One-out-of-q OT Combiners

Oriol Farràs and Jordi Ribes-González

Universitat Rovira i Virgili, Tarragona, Spain
{oriol.farras,jordi.ribes}@urv.cat

Abstract. In 1-*out-of-q Oblivious Transfer (OT)* protocols, a sender Alice is able to send one of $q \geq 2$ messages to a receiver Bob, all while being oblivious to which message was transferred. Moreover, the receiver learns only one of these messages. *Oblivious Transfer combiners* take n instances of OT protocols as input, and produce an OT protocol that is secure if sufficiently many of the n original OT instances are secure.
We present new 1-out-of-q OT combiners that are perfectly secure against active adversaries. Our combiners arise from secret sharing techniques. We show that given an $\mathbb{F}_q$-linear secret sharing scheme on a set of n participants and adversary structure $\mathcal{A}$, we can construct n-server, 1-out-of-q OT combiners that are secure against an adversary corrupting either Alice and a set of servers in $\mathcal{A}$, or Bob and a set of servers B with $\bar{B} \notin \mathcal{A}$. If the normalized share size of the scheme is ℓ, then the resulting OT combiner requires ℓ calls to *OT* protocols, and the total amount of bits exchanged during the protocol is $(q^2 + q + 1)\ell \log q$.
As a consequence of this result, for any prime power $q \geq n$ we present n-server, 1-out-of-q OT combiners that are perfectly secure against active adversaries that corrupt either Alice or Bob, and also a minority of the OT candidates. The total amount of exchanged bits is $(q^2+q+1)n \log q$.

Keywords: Oblivious transfer · OT combiners · Secret sharing schemes

1 Introduction

Oblivious Transfer (OT) protocols involve two parties, a *sender* and a *receiver*, which we also respectively name Alice and Bob. The functionality provided by OT consists in allowing the sender to transfer part of its inputs to the receiver, while guaranteeing that the sender is oblivious to which part of its inputs is actually obtained by the receiver. It also guarantees that the receiver is not able learn more information than it is entitled to as per the protocol.

OT protocols were first introduced by Rabin [44] in 1981. In 1-*out-of*-2 *OT* protocols [22], the sender holds two messages, and the receiver chooses to receive one of them from the sender. Security here consists in the sender being oblivious to the message that was actually transferred, and the receiver getting information on only one of the messages. The type of OT that we study here is called 1-*out-of-q OT*. It is a generalization of 1-out-of-2 OT, first presented by Crépeau, Brassard and Robert [18] in 1986. It lets the sender hold $q \geq 2$ messages instead of just two, and allows the receiver to fetch only one of those messages. The relevance of OT protocols in cryptography lies in their role as a fundamental primitive in many cryptographic constructions. The main functionalities OT has found an application to are secure multi-party computation, zero-knowledge proofs and bit commitment schemes (see [10,33,34,47], for example).

The security of OT protocols is necessarily conditional, since perfectly secure OT protocols would yield unconditionally-secure two-party computation by [33], which is impossible to obtain for some functions (see [11,17]). Hence, the security of OT protocols is based on computational hardness assumptions such as the hardness of RSA [44], the DDH assumption [1,10], code-based assumptions [21] and also lattice-based assumptions [42]. Alternatively, the security of OT can be guaranteed by the existence of a noisy channel between both parties [19], the use of hardware tokens [26], restrictions on the storage [13], and other ways. The conditional security of OT protocols implies that the security guarantees of OT could be compromised. The standard method to mitigate this concern consists in grounding security on various assumptions at once, by simultaneously using several implementations. This motivates the use of OT combiners.

The notion of *combiner* consists of blending various cryptographic implementations into one, so that the resulting combination is secure even if some of the original implementations are insecure. Combiners have been previously studied in other areas of cryptography, for instance in multi-factor authentication, where many authentication methods are used concurrently, as well as in cascading of block ciphers or hybrid key encapsulation.

Using an *OT combiner*, a set of n candidate implementations of OT can be merged to realize a single OT protocol. In other words, an OT combiner can be used to instantiate a protocol between a sender Alice and a receiver Bob that realizes OT by internally using n candidate OT implementations. The resulting protocol is secure as long as sufficiently many of the initial implementations were secure to begin with.

An OT combiner is *black-box* if, during the combined protocol, the candidate OT implementations are used in a black-box way, i.e. ignoring their internal workings. In this work, we only consider black-box OT combiners. Under this assumption, as in [16], we view OT combiners as *server-aided* OT protocols. This means that we model each of the OT candidate implementations as a *server* that implements the 1-out-of-q OT functionality, i.e. that receives q messages $m_0, \ldots, m_{q-1}$ from Alice and an index b from Bob, and outputs the message m_b to Bob. We then say that an OT combiner is n-*server* if it takes n OT candidates as input. An OT combiner is *single-use* if each OT candidate is used only once during the execution of the protocol. In contrast, an OT combiner is *multi-use* if, during the protocol, an OT candidate can be used more than once.

1.1 Related Work

The study of OT combiners was initiated by Harnik, Kilian, Naor, Reingold and Rosen [28] in 2005. They define the notion of (n, t)-*OT combiner*, which consists in taking n candidate 1-out-of-2 OT implementations and combining them into a 1-out-of-2 OT protocol that is secure provided at most t of the OT candidates are faulty. They show that, when $t < n/2$, there exist (n,t)-OT combiners that are unconditionally secure against passive (i.e. semi-honest) adversaries. They prove the tightness of this bound and show that such OT combiners cannot exist for $n = 2, t = 1$.

Next, we present the main results on OT combiners that, like [28], take 1-out-of-2 OT implementations and combine them into a 1-out-of-2 OT protocol. There are other combiners, not covered here, in which the OT functionality is obtained by combining different primitives like oblivious linear function evaluation [43] or private information retrieval [38] protocols.

Meier, Przydatek and Wullschleger [39] present OT combiners that implement the 1-out-of-2 OT functionality and are unconditionally secure against passive adversaries that corrupt either Alice and a number t_A of OT candidates, or Bob and t_B OT candidates for any $t_A + t_B < n$. These protocols were later called (n, t_A, t_B)-*OT combiners*. Their combiner is multi-use, and it makes two calls to each OT candidate. We call $\mathcal{P}_n = \{1, \dots, n\}$ the set of OT candidates.

Harnik, Ishai, Kushilevitz and Nielsen [27] present the first single-use OT combiner. A statistically secure (n,t,t)-OT combiner is given for $t = \Omega(n)$, which makes a constant number of calls to each OT candidate. Their solution is set in the 1-out-of-2 scenario. Additionally, [27] gives a computationally secure OT combiner against active adversaries. Subsequently, Ishai, Prabhakaran and Sahai [30] show that this construction can be turned into an (n,t,t)-OT combiner that is statistically secure against active adversaries for $t = \Omega(n)$. Ishai, Maji, Sahai and Wullschleger [29] present a single-use (n,t,t)-OT combiner that is statistically secure against passive adversaries for $t = n/2 - \omega(\log \kappa)$, where κ is the security parameter.

Following [29], Cascudo, Damgård, Farràs and Ranellucci [16] achieve single-use 1-out-of-2 OT combiners. They generalize the security notion in [27] by defining the notion of *perfect security against active* $(\mathcal{A}, \mathcal{B})$-*adversaries* for some $\mathcal{A}, \mathcal{B} \subseteq 2^{\mathcal{P}_n}$, which we also adopt in this article. This definition considers an active adversary that can corrupt either Alice and a set $A \in \mathcal{A}$ of OT candidates, or Bob and a set $B \in \mathcal{B}$ of OT candidates, obtaining their inputs and full control of their outputs. Their OT combiner achieves perfect (unconditional, zero-error) security against $\mathcal{R}_2$ adversaries, i.e., adversaries with pairs $(\mathcal{A}, \mathcal{B})$ satisfying that $A \cup B \neq \mathcal{P}_n$ for every $A \in \mathcal{A}$ and $B \in \mathcal{B}$.

All the works mentioned above combine various primitives into 1-out-of-2 OT. It is also possible to use of 1-out-of-2 OT to obtain 1-out-of-q OT for any integer $q \geq 2$, by taking as many as $q - 1$ [18] or $\log q$ [40] OT candidates. However, in these constructions, a single faulty 1-out-of-2 OT candidate results in an insecure 1-out-of-q OT protocol. This motivates the search for 1-out-of-q OT combiners with better security guarantees.

Our OT combiners are built with the use a specific kind of secret sharing schemes that has been studied in previous works. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, consider the access structure on the set of participants $P = \{1, \dots, n\} \times \{0,1\}$ whose minimal subsets are $\{(1, x_1), \dots, (n, x_n)\}$ with $f(x_1, \dots, x_n) = 1$ and the subsets $\{(i,0),(i,1)\}$ for $1 \leq i \leq n$. Efficient constructions are known [7,46] for some of these structures. Liu, Vaikuntanathan and Wee [37] presented more efficient schemes, and showed a connection between these schemes and Conditional Disclosure of Secrets (CDS) protocols [25], namely for CDS protocols for the INDEX predicate. That connection was later used to construct better general constructions for secret sharing [2,3,36,8]. In this work, we study access structures determined by functions $f : \{0, \dots, q-1\}^n \to \{0, 1\}$, a case studied in [2,3,25], for example.

1.2 Our Work

This work is dedicated to the construction of efficient 1-out-of-q OT combiners for any $q \geq 2$. We extend the security and consistency notions in [16] from the 1-out-of-2 case to the 1-out-of-q case, and we present OT combiners that attain perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries. As far as we know, this is the first work dedicated to the construction of efficient 1-out-of-q OT combiners with active security for any $q \geq 2$. Our main result is the following theorem, which is constructive.

Theorem 1.1. *Let $\mathbb{F}_q$ be a finite field, let Σ be an $\mathbb{F}_q$-linear secret sharing scheme on $\mathcal{P}_n$ with adversary structure $\mathcal{A}$ and normalized total share size ℓ, and let $\mathcal{B} \subseteq 2^{\mathcal{P}_n}$ satisfying that $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$. Then there exists an n-server, 1-out-of-q OT combiner for messages in $\mathbb{F}_q$ that is perfectly secure against any active $(\mathcal{A}, \mathcal{B})$-adversary and requires exchanging $(q^2 + q + 1)\ell \log q$ bits. If Σ is ideal, then the OT combiner is single-use.*

The communication cost of this theorem does not take into account the communication cost of the OT instantiations because our combiner is black-box. Our 1-out-of-q OT combiner can also be used as 1-out-of-m OT combiner for any integer $m < q$, reducing a bit the communication cost. As a corollary of this theorem and [16], we have that $(\mathcal{A}, \mathcal{B})$ admits a 1-out-of-q OT combiner if and only if $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$. It was already known for $q = 2$ [16], but not for greater q.

This OT combiner makes use of the OT implementations a total of ℓ times. That is, the number of OT calls and the communication complexity increase linearly with the normalized total share size of Σ. Therefore, in our construction, the search of efficient OT combiners can be reduced to the search of efficient linear secret sharing schemes. We can guarantee efficient 1-out-of-q OT combiners for those pairs of $\mathcal{R}_2$ adversary structures $(\mathcal{A}, \mathcal{B})$ for which there exist $\mathbb{F}_q$-linear secret sharing schemes with adversary structure $\mathcal{C}$ satisfying $\mathcal{A} \subseteq \mathcal{C}$ and $\mathcal{B} \subseteq \mathcal{C}^* = \{C \subseteq \mathcal{P}_n : \mathcal{P}_n \setminus C \notin \mathcal{C}\}$. In the threshold adversary structure case, we have the following result.

Theorem 1.2. *Let q be a prime power and let $2 \leq n \leq q$. There exists a single-use, n-server, 1-out-of-q OT combiner that is perfectly secure against active adversaries corrupting either Alice or Bob, and a minority of the OT candidates. The amount of bits exchanged during the protocol is $(q^2 + q + 1)n \log q$.*

In the process of building our 1-out-of-q OT combiners, we study secret sharing schemes associated to affine spaces. Namely, let $W \subseteq \mathbb{F}_q^n$ be an affine space, and let $f : \mathbb{F}_q^n \to \{0,1\}$ be the Boolean function with $f(x_1, \ldots, x_n) = 1$ if and only if $(x_1, \ldots, x_n) \in W$. We present ideal linear secret sharing schemes on the set of nq participants $\{1, \ldots, n\} \times \mathbb{F}_q$ in which a subset $\{(1, v_1), \ldots, (n, v_n)\}$ is authorized if and only if $f(v_1, \ldots, v_n) = 1$. Moreover, from our schemes, it is possible to build n-server CDS protocols for f with domain of secrets $\mathbb{F}_q$, and with optimal message size and certain *robustness*, in the sense of [2].

We found that it is also possible to build 1-out-of-q OT combiners with similar security properties adapting the 1-out-of-q OT construction Crépeau, Brassard and Robert [18] to this setting, combining 1-out-of-2 OT combiners. This construction relies on $\mathbb{F}_2$-linear secret sharing schemes, and 1-out-of-2 OT protocols. For threshold adversary structures, the communication complexity is $(3q-1)n \log n$ for bit-messages and is multi-use, each similar to the one of the construction mentioned above, but it is not single-use, each one of the 1-out-of-2 OT instances is executed $q \log q \log n$ times. For larger messages, the protocol is replicated and combined with zigzag functions [18].

In general, given an $\mathcal{R}_2$ pair $(\mathcal{A}, \mathcal{B})$, the convenience of this construction or the one in Theorem 1.1 will depend on the share size of $\mathbb{F}$-linear secret sharing schemes for $(\mathcal{A}, \mathcal{B})$ for and $\mathbb{F} = \mathbb{F}_2$ or $\mathbb{F} = \mathbb{F}_q$. Notice that the power of linear secret sharing schemes over fields of different characteristics is incomparable. Indeed, there is a super-polynomial separation between any two fields with different characteristics [9].

1.3 Overview of our Constructions

Next we describe the structure of the main constructions of this work (Sections 4 and 7). We have n servers $S_1, \ldots, S_n$, each one performing the 1-out-of-q OT functionality. Bob holds $b \in \{0 \ldots q-1\}$, and Alice holds some messages $m_0, \ldots, m_{q-1}$. At the end of the protocol, Bob can recover m_b, while Alice does not get information about b.

Bob creates n shares of b with a secret sharing scheme Σ, and sends one share to each server. That is, the server S_i receives a share b_i, which will be the selection input of the OT functionality of S_i. Independently, Alice creates shares of her messages. Alice creates nq shares of the message m_0 with some special secret sharing scheme $\mathcal{S}_0$. These shares $m_0^{(i,j)}$ and are indexed by $(i,j) \in \{1 \ldots n\} \times \{0 \ldots q-1\}$. Then Alice creates shares of the rest of messages m_k, each with a different scheme $\mathcal{S}_k$, obtaining the shares $m_k^{(i,j)}$ with $(i,j) \in \{1 \ldots n\} \times \{0 \ldots q-1\}$. All these shares are packed in strings $u_i^j = m_0^{(i,j)} || m_1^{(i,j)} || \cdots || m_{q-1}^{(i,j)}$, and Alice sends $u_i^0, \ldots, u_i^{q-1}$ to server S_i for each $1 \leq i \leq n$. Then, server S_i performs the 1-out-of-q OT functionality with the inputs received from Alice and Bob. Namely, S_i outputs $u_i^{b_i}$ to Bob. Finally, Bob collects all messages sent by the servers, and recovers m_b. A diagram of the protocol for the case $q = 4$ and $n = 3$ is presented in Figure 1.

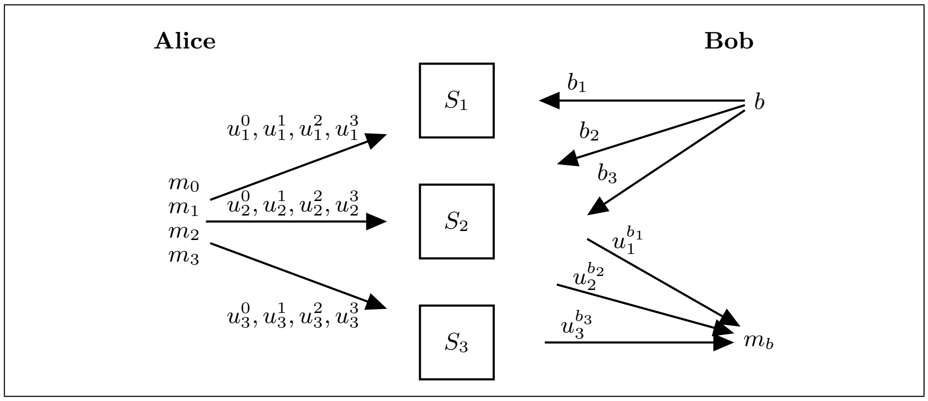

Fig. 1. Diagram of a 1-out-of-4 OT combiner for $n = 3$.

In this setting, Σ and $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$ must be chosen in such a way that the functionality is performed correctly. Roughly speaking, if $b_1, \ldots, b_n$ is a valid sharing of b by Σ, then it must be possible to recover m_b from $u_1^{b_1}, \ldots, u_n^{b_n}$. Hence, the subsets $\{(1, b_1), \ldots, (n, b_n)\}$ must be authorized sets of $\mathcal{S}_b$ if $b_1, \ldots, b_n$ is a valid share of b. Regarding security, if Bob can recover m_b, he must not be able to obtain any information about other m_k for $k \neq b$. Hence, in this case, $u_1^{b_1}, \ldots, u_n^{b_n}$ must not provide any information about m_k, and so $\{(1, b_1), \ldots, (n, b_n)\}$ must be a forbidden subset of $\mathcal{S}_k$.

Additionally, our OT combiners guarantee protection against adversaries taking control of Alice and some servers $A \in \mathcal{A}$, or Bob and some servers $B \in \mathcal{B}$. This protection is guaranteed by restricting the access structures of the secret sharing schemes Σ and $\mathcal{S}_k$ to these requirements, while preserving correctness (see an introduction to secret sharing in Section 2.1). In general, given a secret sharing scheme Σ, it is not known if there exist efficient schemes $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$ adapted to the shares of Σ.

One of the main contributions of this work is that, when Σ is ideal and $\mathbb{F}_q$-linear, we find schemes $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$ that are ideal, $\mathbb{F}_q$-linear, and satisfy the desired security restrictions, and thus we obtain an efficient, single-use, n-server, 1-out-of-q OT combiner. Hence, given an $\mathcal{R}_2$ pair of adversary structures $(\mathcal{A}, \mathcal{B})$, if $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-linear secret sharing scheme, then the resulting OT combiner is perfectly secure against active $(\mathcal{A}, \mathcal{B})$-adversaries. Here, the main difficulty lies in constructing the schemes $\mathcal{S}_k$. As mentioned above, subsets $\{(1, b_1), \ldots, (n, b_n)\}$ for sharings of b must be authorized for $b = k$ and forbidden if $b \neq k$. Additionally, since we guarantee perfect security against an active adversary controlling Alice and servers $\{S_i\}_{i \in A}$ for any $A \in \mathcal{A}$, it must not be possible to obtain b from $\{b_i\}_{i \in A}$. And, since we guarantee perfect security against an active adversary controlling Bob and servers $\{S_i\}_{i \in B}$ for any $B \in \mathcal{B}$, the subset $\{(1, b_1), \ldots, (n, b_n)\} \cup \{(i, j) : i \in B, 0 \leq j < q\}$ must be forbidden in $\mathcal{S}_k$ for every $k \neq b$.

In the construction of Theorem 1.1, we use the fact that if a secret sharing scheme Σ is $\mathbb{F}_q$-linear, then the family of sharings of $b \in \mathbb{F}_q$ form an affine space in $\mathbb{F}_q^n$. Our schemes $\mathcal{S}_k$, as the ones in [16] for $q = 2$, exploit this property. However, the case $q = 2$ differs from the case $q > 2$ in the following sense. For every $\mathbb{F}_2$-affine space W, there exists an ideal $\mathbb{F}_2$-linear secret sharing scheme whose minimal authorized subsets are vectors in W. However, for $q > 2$, $\mathbb{F}_q$-affine spaces do not admit ideal $\mathbb{F}_q$-linear secret sharing schemes, in general.

We circumvent this problem by finding sufficient conditions for a secret sharing scheme to be used as a building block of our OT combiner. We call these schemes W-OT-compatible. Then we find that the schemes $\mathcal{S}_k$ that are the natural extension of the ones in [16] are indeed W-OT-compatible.

In case that Σ is a non-ideal $\mathbb{F}_q$-linear secret sharing scheme with adversary structure $\mathcal{A}$, we can still construct OT combiners that are perfectly secure against active $(\mathcal{A}, \mathcal{B})$-adversaries, but they require some servers to perform more than one OT execution. The number of OT executions coincides with the normalized total share size of Σ. In this case, we transform Σ into an ideal $\mathbb{F}_q$-linear scheme Σ' defined on an extended set of participants, and we run the protocol for the ideal case with Σ'.

1.4 Paper Organization

In Section 2, we lay out the preliminaries on secret sharing and OT combiners. Section 3 states the correctness and security definitions of OT combiners. In Section 4, we present our single-use 1-out-of-q OT combiner. The results of this section imply Theorem 1.2. Section 5 analyzes the secret sharing schemes used in our constructions, and Section 6 presents the correctness and security proofs of this combiner. In Section 7, we extend our construction to the general case where, obtaining a multi-use OT combiner. Results in Sections 4 and 7 imply Theorem 1.1. Finally, we present constructions of 1-out-of-q OT combiners built from 1-out-of-2 OT combiners in Section 8.

2 Preliminaries

In this section, we lay out the background theory needed in the rest of the article. In Sections 2.1 and 2.2 we give an account of secret sharing and we present OT combiners in Section 2.3.

From now on, q denotes an arbitrary positive prime power. By abuse of notation, we denote the finite field of q elements as $\mathbb{F}_q = \{0, \ldots, q-1\}$. The *power set* of a set P is $2^P := \{A : A \subseteq P\}$. Given an integer $n \geq 2$, we denote $\mathcal{P}_n := \{1, \ldots, n\}$ and $\mathcal{P}_{n,q} := \mathcal{P}_n \times \mathbb{F}_q = \{(i,j) : i \in \mathcal{P}_n, j \in \mathbb{F}_q\}$. We also consider the partition $\mathcal{P}_{n,q} = P_1 \cup \ldots \cup P_n$, where $P_i := \{(i,0), (i,1) \ldots, (i, q-1)\}$ for $i = 1, \ldots, n$. For any $A \subseteq \mathcal{P}_n$ we denote $\bar{A} = \mathcal{P}_n \setminus A$. For any $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ we define its *dual* as $\mathcal{A}^* = \{A \subseteq \mathcal{P}_n : \bar{A} \notin \mathcal{A}\}$.

2.1 Secret Sharing Schemes

For convenience, we take the definition of secret sharing scheme from [16], which is equivalent to the standard one [5]. For an introduction to this field, see [5,41].

Definition 2.1 ([16]). *A* secret sharing scheme Σ *on a set of participants* $P = \{1, \ldots, n\}$ *consists of the following two algorithms*

$(x_1, \ldots, x_n) \leftarrow \mathtt{Share}_\Sigma(s, \mathbf{r})$**:** *Probabilistic algorithm that takes as input a* secret *s, belonging to a finite set E_0, and some randomness $\mathbf{r}$ in a set Ω. It returns an array of values $(x_1, \ldots, x_n)$, where each x_i belongs to some finite set E_i. This array is called a* sharing *of s, and each of its elements is a* share *of s.*

$s \leftarrow \mathtt{Reconstruct}_\Sigma((i, x_i)_{i\in A})$**:** *Algorithm that takes a set of pairs $(i, x_i)_{i\in A}$ as input for some $A \subseteq P$, where $x_i \in E_i$. It returns either a secret s, or $\perp$.*

The normalized *total share size is* $\sum_{i=1}^n \log |E_i| / \log |E_0|$.

Following the notation of [16], given a secret s and randomness $\mathbf{r}$, we denote a sharing of the secret s by $[s, \mathbf{r}]_\Sigma = \mathtt{Share}_\Sigma(s, \mathbf{r})$. Whenever we can safely drop the randomness $\mathbf{r}$, we denote this sharing by $[s]_\Sigma$. The indexes i of shares x_i in the input to $\mathtt{Reconstruct}_\Sigma$ are omitted when implicitly clear. With this notation, we continue with more definitions. Let $A \subseteq P$. We say that

- A is *authorized* for Σ if, for every secret s, provided the shares $(x_i)_{i\in A}$ are part of a sharing of s, the function $\mathtt{Reconstruct}((i, x_i)_{i\in A})$ recovers s with probability one. That is, if, for every secret s,
$$\mathsf{Pr}[\mathtt{Reconstruct}_\Sigma((\mathtt{Share}_\Sigma(s, \mathbf{r}))_A) = s] = 1.$$
- A is *forbidden* for Σ when the shares $(x_i)_{i\in A}$ of participants in A do not reveal any information on the secret value s. That is, if, for every $s, s' \in E_0$, and every $(x_i)_{i\in A} \in \prod_{i\in A} E_i$,
$$\mathsf{Pr}[(\mathtt{Share}_\Sigma(s, \mathbf{r}))_A = (x_i)_{i\in A}] = \mathsf{Pr}[(\mathtt{Share}_\Sigma(s', \mathbf{r}))_A = (x_i)_{i\in A}].$$

We define the *access structure* of a scheme Σ as the family of all its authorized subsets, and the *adversary structure* of Σ is the family of its forbidden subsets. We say that Σ is *perfect* if every subset $A \subseteq P$ is either authorized or forbidden. A perfect scheme with total normalized share size n is called *ideal*.

Due to [31], access (adversary) structures are just monotone increasing (decreasing) families of subsets. For an access structure Γ, we define the *minimal access structure of* Γ by $\min \Gamma = \{A \in \Gamma : B \not\subset A \text{ for all } B \in \Gamma\}$. Analogously, given an adversary structure $\mathcal{A}$, we define the *maximal adversary structure of* $\mathcal{A}$ by $\max \mathcal{A} = \{A \in \mathcal{A} : A \not\subset B \text{ for all } B \in \mathcal{A}\}$. Given two adversary structures $\mathcal{A}, \mathcal{B} \subseteq 2^P$, we say that the pair $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$ if $A \cup B \neq P$ for every $A \in \mathcal{A}$ and $B \in \mathcal{B}$. Notice that a pair $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$ if and only if $\mathcal{B} \subseteq \mathcal{A}^*$.

2.2 Linear Secret Sharing Schemes

Linear Secret Sharing schemes (LSSS) are a type of secret sharing schemes that is key to building our 1-out-of-q OT constructions. For convenience, we use the definition in [5,16], where secrets are elements of a finite field.

Definition 2.2. *Let Σ be a secret sharing scheme, where secrets take values in E_0, and shares in $E_1 \times \cdots \times E_n$. Let $\mathbb{F}$ be a finite field. Then Σ is* $\mathbb{F}$-linear *if the following conditions hold*

1. *$E_0 = \mathbb{F}$ and $\Omega, E_1, \ldots, E_n$ are vector spaces of finite dimension over $\mathbb{F}$,*
2. *the randomness $\mathbf{r}$ is chosen uniformly over Ω, and*
3. *$\mathtt{Share}_\Sigma$ is an $\mathbb{F}$-linear surjective map*

$$\mathtt{Share}_\Sigma : E_0 \times \Omega \rightarrow E_1 \times \cdots \times E_n.$$

For each $i \in P$, the i-th share space is $E_i = \mathbb{F}^{\ell_i}$ for some positive integer ℓ_i. Linear schemes are perfect and the normalized total share size of these schemes is $\ell = \sum_{i=1}^{n} \ell_i$.

Every adversary structure admits an $\mathbb{F}_q$-LSSS for every q [31]. However, almost all access structures require $\mathbb{F}_q$-LSSS with normalized share size at least $2^{n/3-o(n)}$ for every q [4]. The characterization of adversary structures admitting efficient LSSSs is an open problem.

Given a secret value $s \in \mathbb{F}_q$, we have that $[s]_\Sigma \in \mathbb{F}_q^\ell = \mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$. In this case, if we denote by V the set of all possible shares $[0]_\Sigma$ of $0 \in \mathbb{F}_q$, then $V = \{\mathtt{Share}_\Sigma(0, \mathbf{r}) \ : \ \mathbf{r} \in \Omega\}$ is a vector subspace of $\mathbb{F}_q^\ell$. Similarly, if we denote by W_k the set of all possible shares $[k]_\Sigma$ of a secret value $b \in \mathbb{F}_q$, we have that $W_k = [k]_\Sigma + V$ is an affine subspace of $\mathbb{F}_q^\ell$, where $[k]_\Sigma$ denotes some share of k using Σ. Note that $W_0 = V$ and $W_0 \cup \cdots \cup W_{q-1} = \mathbb{F}_q^\ell$.

Lemma 2.3. *Let Σ be an $\mathbb{F}_q$-LSSS with $\dim E_0 = 1$. A subset $A \subseteq P$ is forbidden for Σ if and only if there exists a vector $\mathbf{r} \in \Omega$ for which $\mathtt{Share}_\Sigma(1, \mathbf{r}) = (x_1, \ldots, x_n)$ satisfies $x_i = \mathbf{0}$ for every $i \in A$.*

2.3 OT combiners

In 1-out-of-q OT protocols, the sender Alice is assumed to hold q messages $m_0, \ldots, m_{q-1}$, and the receiver Bob chooses a message index $b \in \mathbb{F}_q$. At the end of a protocol implementing this functionality, Bob receives m_b and Alice receives nothing.

Here we lay out the fundamental theory of OT combiners. Before proceeding further, and as in [16], we need to introduce the *ideal 1-out-of-q OT functionality $\mathcal{F}_{OT}$*. We make use of the ideal functionality $\mathcal{F}_{OT}$ in our correctness and security definitions. It consists of an ideal version of a 1-out-of-q OT protocol that implements the functionality correctly and that does not allow any kind of corruption. Hence, $\mathcal{F}_{OT}$ is an abstraction of an ideal OT protocol. Without loss of generality, in this work all 1-out-of-q OT protocols that are considered secure

Ideal 1-out-of-q OT Functionality $\mathcal{F}_{OT}$

Outline of the functionality:

1. The receiver Bob selects $b \in \mathbb{F}_q$, and sends its input $(\texttt{transfer}, b)$ to $\mathcal{F}_{OT}$.
2. The functionality $\mathcal{F}_{OT}$ sends $(\texttt{ready})$ to Alice.
3. The sender Alice sends $q-1$ messages $(\texttt{send}, m_0, \dots, m_{q-1})$ to $\mathcal{F}_{OT}$.
4. If $(\texttt{transfer}, b)$ has been received from Bob, $\mathcal{F}_{OT}$ sends $(\texttt{sent}, m_b)$ to Bob.

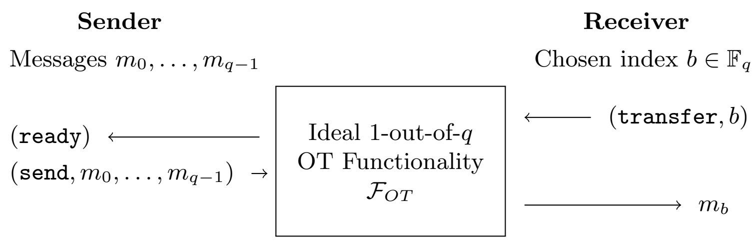

Fig. 2. The ideal 1-out-of-q Oblivious Transfer functionality.

are assumed to follow the blueprint of $\mathcal{F}_{OT}$. Figure 2 depicts the $\mathcal{F}_{OT}$ ideal functionality. Next, we formally define OT combiners, following the notation of [16]. Next, we formally define OT combiners, following the notation of [16].

Definition 2.4. *Let $S_1, \dots, S_n$ be candidate OT implementations. An* OT combiner *is an efficient two-party protocol $\pi = \pi(S_1, \dots, S_n)$, with access to the candidates $S_1, \dots, S_n$, that implements the OT functionality. An OT combiner is* 1-out-of-q *if it implements the* 1*-out-of-q OT functionality.*

From this point onward, we assume OT combiners to be 1-out-of-q, n-server, and black-box. Under this assumption, we can formalize the notion of OT combiner according to the following definition.

Definition 2.5. *We define an n-server, black-box,* 1*-out-of-*q OT combiner $\pi = \pi(S_1, \dots, S_n)$ *by means of the next three polynomial-time algorithms:*

$(b_1, \dots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$**:** *Probabilistic algorithm run by the receiver Bob and taking as input a message index $b \in \mathbb{F}_q$. It returns an n-tuple $(b_1, \dots, b_n)$, where each b_i is to be sent to server S_i.*

$(u_i^j)_{(i,j) \in \mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(m_0, \dots, m_{q-1})$**:** *Probabilistic algorithm run by the sender Alice, taking as input q chosen messages $m_0, \dots, m_{q-1}$. It returns a qn-tuple $(u_i^j)_{(i,j) \in \mathcal{P}_{n,q}}$, where each tuple $(u_i^0, \dots, u_i^{q-1})$ is to be sent to server S_i.*

$m \leftarrow \pi.\mathsf{Rec}(b, (v_1, \dots, v_n))$**:** *Algorithm run by the receiver Bob, that takes as input the chosen message index $b \in \mathbb{F}_q$ and n elements $v_1, \dots, v_n$, where each v_i is received from server S_i. It returns a message m.*

Given an OT combiner $\pi = (\pi.\mathsf{Choose}, \pi.\mathsf{Send}, \pi.\mathsf{Rec})$ and given n servers $S_1, \dots, S_n$ implementing the 1-out-of-q OT functionality, we regard π as a protocol between a sender Alice and a receiver Bob. In this case, the resulting OT protocol $\pi(S_1, \ldots, S_n)$ develops sequentially in five phases:

Choice phase: The receiver Bob chooses a message index $b \in \mathbb{F}_q$.
Bob generates the tuple $(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$.
Bob sends $(\mathtt{transfer}, b_i)$ to server S_i for $i = 1, \ldots, n$.

Ready phase: On receiving b_i from Bob, the server S_i sends $(\mathtt{ready})$ to Alice.

Sending phase: The sender Alice chooses q messages $m_0, \ldots, m_{q-1}$.
Alice generates the corresponding tuple

$$(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(m_0, \ldots, m_{q-1}).$$

After receiving $(\mathtt{ready})$ from every server, Alice sends the generated shares $(\mathtt{send}, u_i^0, \ldots, u_i^{q-1})$ to S_i for $i = 1, \ldots, n$.

Transfer phase: The server S_i sends $(\mathtt{sent}, u_i^{b_i})$ to Bob.

Output phase: Bob reconstructs the message m_b from the shares $u_1^{b_1}, \ldots, u_n^{b_n}$ he received by executing $\pi.\mathsf{Rec}(b, (u_1^{b_1}, \ldots, u_n^{b_n}))$.

3 Correctness and Security Definitions

In this section, we state the definitions used to capture the correctness and security properties of our constructions. In the following discussions, $\mathcal{P}_n$ represents the set of servers.

3.1 Correctness Definitions

The correctness property of OT combiners refers to the fact that, in the eyes of the receiver Bob, the produced protocol should always implement the OT functionality correctly. To define correctness, we need to consider two scenarios: one where the sender Alice follows the protocol honestly, and one where she may act maliciously.

In the first scenario all participants behave honestly. Here, we must ensure that, assuming all servers correctly implement the OT functionality, the protocol produced by the combiner implements the OT functionality correctly. Hence, we have to show that the message retrieved by Bob in the execution of the OT combiner is exactly the one that he should receive as per the OT functionality. This first approach to correctness is expressed by the zero-error property, which we formalize in the following definition. The definition is adapted from [16].

Definition 3.1. *An OT combiner π is* zero-error *if, for every message index $b \in \mathbb{F}_q$ and for any q messages $m_0, \ldots, m_{q-1}$, we have that*

$$m_b \leftarrow \pi.\mathsf{Rec}(b, (u_1^{b_1}, \ldots, u_n^{b_n})),$$

where $(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$ and $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(m_0, \ldots, m_{q-1})$.

In the second scenario, we consider a malicious sender Adv and an honest receiver $\mathbb{B}$. We assume that Adv corrupts a set $A \in \mathcal{A}$ of servers, where $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ is an adversary structure preset according to the threat model of Adv. We assume that Adv can see the inputs $(b_i)_{i\in A}$ of $\mathbb{B}$, and that she can also fix the messages $(z_i)_{i\in A}$ that $\mathbb{B}$ receives. Furthermore, she arbitrarily chooses inputs $(u_i^0, \dots, u_i^{q-1})_{i\in\bar{A}}$ for the non-corrupted servers in $\bar{A}$.

Here correctness states that, regardless of how the malicious sender generates input for each server, the obtained protocol is still an OT protocol. That is, the message index b chosen by $\mathbb{B}$ should determine one and only one message, even if it is malformed (i.e. $\perp$, due to the malicious behavior of Alice). In particular, the received message, which is computed using $\pi.\mathsf{Rec}$, should exclusively depend on b (and not on the randomness associated to the sharing of b sent by $\mathbb{B}$).

This second approach to correctness is formalized in the following definition, which uses the simulation paradigm [35], and which compares the execution of the protocol in the real world and in the ideal world.

In the real world, Adv and $\mathbb{B}$ interact through an OT combiner protocol π. The receiver $\mathbb{B}$ starts by choosing a message index $b \in \mathbb{F}_q$, and distributes each element b_i of the output of $\pi.\mathsf{Choose}(b)$ to each server. The adversary Adv is assumed to completely corrupt every server in a set $A \in \mathcal{A}$, and so she sees all the inputs $(b_i)_{i\in A}$ of $\mathbb{B}$ on those servers. Since the corruption is malicious, Adv also controls the outputs of servers in A, and so she chooses which output values z_i are received by $\mathbb{B}$ for $i \in A$. Non-corrupted servers $i \in \bar{A}$ are assumed to behave as the ideal $\mathcal{F}_{OT}$ functionality, so Adv sends q messages $u_i^0, \dots, u_i^{q-1}$ to each of them and learns no information from that interaction.

In the ideal world, the whole view and output of Adv is controlled by the simulator Sim, and Sim and $\mathbb{B}$ interact exclusively through the ideal OT functionality $\mathcal{F}_{OT}$. Because of this, the adversary Adv does not receive anything from the interaction. By processing all the output that the adversary Adv generates, Sim produces a set of messages $\tilde{m}_0, \dots, \tilde{m}_{q-1}$ and handles them to the $\mathcal{F}_{OT}$ functionality, which outputs the message $\tilde{m}_b$ to $\mathbb{B}$ for the requested message index $b \in \mathbb{F}_q$.

In order to ensure that π behaves as an OT protocol in this setting, we should guarantee the indistinguishability between the reconstruction output by $\mathbb{B}$ in the real world and the view of $\mathbb{B}$ in the ideal world.

Definition 3.2. *Let π be an n-server, 1-out-of-q OT combiner protocol, and let $\mathcal{F}_{OT}$ denote the ideal 1-out-of-q OT functionality. Let Adv denote an adversary-controlled malicious sender, which is assumed to corrupt the set of servers indexed by some set $A \in \mathcal{A}$. Let $\mathbb{B}$ denote the honest receiver, and let $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ be a stateful simulator. We define the probabilistic experiments $\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ and $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ as follows:*

$\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$:

$b \leftarrow \mathbb{B}()$

$(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$

$((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}) \leftarrow \mathsf{Adv}\,((b_i)_{i \in A})$

output $\pi.\mathsf{Rec}(b, ((u_i^{b_i})_{i \in \bar{A}}, (z_i)_{i \in A}))$

$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$:

$b \leftarrow \mathbb{B}()$

$(\textit{ready}) \leftarrow \mathcal{F}_{OT}(\textit{transfer}, b)$

$(b_i)_{i \in A} \leftarrow \mathsf{Sim}_1()$

$((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}) \leftarrow \mathsf{Adv}((b_i)_{i \in A})$

$(m_0, \ldots, m_{q-1}) \leftarrow \mathsf{Sim}_2((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A})$

$(\textit{sent}, m_b) \leftarrow \mathcal{F}_{OT}(\textit{send}, m_0, \ldots, m_{q-1})$

output m_b

We say that π implements the OT functionality correctly for the receiver against active $\mathcal{A}$-adversaries *if, for every set* $A \in \mathcal{A}$*, for all adversarial senders* Adv *corrupting the set of servers indexed by* A*, and for all honest receivers* $\mathbb{B}$*, there exists a simulator* Sim *such that the output values of* $\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ *and* $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ *are identically distributed, where the probabilities are taken over the random coins of* π*,* Adv*,* $\mathbb{B}$ *and* Sim*.*

3.2 Security Definitions

An OT combiner is *unconditionally secure* if its security rests solely on the security assumptions of the OT candidate implementations [16]. That is, if, provided the security of enough OT candidates holds, the resulting OT protocol is perfectly secure. Therefore, unconditional security guarantees that any attack on an OT combiner must forcibly break the security of sufficiently many of the OT candidate implementations in order to be successful. As in [16], an OT combiner is called *perfectly secure* if it is both unconditionally secure and correct.

In order to capture the notion of unconditional security, we formalize it into a simulator-based security definition [35]. We now give the definition of security that we employ in our work, namely *perfect security against active* $(\mathcal{A}, \mathcal{B})$*-adversaries*, which is adapted from [16] and uses the Universal Composability framework [14].

Given two adversary structures $\mathcal{A}, \mathcal{B} \subseteq 2^{\mathcal{P}_n}$, our security definition protects against two types of active adversaries: one that corrupts the sender Alice and a set of servers $A \in \mathcal{A}$, and one that corrupts the receiver Bob and a set of servers $B \in \mathcal{B}$. This respectively corresponds to the case that a set $A \in \mathcal{A}$ of the OT candidates are insecure for the receiver, and to the case that a set $B \in \mathcal{B}$ of the OT candidates are insecure for the sender. To deal with the Alice corruption case, we define the notion of *perfect security for the receiver against active* $\mathcal{A}$*-adversaries*, and in the Bob corruption case we define the notion of *perfect security for the sender against active* $\mathcal{B}$*-adversaries*.

In the Alice corruption case, we consider a malicious (i.e., active) adversary Adv that controls the sender Alice, that interacts with an honest receiver $\mathbb{B}$, and that is able to eavesdrop and fully operate each server in a set $A \in \mathcal{A}$. Our security aim here is to protect the confidentiality of the receiver's choice $b \in \mathbb{F}_q$. Hence, the ability to corrupt the servers in A must give Adv no information on b.

This definition uses the simulation paradigm [35], and compares the execution of the protocol in the real world and in the ideal world. In the real world, Adv and $\mathbb{B}$ interact through an OT combiner protocol π. The setting of this experiment is equivalent to that of Definition 3.2. In the ideal world, the whole view and output of Adv is controlled by the simulator Sim, and Sim and $\mathbb{B}$ interact exclusively through the ideal OT functionality $\mathcal{F}_{OT}$. Because of this, in the ideal experiment the adversary Adv does not receive any information on the choice $b \in \mathbb{F}_q$ of $\mathbb{B}$ from the interaction.

To provide security against malicious senders, Sim takes all the information viewed by Adv in the ideal world, which is the one herself produced, so as to transform it to a view that should be indistinguishable to the information seen by Adv in the real world. Hence, Sim also simulates the private inputs of $\mathbb{B}$ on the corrupted servers.

Definition 3.3. *Let π be an n-server, 1-out-of-q OT combiner protocol, and let $\mathcal{F}_{OT}$ denote the ideal 1-out-of-q OT functionality. Let Adv denote an adversary-controlled malicious sender, which is assumed to corrupt all the servers indexed by some set $A \in \mathcal{A}$. Let $\mathbb{B}$ denote an honest receiver, and let $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_{\mathsf{out}})$ be a stateful simulator. We define the probabilistic experiments $\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ and $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ as follows:*

$\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$:	$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$:
$b \leftarrow \mathbb{B}()$	$b \leftarrow \mathbb{B}()$
$(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$	$(\mathit{ready}) \leftarrow \mathcal{F}_{OT}(\mathit{transfer}, b)$
$((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}) \leftarrow \mathsf{Adv}\,((b_i)_{i \in A})$	$(b_i)_{i \in A} \leftarrow \mathsf{Sim}_1()$
output $((b_i)_{i \in A}, (u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A})$	$((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}) \leftarrow \mathsf{Adv}((b_i)_{i \in A})$
	output $\mathsf{Sim}_{\mathsf{out}}((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A})$

We say that π is perfectly secure for the receiver against active $\mathcal{A}$-adversaries *if, for every set $A \in \mathcal{A}$, for all adversarial senders Adv corrupting the set of servers indexed by A, and for all honest receivers $\mathbb{B}$, there exists a simulator Sim such that the output values of $\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ and $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ are identically distributed, where the probabilities are taken over the random coins of π, Adv, $\mathbb{B}$ and Sim.*

In the Bob corruption case, we consider a malicious (i.e., active) adversary Adv that controls the receiver Bob, that interacts with an honest sender $\mathbb{A}$, and that is able to eavesdrop on and fully operate each server in a set $B \in \mathcal{B}$. Our security aim here is to protect the confidentiality of the sender's messages $m_0, \ldots, m_{q-1}$. Hence, the ability to corrupt the servers in one set $B \in \mathcal{B}$ of servers must give Bob no information on $m_0, \ldots, m_{q-1}$ other than possibly one chosen message. As the previous definition, this definition uses the simulation paradigm [35] and compares the execution of the protocol in the real world and in the ideal world.

In the real world, $\mathbb{A}$ and Adv interact through an OT combiner protocol π. The sender $\mathbb{A}$, who is assumed to act honestly, holds messages $m_0, \ldots, m_{q-1}$ and uses the OT combiner π to generate the input $u_i^0, \ldots, u_i^{q-1}$ that is sent to server S_i for every $i \in \mathcal{P}_n$. The adversary Adv is assumed to completely corrupt every server in a set $B \in \mathcal{B}$, and so he sees all the inputs $(u_i^j)_{i\in B, j\in\mathbb{F}_q}$. He also acts as the receiver, generating an input b_i for the rest of servers $i \in \bar{B}$. Since the servers $i \in \bar{B}$ are assumed to behave as the ideal $\mathcal{F}_{OT}$ functionality, Adv receives $(u_i^{b_i})_{i\in\bar{B}}$ and learns no other information from that interaction.

In the ideal world, the whole view and output of Adv is controlled by the simulator Sim, and Sim and $\mathbb{A}$ interact through the ideal OT functionality $\mathcal{F}_{OT}$. By processing all the output that the adversary Adv generates, Sim produces a message index $\tilde{b}$ and handles it to the $\mathcal{F}_{OT}$ functionality. Then, after the sender $\mathbb{A}$ has sent the messages $m_0, \ldots, m_{q-1}$ to $\mathcal{F}_{OT}$, the adversary Adv receives the message $m_{\tilde{b}}$. To provide security against malicious receivers, Sim takes all the information viewed by Adv in the ideal world, so as to transform it to a view that should be indistinguishable to the one of the real world.

Definition 3.4. *Let π be an n-server, 1-out-of-q OT combiner, and let $\mathcal{F}_{OT}$ denote the 1-out-of-q OT functionality. Let Adv denote an adversary-controlled malicious receiver, which is assumed to corrupt all the servers indexed by some set $B \in \mathcal{B}$. Let $\mathbb{A}$ denote an honest sender, and let $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2, \mathsf{Sim}_{\mathsf{out}})$ be a stateful simulator. We define the probabilistic experiments $\mathsf{Real}^{\pi}_{\mathbb{A},\mathsf{Adv}}()$ and $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathbb{A},\mathsf{Adv},\mathsf{Sim}}()$ as follows:*

$\mathsf{Real}^{\pi}_{\mathbb{A},\mathsf{Adv}}()$:

$(m_0, \ldots, m_{q-1}) \leftarrow \mathbb{A}()$

$(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(\textit{send}, m_0, \ldots, m_{q-1})$

$(b_i)_{i\in\bar{B}} \leftarrow \mathsf{Adv}((u_i^j)_{i\in B, j\in\mathbb{F}_q})$

output $((u_i^j)_{i\in B, j\in\mathbb{F}_q}, (u_i^{b_i})_{i\in\bar{B}}, (b_i)_{i\in\bar{B}})$

$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathbb{A},\mathsf{Adv},\mathsf{Sim}}()$:

$(u_i^j)_{i\in B, j\in\mathbb{F}_q} \leftarrow \mathsf{Sim}_1()$

$(b_i)_{i\in\bar{B}} \leftarrow \mathsf{Adv}((u_i^j)_{i\in B, j\in\mathbb{F}_q})$

$\tilde{b} \leftarrow \mathsf{Sim}_2\left((b_i)_{i\in\bar{B}}\right)$

$(\textit{ready}) \leftarrow \mathcal{F}_{OT}(\textit{transfer}, \tilde{b})$

$(m_0, \ldots, m_{q-1}) \leftarrow \mathbb{A}()$

$(\textit{sent}, m_{\tilde{b}}) \leftarrow \mathcal{F}_{OT}(\textit{send}, m_0, \ldots, m_{q-1})$

output $\mathsf{Sim}_{\mathsf{out}}(\tilde{b}, m_{\tilde{b}}, (b_i)_{i\in\bar{B}})$

We say that π is perfectly secure for the sender against active $\mathcal{B}$-adversaries *if, for every $B \in \mathcal{B}$, for all adversarial receivers Adv corrupting the set of servers indexed by B, and for all honest senders $\mathbb{A}$, there exists a simulator Sim such that the output values of $\mathsf{Real}^{\pi}_{\mathbb{A},\mathsf{Adv}}()$ and $\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathbb{A},\mathsf{Adv},\mathsf{Sim}}()$ are identically distributed, where the probabilities are taken over the random coins of π, $\mathbb{A}$, Adv and Sim.*

The two previous definitions, on top of the correctness definitions, make up the security definition considered in this work, namely perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries. We next formally state this.

Definition 3.5. *Let π be an n-server, 1-out-of-q OT combiner, and let $\mathcal{A}, \mathcal{B} \subseteq 2^{\mathcal{P}_n}$. We say that π is* perfectly secure against active $(\mathcal{A}, \mathcal{B})$-adversaries *if it is perfectly secure for the sender against active $\mathcal{B}$-adversaries and for the receiver against active $\mathcal{A}$-adversaries, it is zero-error, and it implements the OT functionality correctly for the receiver against active $\mathcal{A}$-adversaries.*

4 Single-Use 1-out-of-q OT Combiners

In this section, we present our 1-out-of-q OT combiner in the particular setting where Σ is an ideal $\mathbb{F}_q$-LSSS. We achieve a single-use OT combiner with perfect security against active adversaries. In Section 7, we describe our construction for general LSSSs.

Single-use 1-out-of-q OT Combiner π_{OT}

π_{OT}.Choose(b)**:** Given $b \in \mathbb{F}_q$, compute a sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ of b using Σ.
Note that each $b_i \in \mathbb{F}_q$ because Σ is ideal.
Output $(b_1, \ldots, b_n)$.

π_{OT}.Send$(m_0, \ldots, m_{q-1})$**:** For each message m_k, independently compute a sharing

$$[m_k]_{\mathcal{S}_k} = (m_k^{(i,j)})_{(i,j)\in\mathcal{P}_{n,q}}.$$

Then, for each $(i,j) \in \mathcal{P}_{n,q}$, compute the values

$$u_i^j := m_0^{(i,j)}||m_1^{(i,j)}||\cdots||m_{q-1}^{(i,j)}.$$

Output $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}}$, where each tuple $(u_i^0, \ldots, u_i^{q-1})$ is to be sent to server S_i.

π_{OT}.Rec$(b, (v_1, \ldots, v_n))$**:** Parse each v_i, which is received from server S_i, as

$$v_i = n_0^{(i)}||n_1^{(i)}||\cdots||n_{q-1}^{(i)},$$

where $n_k^{(i)} \in \mathbb{F}_q$ for each $k \in \mathbb{F}_q$.
Retrieve m_b by evaluating

$$\texttt{Reconstruct}_{\mathcal{S}_b}((n_b^{(i)})_{i\in\mathcal{P}_n}).$$

If the reconstruction fails at any step, output $\perp$. Otherwise, output m_b.

When the protocol follows correctly, π_{OT}.Rec is executed with $(v_1, \ldots, v_n) = (u_1^{b_1}, \ldots, u_n^{b_n})$ and retrieves $\texttt{Reconstruct}_{\mathcal{S}_b}((m_b^{(1,b_1)}, \ldots, m_b^{(n,b_n)})$.

Fig. 3. Single-use 1-out-of-q OT combiner π_{OT} in the case Σ is an ideal $\mathbb{F}_q$-LSSS. The schemes $\mathcal{S}_k$ are defined in Figure 4.

The proposed 1-out-of-q OT combiner is shown in Figure 3. It is described according to Definition 2.5 and follows the structure described in Section 1.3.

The Secret Sharing Scheme $\mathcal{S}_k$

$\texttt{Share}_{\mathcal{S}_k}$**:** To share a message $m \in \mathbb{F}_q$, first

- let $\mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{F}_q^n$ be a sharing of k using Σ
- sample $r_1, \ldots, r_{n-1} \in \mathbb{F}_q$ uniformly at random, and let $r_n = m - \sum_{i=1}^{n-1} r_i$
- sample $\mathbf{h} = (h_1, \ldots, h_n)$ uniformly at random from $V^\perp$

For every $i \in \mathcal{P}_n$ and for every $j \in \mathbb{F}_q$, define the (i,j)-th share as

$$m^{(i,j)} = r_i + (k_i - j)h_i.$$

$\texttt{Reconstruct}_{\mathcal{S}_k}$**:** In this work we are just interested in reconstructing the message for subsets $A = \{(1, b_1), \ldots, (n, b_n)\}$ with $(b_1, \ldots, b_n) \in W_k$. In this case, the reconstruction function for $(m^{(i,j)})_{(i,j)\in A}$ is

$$\sum_{i=1}^{n} m^{(i,b_i)}.$$

Fig. 4. The $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ related to the secret sharing scheme Σ, the affine subspace $W_k = [k]_\Sigma$, and $V = [0]_\Sigma$.

Theorem 4.1. *Let Σ be an ideal $\mathbb{F}_q$-LSSS with adversary structure $\mathcal{A}$. The OT combiner π_{OT} defined in Figure 3 is perfectly secure against active $(\mathcal{A}, \mathcal{A}^*)$-adversaries.*

The proof of Theorem 4.1 is split in three blocks: First, in Section 5, we analyze the secret sharing schemes S_k. In Section 6, we prove the correctness of the protocol and then its security. Theorem 4.1 implies Theorem 1.1 for the case that Σ is an ideal LSSS, and the non-ideal case is implied by Theorem 7.1. At the end of Section 6 we prove Theorem 1.2 by combining these results.

Remark 4.2 (Structure). The protocol runs between a sender Alice and a receiver Bob, who communicate through a set of n servers $S_1, \ldots, S_n$ that implement the ideal 1-out-of-q OT functionality $\mathcal{F}_{OT}$ (described in Figure 2).

An ideal $\mathbb{F}_q$-LSSS Σ is used by the receiver Bob to request the message with the selected index $b \in \mathbb{F}_q$: He generates a sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ of b with Σ, and queries each server S_i with $b_i \in \mathbb{F}_q$. It corresponds to function $\pi_{OT}.\mathsf{Choose}$.

In order for Alice to distribute the messages $m_0, \ldots, m_{q-1}$, she makes use of q different secret sharing schemes $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$, which are related to the affine subspaces $W_0, \ldots, W_{q-1}$, respectively. It corresponds to function $\pi_{OT}.\mathsf{Send}$.

Then, the servers execute the OT functionality with the inputs they received, and send their outputs to Bob. Bob executes $\pi_{OT}.\mathsf{Rec}$ and retrieves the message.

Remark 4.3 (Communication complexity). In the choice phase, Bob sends a total of $n \log q$ bits to servers. In the sending phase, Alice sends a total of $q^2 n \log q$ bits to the servers. In the transfer phase, servers send a total of $n \log q$ bits to Bob. Hence, the communication complexity is $(q^2 + q + 1)n \log q$.

Remark 4.4 (Adversary structure). A protocol is secure against $(\mathcal{A}, \mathcal{A}^*)$-adversaries if and only if it is secure against $(\mathcal{A}, \mathcal{B})$-adversaries for any $\mathcal{R}_2$ pair $(\mathcal{A}, \mathcal{B})$. Also, it is equivalent to saying that it is secure against an adversary corrupting either Alice and a set of servers in $\mathcal{A}$, or Bob and a set of servers B with $\bar{B} \notin \mathcal{A}$, as stated in the abstract. This 1-out-of-q OT combiner is also secure against any $(\mathcal{A}', \mathcal{B}')$-adversary satisfying $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{A}^*$.

Remark 4.5 (Variants). The first observation is that the protocol can be run for messages that are shorter than $\log q$. For instance, if the messages are just one bit, the protocol is not affected and the communication complexity is the same.

The second observation is that the protocol can be adapted to run 1-out-of-q' OT combiner for $q' < q$. In this case, Bob will choose m_b among $m_1, \ldots, m_{q'}$. We attain the same level of security, but the communication is reduced. Next, we detail the changes that should be done.

The $\pi_{OT}.\mathsf{Send}$ algorithm does not change. The algorithm $\pi_{OT}.\mathsf{Send}$ has only q' inputs, so the values u_i^j will be shorter, i.e., $u_i^j = m_0^{(i,j)} || m_1^{(i,j)} || \cdots || m_{q'-1}^{(i,j)}$. The servers will perform the 1-out-of-q functionality, and they will send to Bob a shorter v_i. The algorithm $\pi_{OT}.\mathsf{Rec}$ is executed analogously. The communication complexity now is $n \log q + qq'n \log q + q'n \log q = (qq' + q' + 1)n \log q$.

The flexibility of the construction may be useful for certain adversary structures. It is known that there are adversary structures $(\mathcal{A}, \mathcal{A}^*)$ for which there only exist efficient $\mathbb{F}$-LSSSs Σ for fields of a certain characteristic [9].

Remark 4.6 (Multisecret sharing schemes). In the sending phase of our protocol, we share each of the q messages independently. For $q = 2$, this process was improved in [16] by creating sharings of the two messages at the same time, which reduces the number of shares from $4n$ to $2n$. The scheme in [16] can be seen as a *multi-secret* sharing scheme [12,32]. In such schemes, n shares are generated from a sequence of $k > 1$ secrets, and each secret can be recovered from the shares, but each secret has its own access structure. Observe that we can define our 1-out-of-q constructions from multi-secret sharing schemes. Since our multi-secret sharing scheme is just a combination of independent secret sharing schemes, we decided to simplify the notation. A research line in the direction of this work is to build more efficient 1-out-of-q OT-combiners with multi-secret sharing schemes.

5 Secret Sharing Schemes for OT Combiners

In this section, we introduce a family of secret sharing schemes that are useful to build 1-out-of-q OT combiners. We call them the OT-compatible secret sharing schemes. In Section 5.2, we show that the schemes $\mathcal{S}_k$ in Figure 4 are indeed W_k-OT-compatible. This fact simplifies the security proof of our 1-out-of-q OT combiners.

5.1 W-OT-Compatible Secret Sharing Schemes

Recall the discussion in Section 1.3 about the properties involved in our construction. This section is dedicated to the study of the schemes $\mathcal{S}_i$ that guarantee the correctness and security of our protocol. Consider the following definition.

Definition 5.1. *Let F_q be a finite field, and let $W \subseteq \mathbb{F}_q^n$. We define Γ_W as the access structure on $\mathcal{P}_{n,q}$ determined by the minimal access structure*

$$\min \Gamma_W = \{\{(1, b_1), (2, b_2), \dots, (n, b_n)\} : (b_1, b_2, \dots, b_n) \in W\}.$$

The study of the share size of access structures Γ_W for general W is of independent interest. If $n = 2$, then Γ_W is a bipartite graph access structure, and this case is studied in several works as [6,20]. As a consequence of [36,2,3], improvements on the efficiency of schemes for Γ_W will result in improvements in the efficiency of CDS protocols, and in the efficiency of secret sharing schemes for general access structures.

Following the discussion in Section 1.3, we are here interested in the construction of secret sharing schemes with access structure Γ_W where W is the collection of sharings of b by Σ. Since we only consider linear schemes, $[b]_\Sigma$ is always a $\mathbb{F}_q$ affine space. Hence, from now on, we restrict the study of the access structures Γ_W when W is an affine space.

If $W \subseteq \mathbb{F}_2^n$ is a binary affine subspace, then the access structure Γ_W described above always admits an ideal $\mathbb{F}_2$-LSSS [16]. However, in general, given an affine subspace $W \subseteq \mathbb{F}_q^n$, ideal $\mathbb{F}_q$-LSSS for the access structure Γ_W are not expected to exist.

Instead of looking for $\mathbb{F}_q$-LSSS with access structures of the form Γ_W, which will derive non-efficient OT combiners, we explore the possibility of relaxing the restrictions on the access structure of $\mathcal{S}_i$, while maintaining the our security needs. With the aim of building schemes $\mathcal{S}_i$ for the protocol in Figure 3, we define the notion of *W-OT-compatibility.*

Definition 5.2. *Let $W \subseteq \mathbb{F}_q^n$. Let $\Delta \subseteq 2^{\mathcal{P}_{n,q}}$ be the family of subsets defined by*

$$\Delta = \{A_1 \cup \ldots \cup A_n : A_i \subseteq P_i \text{ and } |A_i| = 0, 1 \text{ or } q \text{ for } i = 1, \dots, n\}.$$

We say that an access structure $\Gamma \subseteq 2^{\mathcal{P}_{n,q}}$ is W-OT-compatible *if $\Gamma \cap \Delta = \Gamma_W \cap \Delta$. Similarly, we say that a secret sharing scheme is* W-OT-compatible *if its access structure is* W-OT-compatible.

The motivation behind this definition is the following: The secret sharing schemes $\mathcal{S}_0, \dots, \mathcal{S}_{q-1}$ to be used by Alice that we design are built so that an adversary controlling Bob, and possibly some servers, can learn from each server S_i either

- no shares, e.g. in the case where an active adversary corrupts Alice and S_i,
- one share, e.g. in the case that the server S_i is not corrupted, or
- all q shares sent to S_i, in the case that an adversary corrupts Bob and S_i.

Under this assumption, since P_i corresponds to the shares sent to server S_i, the shares that an adversary controlling Bob is able to see in any execution of the OT combiner are always determined by some subset of Δ. Therefore, even if the obtained $\mathbb{F}_q$-LSSS has an access structure Γ other than Γ_W, it serves our security purposes as long as Γ coincides with Γ_W when restricting it to Δ. That is, as long as Γ is W-OT-compatible.

We next state some properties of W-OT-compatible access structures.

Remark 5.3. If an access structure $\Gamma \subseteq 2^{\mathcal{P}_{n,q}}$ is W-OT-compatible, then

– $\min \Gamma_W \subseteq \min \Gamma$. Hence,

$$\{(1, b_1), \dots, (n, b_n)\} \in \Gamma \text{ for every } (b_1, \dots, b_n) \in W.$$

– $\{(1, v_1), \dots, (n, v_n)\} \notin \Gamma$ for every $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{F}_q^n \setminus W$.
– If $A \in \mathcal{P}_{n,q}$ has size $|A| < n$, then $A \notin \Gamma$.
– If $A \in \Gamma$ has size $|A| = n$, then $A \in \min \Gamma_W$ and $A \in \min \Gamma$.
– $\mathcal{P}_{n,q} \setminus P_i \notin \Gamma$ for $i = 1, \dots, n$.

In the full version of the paper [24] we give some examples of W-OT-compatible access structures.

5.2 Analysis of $\mathcal{S}_k$

This subsection is dedicated to the analysis the scheme $\mathcal{S}_k$ presented in Figure 4. It is an ideal $\mathbb{F}_q$-LSSS defined on the set of nq participants $\mathcal{P}_{n,q}$. It is used by Alice to generate a sharing of the k-th message m_k, which is distributed among the OT servers. Proposition 5.5 states that $\mathcal{S}_k$ is W_k-OT-compatible for $W_k = [k]_\Sigma$. First, we present a technical lemma needed for its proof.

Lemma 5.4. *Let $\mathbb{F}_q$ be a finite field, and $V \subsetneq \mathbb{F}_q^n$ be a vector subspace. Let $t \leq n$ and $y_1, \dots, y_t \in \mathbb{F}_q$. If $(y_1, \dots, y_t, x_{t+1}, \dots, x_n) \notin V$ for every $x_{t+1}, \dots, x_n \in \mathbb{F}_q$, then there exists $\mathbf{h} \in V^\perp$ such that $y_1 h_1 + \cdots + y_t h_t = 1$ and $h_{t+1} = \cdots = h_n = 0$.*

Proof. We prove this lemma by backward induction in t. The lemma holds for $t = n$ since, given $y = (y_1, \dots, y_n) \notin V$, there always exists an $\mathbf{h} \in V^\perp$ such that $\langle y, \mathbf{h} \rangle = 1$.

Assume that $t < n$. Suppose that there exist $y_1, \dots, y_t \in \mathbb{F}_q$ satisfying $(y_1, \dots, y_t, x_{t+1}, \dots, x_n) \notin V$ for all $x_{t+1}, \dots, x_n \in \mathbb{F}_q$. By induction hypothesis we have that, for every $x \in \mathbb{F}_q$, there exists an $\mathbf{h}^x = (h_1^x, \dots, h_n^x) \in V^\perp$ with

$$\sum_{i=1}^{t} y_i h_i^x + x h_{t+1}^x = 1 \quad \text{ and } \quad h_{t+2}^x = \cdots = h_n^x = 0.$$

If $h_{t+1}^x = 0$ for some $x \in \mathbb{F}_q$, then $\mathbf{h}^x$ satisfies the lemma. Otherwise, by the pigeonhole principle, let x and z be two distinct elements of $\mathbb{F}_q$ such that $h_{t+1}^x = h_{t+1}^z \neq 0$. Define

$$\mathbf{h} = \frac{\mathbf{h}^x - \mathbf{h}^z}{h_{t+1}^x (z - x)}.$$

Since $\mathbf{h} = (h_1, \ldots, h_n)$ is in $V^\perp$ and satisfies $h_{t+1} = \cdots = h_n = 0$ and

$$\begin{aligned} y_1 h_1 + \cdots + y_t h_t &= \frac{1}{h_{t+1}^x (z - x)} \left(\sum_{i=1}^{t} y_i h_i^x - \sum_{i=1}^{t} y_i h_i^z \right) \\ &= \frac{1}{h_{t+1}^x (z - x)} \big((1 - x h_{t+1}^x) - (1 - z h_{t+1}^z) \big) = 1, \end{aligned}$$

we have that $\mathbf{h}$ satisfies the lemma. □

Proposition 5.5. *Let Σ be an ideal $\mathbb{F}_q$-LSSS. For every $k \in \mathbb{F}_q$, the secret sharing scheme $\mathcal{S}_k$ defined in Figure 4 is an ideal $\mathbb{F}_q$-LSSS that is W_k-OT-compatible.*

Proof. In order to prove that $\mathcal{S}_k$ is W_k-OT-compatible, we prove that Γ, the access structure of $\mathcal{S}_k$, satisfies $\Gamma_{W_k} \cap \Delta = \Gamma \cap \Delta$.

Let $A = \{(1, b_1), \ldots, (n, b_n)\}$ be a subset in $\min \Gamma_{W_k}$, and let $(m^{(i,j)})_{(i,j) \in A}$ be shares by $\mathcal{S}_k$. Since $\sum_{i=1}^{n} r_i = m$, then

$$\sum_{i=1}^{n} m^{(i,b_i)} = \sum_{i=1}^{n} (r_i + (k_i - b_j) h_i) = \sum_{i=1}^{n} r_i + \langle \mathbf{k} - \mathbf{b}, \mathbf{h} \rangle = m,$$

where we used that $\mathbf{h} \in V^\perp$ and $\mathbf{k} - \mathbf{b} \in V$ because $\mathbf{k}, \mathbf{b} \in W_k$. Hence, $\Gamma_{W_k} \subseteq \Gamma$ and so $\Gamma_{W_k} \cap \Delta \subseteq \Gamma \cap \Delta$.

Now, we prove $\Gamma_{W_k} \cap \Delta \supseteq \Gamma \cap \Delta$ by showing that, for every $A \in \Delta$, if $A \notin \Gamma_{W_k}$ then $A \notin \Gamma$. Let $A \in \Delta \setminus \Gamma_{W_k}$. If $A \cap P_i = \emptyset$ for some i, then A is forbidden for $\mathcal{S}_k$, so $A \notin \Gamma$. Otherwise assume, without loss of generality, that for some $t \leq n$ we can express

$$A = \{(1, v_1), \ldots, (t, v_t)\} \cup P_{t+1} \cup \cdots \cup P_n.$$

We now use Lemma 2.3. More concretely, we show that there exists randomness $r_1, \ldots, r_{n-1} \in \mathbb{F}_q$ and $\mathbf{h} = (h_1, \ldots, h_n) \in V^\perp$ so that the sharing of the message $m = 1$ satisfies $m^{(i,j)} = 0$ for every $(i, j) \in A$.

Since $A \notin \Gamma_{W_k}$, we have that $(v_1 - k_1, \ldots, v_t - k_t, x_{t+1}, \ldots, x_n) \notin V$ for every $x_{t+1}, \ldots, x_n \in \mathbb{F}_q$. By Lemma 5.4, there exists an $\mathbf{h} = (h_1, \ldots, h_n) \in V^\perp$ such that $\sum_{i=1}^{t} (v_i - k_i) h_i = 1$ and $h_{t+1} = \cdots = h_n = 0$. We choose as randomness this $\mathbf{h} \in V^\perp$ and

$$\begin{aligned} r_i &= -(k_i - v_i) h_i && \text{for } i = 1, \ldots, t, \\ r_i &= 0 && \text{for } i = t+1, \ldots, n-1. \end{aligned}$$

Since we want a sharing of the message $m = 1$, we take $r_n = 1 - \sum_{i=1}^{n-1} r_i = 0$. Then, for $(i, j) \in A$, the shares $m_{(i,j)} = r_i + (k_i - j) h_i$ of the message $m = 1$ are

$$\begin{aligned} m^{(i,v_i)} &= -(k_i - v_i) h_i + (k_i - v_i) h_i = 0 && \text{for } i = 1, \ldots, t, \\ m^{(i,j)} &= 0 + (k_i - j) \cdot 0 = 0 && \text{for } i = t+1, \ldots, n. \end{aligned}$$

□

For the particular case $q = 2$, the proofs of Lemma 5.4 and Proposition 5.5 can be simplified [15]. Indeed, for $q = 2$, the access structure of the scheme in Figure 4 is simply Γ_{W_k} [15].

As a consequence of this result, for every affine space $W \subseteq \mathbb{F}_q^n$ there exists an ideal $\mathbb{F}_q$-LSSS that is W-OT-compatible.

6 Correctness and security proofs

We start with the proof of correctness in the setting where all parties follow the OT combiner protocol honestly.

Proposition 6.1. *The OT combiner π_{OT} defined in Figure 3 is zero-error. That is, provided both Alice and Bob are semi-honest, π_{OT} implements the* 1*-out-of-q OT functionality correctly.*

Proof. If Alice and Bob follow the protocol honestly, at the end of the protocol Bob receives the shares $m_b^{(1,b_1)}, \dots, m_b^{(n,b_n)}$ of the message m_b, where $[b]_\Sigma = (b_1, \dots, b_n) \in W_b$ is some sharing of his input b. Since $\{(1, b_1), \dots, (n, b_n)\} \in \min \Gamma_{W_b}$ is authorized for $\mathcal{S}_b$, Bob can reconstruct the message m_b. □

Now, we consider the case of Definition 3.2, where Alice is controlled by an active adversary Adv.

Proposition 6.2. *Let Σ be an ideal $\mathbb{F}_q$-LSSS with adversary structure $\mathcal{A}$. Then the OT combiner π_{OT} defined in Figure 3 implements the OT functionality correctly for the receiver against active $\mathcal{A}$-adversaries.*

Proof. We start by defining the simulator appearing in Definition 3.2, and we then compare the output of the ideal experiment to that of the real experiment in the security definition.

$\mathsf{Sim}_1()$: Generate a uniformly random sharing of $0 \in \mathbb{F}_q$, $[0]_\Sigma = (b_1^0, \dots, b_n^0)$. Output $(b_i^0)_{i \in A}$.

$\mathsf{Sim}_2((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (u_i)_{i \in A})$: Retrieve, from the state of Sim, the sharing $[0]_\Sigma = (b_i^0)_{i \in \mathcal{P}_n}$ that was previously generated in the execution of Sim_1.
Generate a uniformly random sharing $[k]_\Sigma = (b_1^k, \dots, b_n^k)$ of every nonzero element $k \in \mathbb{F}_q \setminus \{0\}$, subject to the restriction that $b_i^k = b_i^0$ for every $i \in A$. Note that these sharings exist, because A is forbidden for Σ. In practice, this step requires solving a compatible system of $|A|$ linear equations.
Parse each u_i^j as $u_i^j = m_0^{(i,j)} || \cdots || m_{q-1}^{(i,j)}$ whenever it is possible. If some u_i^j is not of this form (as it has been malformed by Alice), set $m_k = \perp$ for every $k \in \mathbb{F}_q$ such that $b_i^k = j$.
For every $k \in \mathbb{F}_q$, if m_k has not already been set to $\perp$ in the previous step, then try to reconstruct Alice's input by executing

$$\mathtt{Reconstruct}_{\mathcal{S}_k}\left((m_k^{(i,b_i^k)})_{i \in \mathcal{P}_n}\right).$$

If the reconstruction succeeds, let m_k be its output. Otherwise, set $m_k = \perp$. Output $(m_0, \dots, m_{q-1})$.

In order to prove indistinguishability remind that, in the real world, Bob generates a sharing $[b]_\Sigma = (b_1, \dots, b_n)$ of his input $b \in \mathbb{F}_q$. Note that the shares $(b_i)_{i \in A}$ correspond to the set $A \in \mathcal{A}$, which is forbidden for Σ. Hence, they are distributed identically to the A-shares in a uniformly random sharing of any other $b' \neq b$.

Because of the previous observation, the messages $\left((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\right)$ generated by Adv are identically distributed in both the real and the ideal world. Also because of the previous observation, the A-shares of the sharing $[b]_\Sigma = (b_1, \dots, b_n)$ generated in the real world and the shares $(b_i)_{i \in A}$ generated by Sim in the ideal world are identically distributed.

Therefore, the reconstruction of the messages m_b is carried in exactly the same way in the real and the ideal world. This proves indistinguishability. □

The security properties of our constructions are stated in Theorem 4.1. To proceed with its proof, we first need to establish Lemma 6.3.

Suppose that an adversary controlling Bob corrupts a set $B \in \mathcal{B}$ of servers. As a consequence of the next lemma, if the shares $(b_i)_{i \in \bar{B}}$ sent to non-corrupted servers in $\bar{B}$ do not correspond to any sharing $[b]_\Sigma$ of b, the adversary can not get any information on the message m_b.

Lemma 6.3. *Let $m_0, \dots, m_{q-1} \in \mathbb{F}_q$ be arbitrary messages, and fix independent sharings $[m_k]_{\mathcal{S}_k} = (m_k^{(i,j)})_{(i,j) \in \mathcal{P}_{n,q}}$ for every $k \in \mathbb{F}_q$. Let $B \subseteq \mathcal{P}_n$ and $(b'_1, \dots, b'_n) \in \mathbb{F}_q^n$, and define the set $\mathcal{H} \subseteq \mathcal{P}_{n,q}$ by*

$$\mathcal{H} = \{(i, b'_i) \,:\, i \in \bar{B}\} \cup \{(i,j) \,:\, i \in B,\ j \in \mathbb{F}_q\}.$$

Fix $b \in \mathbb{F}_q$. Then, if the shares $(b'_i)_{i \in \bar{B}}$ are not part of any sharing $[b]_\Sigma$, the shares $\{m_k^{(i,j)} \,:\, (i,j) \in \mathcal{H},\ k \in \mathbb{F}_q\}$ give no information about m_b.

Proof. Since the sharing of every message is done independently, only the shares $(m_b^{(i,j)})_{(i,j) \in \mathcal{H}}$ could potentially give information on m_b. We prove that $\mathcal{H}$ is forbidden for $\mathcal{S}_b$. Since $\mathcal{S}_b$ is W_b-OT-compatible and since $\mathcal{H} \in \Delta$, if $\mathcal{H}$ were authorized for $\mathcal{S}_b$ then $\mathcal{H} \in \Gamma_{W_b}$, and thus it would contain a set $\{(1, b_1), \dots, (n, b_n)\}$ for some $(b_1, \dots, b_n) \in W_b$. However, then necessarily $b_i = b'_i$ for all $i \in \bar{B}$, and this would mean that $(b'_i)_{i \in \bar{B}}$ belongs to a sharing $[b]_\Sigma$, a contradiction. □

Now we can complete the proof of Theorem 4.1.

Proof (Proof of Theorem 4.1). Correctness is proved in Propositions 6.1 and 6.2. The rest of the proof is split in two parts, corresponding to Definitions 3.3 and 3.4. In each case, we define the simulators and compare the outputs of the ideal and real experiments. Let $A \in \mathcal{A}$ and $B \in \mathcal{A}^*$.

Perfect security for the receiver against active $\mathcal{A}$-adversaries:

$\mathsf{Sim}_1()$: Generate a sharing of $0 \in \mathbb{F}_q$,

$$[0]_\Sigma = (b_1^0, \dots, b_n^0).$$

Output $(b_i^0)_{i \in A}$.

$\mathsf{Sim}_{\mathsf{out}}((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A})$: Retrieve, from the state of Sim, the sharing $[0]_\Sigma = (b_i^0)_{i \in \mathcal{P}_n}$ that was generated in the previous execution of Sim_1.

Output $\Big((b_i^0)_{i \in A}, (u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\Big)$.

Note that the shares $(b_i)_{i \in A}$ that the adversary Adv takes as input correspond to the set $A \in \mathcal{A}$, which is forbidden for Σ. Hence, they are distributed identically to the A-shares of a sharing of any other $b' \neq b$ (in particular, of $0 \in \mathbb{F}_q$), and so they do not carry any information on b. The messages $\Big((u_i^j)_{i \in \bar{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\Big)$ generated by Adv are thus identically distributed in both worlds.

Since the shares $(b_i)_{i \in A}$ do not allow to distinguish between the real and the ideal world, we have proved indistinguishability.

Perfect security for the sender against active $\mathcal{B}$-adversaries:

$\mathsf{Sim}_1()$: For every $k \in \mathbb{F}_q$, choose $m'_k \in \mathbb{F}_q$ uniformly at random and generate the sharing $[m'_k]_{\mathcal{S}_k} = ({m'}_k^{(i,j)})_{(i,j) \in \mathcal{P}_{n,q}}$. Then, create the values $u_i^j = {m'}_0^{(i,j)} || \cdots || {m'}_{q-1}^{(i,j)}$ for every $(i,j) \in B \times \mathbb{F}_q$. Output $(u_i^j)_{i \in B, j \in \mathbb{F}_q}$.

$\mathsf{Sim}_2((b_i)_{i \in \bar{B}})$: Try to reconstruct the input b of the adversary Adv by executing $\texttt{Reconstruct}_\Sigma$ on the input. If the reconstruction succeeds, output the reconstructed message index $\tilde{b}$. Otherwise, output $\perp$.

$\mathsf{Sim}_{\mathsf{out}}(\tilde{b}, m_{\tilde{b}}, (b_i)_{i \in \bar{B}})$: Retrieve, from the state of Sim and for every $k \in \mathbb{F}_q$, the messages m'_k, the sharings $[m'_k]_{\mathcal{S}_k} = ({m'}_k^{(i,j)})_{(i,j) \in \mathcal{P}_{n,q}}$ and the messages $(u_i^j)_{i \in B, j \in \mathbb{F}_q}$ that were generated in the previous execution of Sim_1.

Proceed as follows, depending on whether the reconstruction in Sim_2 failed or not:

- If $\tilde{b} \neq \perp$, let $\tilde{m}_{\tilde{b}} = m_{\tilde{b}}$. Generate a sharing $[\tilde{m}_{\tilde{b}}]_{\mathcal{S}_{\tilde{b}}} = (\tilde{m}_{\tilde{b}}^{(i,j)})_{(i,j) \in \mathcal{P}_{n,q}}$ subject to the restriction that $\tilde{m}_{\tilde{b}}^{(i,j)} = {m'}_{\tilde{b}}^{(i,j)}$ for every $(i,j) \in B \times \mathbb{F}_q$. Note that this is possible, since $B \times \mathbb{F}_q$ is forbidden for $\mathcal{S}_0, \dots, \mathcal{S}_{q-1}$. For every $k \in \mathbb{F}_q \setminus \{\tilde{b}\}$ and every $(i,j) \in \bar{B} \times \mathbb{F}_q$, set $\tilde{m}_k^{(i,j)} := {m'}_k^{(i,j)}$.
- If $\tilde{b} = \perp$, for every $k \in \mathbb{F}_q$ and $(i,j) \in \bar{B} \times \mathbb{F}_q$, let

$$\tilde{m}_k^{(i,j)} = {m'}_k^{(i,j)}.$$

Create the values $u_i^{b_i} = \tilde{m}_0^{(i,b_i)} || \cdots || \tilde{m}_{q-1}^{(i,b_i)}$ for every $i \in \bar{B}$.

Output $\Big((u_i^j)_{i \in B, j \in \mathbb{F}_q}, (u_i^{b_i})_{i \in \bar{B}}, (b_i)_{i \in \bar{B}}\Big)$.

In order to prove indistinguishability we first note that, since $B \in \mathcal{A}^*$, the set $\bar{B}$ is not in $\mathcal{A}$ and so it is authorized for Σ. By the definition of $\mathcal{S}_k$, we see that at least one share per server is needed to reconstruct a message. Hence, the set $B \times \mathbb{F}_q$ is forbidden for $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$, and so the shares $(u_i^j)_{i \in B, j \in \mathbb{F}_q}$ do not hold any information on the messages $m_0, \ldots, m_{q-1}$. Therefore, the shares $(b_i)_{i \in \bar{B}}$ generated by the adversary Adv in the real world and in the ideal world are identically distributed.

Since $\bar{B}$ is authorized for Σ, we have two possibilities as for the shares $(b_i)_{i \in \bar{B}}$ received by Sim: either they are part of a sharing $[b]_\Sigma$, or they are not part of any sharing for Σ (due to the malicious behavior of Adv).

In the first case, Sim_2 successfully reconstructs b. The set $\{(i, b_i) \ : \ i \in \bar{B}\} \cup (B \times \mathbb{F}_q)$ is then authorized for $\mathcal{S}_b$ and, by Lemma 6.3, it is forbidden for all the other $\mathbb{F}_q$-LSSS $\mathcal{S}_k$. Since the sharings for m_b generated by $\mathsf{Sim}_{\mathsf{out}}$ are distributed identically to those of the real world, this proves indistinguishability.

In the second case, Lemma 6.3 shows that the shares corresponding to the participants $\{(i, b_i) \ : \ i \in \bar{B}\} \cup (B \times \mathbb{F}_q)$ give no information about any message in the real world. Therefore, since here $\mathsf{Sim}_{\mathsf{out}}$ generates these shares from random messages in the ideal world, they obey the same distribution as in the real world, as required. $\square$

Finally, we can prove Theorem 1.2.

Proof (Theorem 1.2). For $q > n$, we choose Σ to be the Shamir secret sharing scheme [45] over $\mathbb{F}_q$ with adversary structure $\mathcal{A} = \{A \subseteq \mathcal{P}_n \ : \ |A| < n/2\}$. If n is odd, then $\mathcal{A} = \mathcal{A}^*$; else $\mathcal{A}^* = \{A \subseteq \mathcal{P}_n \ : \ |A| \leq n/2\}$. For $q = n$, we choose the canonical one-participant-extension of the the Shamir secret sharing scheme with adversary structure $\mathcal{A}$. In both cases, Σ is ideal and $\mathbb{F}_q$-linear. Then the result follows from and Theorem 4.1 and Remark 4.3. $\square$

7 Our Multi-Use, One-out-of-q OT Combiner

In this section, we show how our protocol π_{OT} from Section 4 extends to the general case where the adversary structure $\mathcal{A}$ does not necessarily admit an ideal $\mathbb{F}_q$-LSSS. Theorems 4.1 and 7.1 imply Theorem 1.1.

7.1 OT-Compatible Secret Sharing Schemes

Let Σ be an $\mathbb{F}_q$-LSSS for n participants with adversary structure $\mathcal{A}$. Since Σ is now not necessarily ideal, if $[b]_\Sigma = (\tilde{b}_1, \ldots, \tilde{b}_n)$ is a sharing of b using Σ, we note that each share $\tilde{b}_i$ belongs to some vector space $E_i = \mathbb{F}_q^{\ell_i}$ for some integer $\ell_i \geq 1$. Hence, unlike in the ideal case, $\tilde{b}_i$ may not correspond to a single message index but a sequence of them.

Denote by $\ell = \sum_{i=1}^n \ell_i$ the normalized share size of Σ. Rather than looking at the sharings $(\tilde{b}_1, \ldots, \tilde{b}_n)$ as elements of $\mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$, we concatenate their components and we see them as elements of the vector space $\mathbb{F}_q^\ell$. Denote the corresponding vector space isomorphism by

$$\varphi : \mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n} \to \mathbb{F}_q^\ell.$$

According to this, given Σ with the $\texttt{Share}_\Sigma$ function, we can define the ideal scheme Σ' on $\mathcal{P}_\ell = \{1, \dots, \ell\}$ with share spaces $E'_i = \mathbb{F}_q$ for every i, satisfying that $[b]_{\Sigma'} = \varphi([b]_\Sigma) = (b_1, \dots, b_\ell)$ for every $b \in \mathbb{F}_q$, where each $b_i \in \mathbb{F}_q$.

As in the previous section, let $V \subseteq \mathbb{F}_q^\ell$ denote the vector space consisting of all the sharings of 0 under the scheme Σ'. Given any $b \in \mathbb{F}_q$, let $W_b \subseteq \mathbb{F}_q^\ell$ be the affine subspace of sharings of b for Σ'.

Given the $\mathbb{F}_q$-LSSS Σ' and $k \in \mathbb{F}_q$, we instantiate the $\mathbb{F}_q$-LSSS $\mathcal{S}'_k$ associated to the affine subspace W_k in Figure 5. The scheme $\mathcal{S}'_k$ is now defined on the set of ℓq participants $\mathcal{P}_{\ell,q}$ and it is $\mathbb{F}_q$-linear and ideal.

The Secret Sharing Scheme $\mathcal{S}'_k$

$\texttt{Share}_{\mathcal{S}'_k}$**:** To share a message $m \in \mathbb{F}_q$, first

- let $\mathbf{k} = (k_1, \dots, k_\ell) \in \mathbb{F}_q^\ell$ be a sharing of k using Σ'
- sample $r_1, \dots, r_{\ell-1} \in \mathbb{F}_q$ uniformly at random, and let $r_\ell = m - \sum_{i=1}^{\ell-1} r_i$
- sample $\mathbf{h} = (h_1, \dots, h_\ell)$ uniformly at random from $V^\perp$

For every $i \in \mathcal{P}_\ell$ and for every $j \in \mathbb{F}_q$, define the (i, j)-th share as

$$m^{(i,j)} = r_i + (k_i - j)h_i.$$

$\texttt{Reconstruct}_{\mathcal{S}'_k}$**:** Analogously to $\mathcal{S}_k$ (Figure 4), it is enough to define the reconstruction function for subsets $A = \{(1, b_1), \dots, (\ell, b_\ell)\}$ in $\min \Gamma_{W_k}$. The reconstruction function for $(m^{(i,j)})_{(i,j)\in A}$ is

$$\sum_{i=1}^{\ell} m^{(i,b_i)}.$$

Fig. 5. The $\mathbb{F}_q$-LSSS $\mathcal{S}'_k$ related to the affine subspace $W_k \subseteq \mathbb{F}_q^\ell$.

The reconstruction function of $\mathcal{S}'_k$ in Figure 5 is analogous to the one in Figure 5. Note that it effectively retrieves m_k because

$$\sum_{i=1}^{\ell} m_k^{(i,b_i)} = \sum_{i=1}^{\ell} (r_i + (k_i - b_i)h_i) = \sum_{i=1}^{\ell} r_i + \langle \mathbf{k} - \mathbf{b}, \mathbf{h} \rangle = m_k.$$

As a direct consequence of Proposition 5.5 we have that, for every $k \in \mathbb{F}_q$, the secret sharing schemes $\mathcal{S}'_k$ are $\mathbb{F}_q$-linear, perfect, ideal and W_k-OT-compatible.

7.2 The Protocol

We now generalize the 1-out-of-q OT combiner presented previously to the case where Σ is not ideal. The obtained OT combiner is still black-box and n-server, but it is no longer single-use, because we assume that each of the n OT candidate servers S_i are called a total of ℓ_i times. To this end, for $i \in \mathcal{P}_n$, denote by

$I_i = \left\{\sum_{j=1}^{i-1} \ell_i + 1, \ldots, \sum_{j=1}^{i} \ell_i\right\}$ the set of participants of $\mathcal{P}_\ell$ whose shares are associated to the same server S_i. We describe our multi-use OT combiner in Figure 6.

Our 1-out-of-q OT Combiner Protocol π'_{OT}

$\pi'_{OT}.\mathsf{Choose}(b)$**:** Given $b \in \mathbb{F}_q$, compute a sharing $[b]_{\Sigma'} = (b'_1, \ldots, b'_\ell)$ of b using Σ'.
Note that each $b'_i \in \mathbb{F}_q$ because Σ' is ideal.
Output $(b_1, \ldots, b_n)$, where $b_i = (b'_j)_{j \in I_i}$.

$\pi'_{OT}.\mathsf{Send}(m_0, \ldots, m_{q-1})$**:** For each message m_k, independently compute a sharing

$$[m_k]_{\mathcal{S}'_k} = (m_k^{(i,j)})_{(i,j)\in\mathcal{P}_{\ell,q}}.$$

Then, for every $(i,j) \in \mathcal{P}_{\ell,q}$, compute the values

$$w_i^j := m_0^{(i,j)} || m_1^{(i,j)} || \cdots || m_{q-1}^{(i,j)}.$$

Output $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}}$, where $u_i^j = (w_k^j)_{k\in I_i}$.

$\pi'_{OT}.\mathsf{Rec}(b, (v_1, \ldots, v_\ell))$**:** Parse each v_i as $v_i = (v'_k)_{k\in I_i}$, where

$$v'_k = n_0^{(k)} || n_1^{(k)} || \cdots || n_{q-1}^{(k)},$$

and where $n_j^{(k)} \in \mathbb{F}_q$ for each $(k,j) \in \mathcal{P}_{\ell,q}$.
Retrieve m_b by evaluating $\texttt{Reconstruct}_{\mathcal{S}'_b}((n_b^{(k)})_{k\in\mathcal{P}_\ell})$.
If the reconstruction fails at any step, output $\perp$.
Otherwise, output the reconstructed message m_b.

Fig. 6. Our 1-out-of-q OT combiner π'_{OT} for a general access structure $\mathcal{A}$.

The next theorem states the correctness and security properties of our construction.

Theorem 7.1. *Let Σ be an $\mathbb{F}_q$-LSSS with adversary structure $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$. The OT combiner π'_{OT} defined in Figure 6 is perfectly secure against active $(\mathcal{A}, \mathcal{A}^*)$-adversaries.*

Proof. Consider the ideal $\mathbb{F}_q$-LSSS Σ' over $\mathcal{P}_\ell$ defined in Figure 6, and let $\mathcal{A}' \subseteq 2^{\mathcal{P}_\ell}$ be its adversary structure. By Theorem 4.1, this construction is perfectly secure against active $(\mathcal{A}', (\mathcal{A}')^*)$-adversaries. The correctness of the protocol in Figure 6 follows from the correctness of the construction in Figure 3, proved in Section 6.

We next prove security against $(\mathcal{A}, \mathcal{A}^*)$-adversaries. Note that each server S_i indexed by $i \in \mathcal{P}_n$ corresponds to the participants $I_i \subseteq \mathcal{P}_\ell$ of the scheme Σ'.

First note that, by construction of Σ', we have $A \in \mathcal{A}$ if and only if $A' = \cup_{i\in A} I_i \in \mathcal{A}'$. Hence, any adversary that corrupts Alice and a set $A \in \mathcal{A}$ in the extended multi-use construction, will have as many capabilities as an adversary corrupting Alice and the servers corresponding to A' in the single-use construction, since the interaction with a particular OT implementation can be thought of as the concatenation of the interactions of the different calls to it.

Next, suppose that an adversary corrupts Bob and a set $B \subseteq \mathcal{P}_n$ of servers. Analogously to the previous case, the adversary will have as many capabilities as an adversary corrupting Bob and the servers $B' = \cup_{i \in B} I_i$. Then B' is in $(\mathcal{A}')^*$, and so the protocol is secure against the active corruption of this adversary. □

Remark 7.2. If the normalized total share size of Σ is ℓ, Bob sends a total of $\ell \log q$ bits to servers in the choice phase. In the sending phase, Alice sends a total of $q^2 \ell \log q$ bits to the servers. In the transfer phase, servers send a total of $\ell \log q$ bits to Bob. Hence, the communication complexity is $(q^2 + q + 1)\ell \log q$.

Now we state a result from [16] that characterizes the pairs $(\mathcal{A}, \mathcal{B})$ of adversary structures for which perfectly secure 1-out-of-2 OT combiners are known to be impossible to attain. It is used in 7.4.

Proposition 7.3 ([16]). *If $(\mathcal{A}, \mathcal{B})$ is not an $\mathcal{R}_2$ pair of adversary structures, then perfectly secure 1-out-of-2 OT combiners against active $(\mathcal{A}, \mathcal{B})$-adversaries cannot exist.*

Corollary 7.4. *Let $(\mathcal{A}, \mathcal{B})$ be a pair adversary structures. There exist perfectly secure 1-out-of-q OT combiners against active $(\mathcal{A}, \mathcal{B})$-adversaries if and only if $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$.*

Proof. Suppose that $(\mathcal{A}, \mathcal{B})$ is $\mathcal{R}_2$. By [31], $\mathcal{A}$ admits an $\mathbb{F}_q$-LSSS Σ. By Theorem 7.1, Σ provides a secure OT combiner for $(\mathcal{A}, \mathcal{B})$. Now suppose, for the sake of contradiction, that $(\mathcal{A}, \mathcal{B})$ is not $\mathcal{R}_2$. If there exists a secure 1-out-of-q combiner for $(\mathcal{A}, \mathcal{B})$, then there exists a secure 1-out-of-2 combiner for $(\mathcal{A}, \mathcal{B})$, which contradicts Proposition 7.3. □

8 One-out-of-q OT-combiners from 1-out-of-2 OT-combiners

We dedicate this section to construct 1-out-of-q OT combiners from 1-out-of-2 OT combiners. For this purpose, we adapt a technique to construct a 1-out-of-q OT instance from 1-out-of-q OT instances that presented by Crépeau, Brassard and Robert [18]. As far as we know, this approach has not been studied in any previous works. As we will see, for certain adversary structures, it is possible to get better communication complexity than using the protocols described above. However, the construction is inherently multi-use, requires multiple calls to 1-out-of-2 OT instances. The suitability of this construction with respect the ones presented above will depend on the efficiency of $\mathbb{F}_2$-linear secret sharing schemes for the adversary structure with respect $\mathbb{F}_q$-linear secret sharing schemes, the efficiency of the OT instances involved, and the length of the messages.

We describe the 1-out-of-q OT combiner π'_{OT} for bit messages in Figure 7. For messages of larger bitsize $s > 1$, we can extend this construction using

1-out-of-q OT Combiner π'_{OT} from a 1-out-of-2 OT Combiner π_{OT}

$\pi'_{OT}.\mathsf{Choose}(b)$**:** Given $b \in \mathbb{F}_q$, let $(b_0, \ldots, b_{q-2})$ be a sequence of bits defined as

$$b_k = \begin{cases} 1 & \text{if } k < b \\ 0, & \text{otherwise.} \end{cases}$$

For $k = 0, \ldots, q-2$, let

$$(b_1^{(k)}, \ldots, b_n^{(k)}) = \pi_{OT}.\mathsf{Choose}(b_k)).$$

Output $\left((b_i^{(0)}, \ldots, b_i^{(q-2)})\right)_{i \in \mathcal{P}_n}$.

$\pi'_{OT}.\mathsf{Send}(m_0, \ldots, m_{q-1})$**:** Sample $q-2$ bits $r_1, \ldots, r_{q-2}$ uniformly at random. Then, compute the values $(w_0, \ldots, w_{q-2})$ as

$$w_k = \begin{cases} \pi_{OT}.\mathsf{Send}(m_0, r_1) & \text{if } k = 0, \\ \pi_{OT}.\mathsf{Send}(m_k \oplus r_k, r_{k+1} \oplus r_k) & \text{if } 1 \le k < q-2, \\ \pi_{OT}.\mathsf{Send}(m_{q-2} \oplus r_{q-2}, m_{q-1} \oplus r_{q-2}) & \text{if } k = q-2. \end{cases}$$

Parse $w_k = (w_i^{j,(k)})_{(i,j) \in \mathcal{P}_{n,2}}$.
For every $i = 1, \ldots, n$, let $u_i = (w_i^{j,(k)})_{(k,j) \in \mathcal{P}_{q-1,2}}$.
Output $(u_1, \ldots, u_n)$.

$\pi'_{OT}.\mathsf{Rec}(b, (v_1, \ldots, v_n))$**:** For every $i = 1, \ldots, n$, parse $v_i = (v_i^0, \ldots, v_i^{q-2})$.
For every $k < b$, let $y_k = \pi_{OT}.\mathsf{Rec}(1, (v_1^k, \ldots, v_n^k))$.
If $b < q-1$, let $y_b = \pi_{OT}.\mathsf{Rec}(0, (v_1^b, \ldots, v_n^b))$.
If the reconstruction fails at any step, output $\perp$.
Otherwise, if $b < q-1$, output $\bigoplus_{k=0}^{b} y_k$. Else, if $b = q-1$, output $\bigoplus_{k=0}^{q-2} y_k$.

Fig. 7. 1-out-of-q OT combiner π'_{OT} from an 1-out-of-2 OT combiner π_{OT}.

zigzag functions [18], which requires $3^{\lceil \log_2 s \rceil}$ executions of this bit-OT protocol. In the figure, we call π_{OT} the 1-out-of-2 n-server bit-OT combiner. For this construction, we can instantiate π_{OT} with the general 1-out-of-q OT combiner presented in Figure 6 for the case $q = 2$, which is equivalent to the one presented in [15] for non-ideal schemes.

One advantage of this construction is that it only uses 1-out-of-2 OT protocols. As mentioned above, it results in a highly multi-use solution, as in general it requires a multi-use combiner for each of the 1-out-of-2 OT combiners, and the multiple calls to it required by the 1-out-of-q combiner technique of [18].

Next, we evaluate the communication cost of this protocol for messages of one bit and general adversary structures. We instantiate it with the 1-out-of-2 OT combiner from Figure 6). Suppose that the pair $(\mathcal{A}, \mathcal{B})$ requires a $\mathbb{F}_2$-LSSS Σ with normalized share size ℓ. In the Choose phase, the output of each $\pi_{OT}.\mathsf{Choose}$ has ℓ bits, and the total is $(q-1)\ell$ bits. In the Sending phase, the output of each $\pi_{OT}.\mathsf{Send}$ is 2ℓ bits, so it outputs $2(q-1)\ell$ bits in total. In the Transfer phase, servers send ℓ bits, in total. Therefore, the communication cost is $(3q-1)\ell$ bits. There are ℓ executions of bit-OT candidates, in total.

In order to compare it with the other construction of this work, we consider the threshold t adversary structure $\mathcal{A}$, for $1 < t < n$, and the pair $(\mathcal{A}, \mathcal{A}^*)$. This adversary pair requires a scheme with normalized share size $\ell = n \log n$. The resulting protocol exchanges $(3q-1)n \log n$ bits, and it uses each of the n 1-out-of-2 bit-OT candidates $q \log n$ times. For messages in $\mathbb{F}_q$, the use of zigzag functions implies increasing the communication cost and the OT calls by a factor of the order of $\log q$.

Acknowledgments

This article is supported by the Government of Catalonia (grant 2017 SGR 705), by the Spanish Government (project RTI2018-095094-B-C21, "CONSENT").

References

1. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer (2001)
2. Applebaum, B., Beimel, A., Nir, O., Peter, N.: Better secret sharing via robust conditional disclosure of secrets. In: STOC 2020. pp. 280–293 (2020)
3. Applebaum, B., Beimel, A., Farràs, O., Nir, O., Peter, N.: Secret-sharing schemes for general and uniform access structures. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 441–471. Springer (2019)
4. Beimel, A., Farràs, O.: The share size of secret-sharing schemes for almost all access structures and graphs. In: TCC 2020. LNCS, vol. 12552, pp. 499–529. Springer (2020)
5. Beimel, A.: Secret-sharing schemes: A survey. In: International Conference on Coding and Cryptology - IWCC'11. p. 11–46. Springer-Verlag (2011)
6. Beimel, A., Farràs, O., Mintz, Y.: Secret-sharing schemes for very dense graphs. Journal of Cryptology **29**(2), 336–362 (2016)
7. Beimel, A., Ishai, Y.: On the power of nonlinear secret-sharing. SIAM Journal on Discrete Mathematics **19**(1), 258–280 (2005)
8. Beimel, A., Othman, H., Peter, N.: Degree-2 secret sharing and conditional disclosure of secrets. IACR Cryptol. ePrint Arch. **2021**, 285 (2021), `https://eprint.iacr.org/2021/285`
9. Beimel, A., Weinreb, E.: Separating the power of monotone span programs over different fields. SIAM J. Comput. **34**(5), 1196–1215 (2005)
10. Bellare, M., Micali, S.: Non-interactive oblivious transfer and applications. In: Brassard, G. (ed.) CRYPTO' 89. LNCS, vol. 435, pp. 547–557. Springer (1990)
11. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC '88. pp. 1–10. ACM (1988)
12. Blundo, C., De Santis, A., Di-Crescenzo, G., Giorgio Gaggia, A., Vaccaro, U.: Multi-secret sharing schemes. In: CRYPTO'94. LNCS, vol. 839, pp. 150–163 (1994)
13. Cachin, C., Crépeau, C., Marcil, J.: Oblivious transfer with a memory-bounded receiver. In: FOCS 1998. pp. 493–502 (1998)

14. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067 (2000), `https://eprint.iacr.org/2000/067`
15. Cascudo, I., Damgård, I., Farràs, O., Ranellucci, S.: Server-aided two-party computation with minimal connectivity in the simultaneous corruption model. Cryptology ePrint Archive, Report 2014/809 (2014), `https://eprint.iacr.org/2014/809`
16. Cascudo, I., Damgård, I., Farràs, O., Ranellucci, S.: Resource-efficient OT combiners with active security. In: TCC 2017. LNCS, vol. 10678, pp. 461–486. Springer (2017)
17. Chor, B., Kushilevitz, E.: A zero-one law for boolean privacy. In: STOC '89. pp. 62–72. ACM (1989)
18. Crépeau, C., Brassard, G., Robert, J.M.: Information theoretic reductions among disclosure problems. In: SFCS 1986. pp. 168–173 (1986)
19. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: FOCS'88. pp. 42–52 (1988)
20. Csirmaz, L., Ligeti, P., Tardos, G.: Erdös-pyber theorem for hypergraphs and secret sharing. Graphs and Combinatorics (2014)
21. Dowsley, R., van de Graaf, J., Müller-Quade, J., Nascimento, A.C.A.: Oblivious transfer based on the McEliece assumptions. In: Safavi-Naini, R. (ed.) Information Theoretic Security. pp. 107–117. Springer (2008)
22. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Communications of the ACM **28**(6), 637–647 (1985)
23. Farràs, O., Kaced, T., Martín, S., Padró, C.: Improving the linear programming technique in the search for lower bounds in secret sharing. IEEE Trans. on Inf. Theory **66**(11), 7088–7100 (2020)
24. Farràs, O., Ribes-González, J.: One-out-of-q ot combiners. Cryptology ePrint Archive, Report 2021/822 (2021), `https://eprint.iacr.org/2021/822`
25. Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. J. Comput. Syst. Sci. **60**(3), 592–629 (2000)
26. Goyal, V., Ishai, Y., Sahai, A., Venkatesan, R., Wadia, A.: Founding cryptography on tamper-proof hardware tokens. IACR Cryptology ePrint Archive **2010**, 153 (2010)
27. Harnik, D., Ishai, Y., Kushilevitz, E., Buus Nielsen, J.: OT-combiners via secure computation. In: Canetti, R. (ed.) TCC 2008. pp. 393–411. LNCS, Springer (2008)
28. Harnik, D., Kilian, J., Naor, M., Reingold, O., Rosen, A.: On robust combiners for oblivious transfer and other primitives. In: Cramer, R. (ed.) EUROCRYPT 2005. pp. 96–113. LNCS, Springer (2005)
29. Ishai, Y., Maji, H.K., Sahai, A., Wullschleger, J.: Single-use OT combiners with near-optimal resilience. In: ISIT. pp. 1544–1548. IEEE (2014)
30. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: CRYPTO 2008. pp. 572–591. LNCS, Springer-Verlag (2008)
31. Ito, M., Saito, A., Nishizeki, T.: Secret sharing scheme realizing general access structure. Electronics and Communications in Japan, Part III **72**(9), 56–64 (1989)
32. Jackson, W., Martin, K.M., O'Keefe, C.M.: Multisecret threshold schemes. In: CRYPTO'93. LNCS, vol. 773, pp. 126–135 (1994)
33. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC '88. pp. 20–31. ACM (1988)
34. Kilian, J., Micali, S., Ostrovsky, R.: Minimum resource zero-knowledge proofs. In: Brassard, G. (ed.) CRYPTO'89. pp. 545–546. Springer (1990)
35. Lindell, Y.: How to simulate it - a tutorial on the simulation proof technique. In: Tutorials on the Foundations of Cryptography (2016)

36. Liu, T., Vaikuntanathan, V.: Breaking the circuit-size barrier in secret sharing. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) STOC 2018. pp. 699–708. ACM (2018)
37. Liu, T., Vaikuntanathan, V., Wee, H.: Towards breaking the exponential barrier for general secret sharing. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 567–596. Springer (2018)
38. Meier, R., Przydatek, B.: On robust combiners for private information retrieval and other primitives. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 555–569. Springer (2006)
39. Meier, R., Przydatek, B., Wullschleger, J.: Robuster combiners for oblivious transfer. In: Vadhan, S.P. (ed.) TCC 2007. pp. 404–418. LNCS, Springer (2007)
40. Naor, M., Pinkas, B.: Computationally secure oblivious transfer. Journal of Cryptology **18**(1), 1–35 (2005)
41. Padró, C.: Lecture notes in secret sharing. IACR Cryptology ePrint Archive p. 674 (2012), `http://eprint.iacr.org/2012/674`
42. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. pp. 554–571. Springer (2008)
43. Przydatek, B., Wullschleger, J.: Error-tolerant combiners for oblivious primitives. In: Automata, Languages and Programming. pp. 461–472. Springer (2008)
44. Rabin, M.O.: How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive (2005), `http://eprint.iacr.org/2005/187`
45. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
46. Vaikuntanathan, V., Nalini Vasudevan, P.: Secret sharing and statistical zero knowledge. In: ASIACRYPT 2015. pp. 656–680. LNCS, Springer-Verlag, Inc. (2015)
47. Yao, A.C.C.: Protocols for secure computations. In: SFCS '82. pp. 160–164. IEEE Computer Society (1982)

A Examples of W-OT-Compatible Access Structures

We here give examples of W-OT-compatible access structures.

Example A.1. Consider the Γ_W access structure defined as follows. Let $n = q = 3$. Then,

$$P_1 = \{(1,0),(1,1),(1,2)\}, P_2 = \{(2,0),(2,1),(2,2)\}, P_3 = \{(3,0),(3,1),(3,2)\}$$

$$\mathcal{P}_{3,3} = P_1 \cup P_2 \cup P_3$$

Let $W = \langle (0,1,2),(1,0,2) \rangle \subseteq \mathbb{F}_3^3$. The vector subspace W has 9 vectors, and so Γ_W has 9 minimal authorized subsets. It can be checked that Γ_W is not a matroid port, and so it does not admit any ideal linear secret sharing scheme [23].

Example A.2. Let $n = q = 3$ as in the previous example. Then,

$$\Delta = \{A \subseteq \mathcal{P}_{3,3} : |A \cap P_i| = 0, 1 \text{ or } 3 \text{ for } i = 1,2,3\}.$$

Note that $|\Delta| = \left(\binom{3}{0} + \binom{3}{1} + \binom{3}{3}\right)^3 = 125$. Let $W \subseteq \mathbb{F}_3^3$ be the affine subspace defined by $W = \mathbf{k} + V$, where

$$\mathbf{k} = (1,1,1) \quad \text{and} \quad V = \langle (1,0,2) \rangle_{\mathbb{F}_3} = \{(0,0,0),(1,0,2),(2,0,1)\}\,,$$

so $W = \{(1,1,1),(2,1,0),(0,1,2)\}$. The access structure $\mathit{\Gamma}_W$ on $\mathcal{P}_{3,3}$ is defined by the minimal access structure

$$\min \mathit{\Gamma}_W = \{\{(1,1),(2,1),(3,1)\},\{(1,2),(2,1),(3,0)\},\{(1,0),(2,1),(3,2)\}\}\,.$$

We note that $\mathit{\Gamma}_W$ is trivially W-OT-compatible, and that any W-OT-compatible access structure $\mathit{\Gamma}$ satisfies $\min \mathit{\Gamma}_W \subseteq \min \mathit{\Gamma}$. Observe that the access structure $\mathit{\Gamma}$ with

$$\min \mathit{\Gamma} = \min \mathit{\Gamma}_W \cup \{(1,1),(2,1),(3,0),(3,2)\}$$

is also W-OT-compatible. This is because, for any set $A \in \mathit{\Gamma} \cap \mathit{\Delta}$ containing $\{(1,1),(2,1),(3,0),(3,2)\}$, we have that $(1,1) \in A \cap P_1$, that $(2,1) \in A \cap P_2$ and $A \cap P_3 = P_3$. Hence, A contains $\{(1,1),(2,1),(3,1)\}$, and so $\mathit{\Gamma} \cap \mathit{\Delta} \subseteq \mathit{\Gamma}_W \cap \mathit{\Delta}$.