) and it has 2-reconstruction (every set of two players is qualified). Note that we want to find a threshold secret sharing scheme, but Shamir's scheme cannot be used directly over $\mathbb{F}_2$ (we would tolerate at most 2 players). One could instead use Shamir's scheme over the extension field $\mathbb{F}_4$, and in this case we have shares which are each in $\{0,1\}^2$. This yields an OT-combiner where each server is called twice, which matches the number of calls in a construction in [HKN$^+$05]. However, we show that one can do better with the following LSSS $\mathcal{S}$.

Lemma 8.1. *$\mathcal{S}$ has 2-reconstruction and 1-privacy.*

Corollary 8.2. *There exists an OT combiner for 3 OT servers which is perfectly secure against an $(1,1)$-adversary and makes 1 call to one of the OT servers and 2 calls to each of the other 2 servers.*

Now we apply Theorem 1.3 in combination with the results from [CCX13] to show that this is optimal in the total number of server calls. Theorem 1.3 states that given an OT-combiner in the conditions above, there needs to exist a secret sharing scheme (linear or not) for 3

Secret sharing scheme $\mathcal{S}$

To share $s \in \{0,1\}$.

- Sample r and r' uniformly at random in $\{0,1\}$.
- Send:
  1. r to Player 1.
  2. $(s-r, r')$ to Player 2.
  3. $(s-r, s-r')$ to Player 3.

Figure 5: A 2-out-of-3 threshold linear secret sharing scheme $\mathcal{S}$

players with 1-privacy, 2-reconstruction and share lengths matching the number of calls to the OT-servers. On the other hand we have

Theorem 8.3 ([CCX13]). *Suppose there exists a secret sharing scheme for n players, where the i-th share takes values in an alphabet A_i, and such that it has t-privacy and r-reconstruction. Let $\overline{q} = \frac{1}{n}\sum_{i=1}^{n}|A_i|$ be the average cardinality of the share-alphabets. Then*

$$r - t \geq \frac{n-t+1}{\overline{q}}.$$

Therefore, a secret sharing in the conditions above must satisfy that the average cardinality of the share-alphabets is $\overline{q} \geq 3$. Now note that in our case the shares are in $\{0,1\}^{\ell_i}$, which are alphabets of cardinality 2^{ℓ_i}, and we can rule out degenerate cases where $\ell_i = 0$ (since in that case, clearly it cannot happen simultaneously that $\{i,j\}$ is qualified and $\{j\}$ is unqualified). Under all these conditions, one can easily check that $\sum_{i=1}^{3}\ell_i < 5$ and $\overline{q} = \frac{1}{3}\sum_{i=1}^{3}2^{\ell_i} \geq 3$ cannot be achieved simultaneously. Therefore,

Corollary 8.4. *The minimal number of calls for a OT combiner for 3 OT servers which is perfectly secure against an $(1,1)$-adversary is 5.*

□

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 119–135, 2001.

[BI01] Amos Beimel and Yuval Ishai. On the power of nonlinear secret-sharing. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001*, pages 188–202, 2001.

[Bla79] George Blakley. Safeguarding cryptographic keys. In *Proceedings of the 1979 AFIPS National Computer Conference*, volume 48, pages 313–317, June 1979.

[BM89] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and sppli-cations. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 547–557, 1989.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145. IEEE, 2001.

[CCG+07] Hao Chen, Ronald Cramer, Shafi Goldwasser, Robbert de Haan, and Vinod Vaikuntanathan. Secure computation from random error correcting codes. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, pages 291–310, 2007.

[CCM98] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 493–502, 1998.

[CCX13] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. Bounds on the Threshold Gap in Secret Sharing and its Applications. *IEEE Trans. Information Theory*, 59(9):5600–5612, 2013.

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing.* Cambridge University Press, 2015.

[CK88] Claude Crépeau and Joe Kilian. Achieving Oblivious Transfer Using Weakened Security Assumptions (Extended Abstract). In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 42–52, 1988.

[DvdGMN08] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade, and Anderson C. A. Nascimento. Oblivious transfer based on the mceliece assumptions. In *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, pages 107–117, 2008.

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A Randomized Protocol for Signing Contracts. In *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 205–210, 1982.

[Gab] Philippe Gaborit. Tables of self-dual codes. Available at `http://www.unilim.fr/pages_perso/philippe.gaborit/SD/`.

[GIS+10] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, pages 308–326, 2010.

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In *Theory of Cryptography*, pages 393–411. Springer, 2008.

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In *Advances in Cryptology–EUROCRYPT 2005*, pages 96–113. Springer, 2005.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and Jürg Wullschleger. Constant-Rate Oblivious Transfer from Noisy Channels. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st*

Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings, volume 6841 of *Lecture Notes in Computer Science*, pages 667–684. Springer, 2011.

[IMSW13] Yuval Ishai, Hemanta K. Maji, Amit Sahai, and Jürg Wullschleger. Single-Use Oblivious Transfer Combiners. Full version of [IMSW14]. Available at `https://www.cs.purdue.edu/homes/hmaji/papers/IshaiMaSaWu13.pdf`, 2013.

[IMSW14] Yuval Ishai, Hemanta K. Maji, Amit Sahai, and Jürg Wullschleger. Single-Use OT Combiners with Near-Optimal Resilience. In *2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014*, pages 1544–1548, 2014.

[IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer - efficiently. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 572–591. Springer, 2008.

[ISN87] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE GlobeCom '87 Tokyo*, pages 99–102, 1987.

[JMO93] Wen-Ai Jackson, Keith M. Martin, and Christine M. O'Keefe. Multisecret threshold schemes. In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 126–135, 1993.

[Kil88] Joe Kilian. Founding Cryptography on Oblivious Transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 20–31, 1988.

[Mas93] James L. Massey. Minimal codewords and secret sharing. In *Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory*, pages 276–279, 1993.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, pages 554–571, 2008.

[PW08] Bartosz Przydatek and Jürg Wullschleger. Error-tolerant combiners for oblivious primitives. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pages 461–472, 2008.

[Rab81] Michael Rabin. How to Exchange Secrets with Oblivious Transfer. Technical report, Aiken Computation Lab, Harvard University, 1981.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[VV15] Vinod Vaikuntanathan and Prashant Nalini Vasudevan. Secret sharing and statistical zero knowledge. In *Advances in Cryptology - ASIACRYPT 2015 - 21st*

International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I, pages 656–680, 2015.