Universitat Oberta de Catalunya

Master's Thesis

One-out-of-q OT Combiners

Author:
Jordi Ribes-González

Supervisor:
Dr. Oriol Farràs

June 11, 2019

UNIVERSITAT OBERTA DE CATALUNYA

Abstract

Master's Degree in Security of the Information and Communication Technologies

One-out-of-q OT Combiners

by Jordi Ribes-González

Oblivious Transfer (OT) protocols run between two users, a sender Alice and a receiver Bob. They allow Alice, who holds some information, to send partial knowledge on that information to Bob while being oblivious to what was actually transferred. Moreover, the receiver does not have access to more information than intended as per the protocol.

In the flavor of OT protocols considered in this work, called 1*-out-of-q OT*, Alice holds a set of $q \geq 2$ messages. After the execution of the OT protocol, Alice wants Bob to receive a message of his choice. The security of OT protocols guarantees that Bob learns only one of Alice's messages, and that Alice is oblivious to which message was received by Bob after the execution of the protocol.

Oblivious Transfer protocols are of great importance in cryptography. Their relevance lies in their use as fundamental cryptographic primitives, as they have been employed to realize several useful cryptographic constructions.

Regarding security, perfectly secure OT protocols have been proved impossible to achieve. Hence, to build OT protocols, one must rely on a variety of security assumptions. To guard against the event that these assumptions are broken, the standard method is to ground the security of OT protocols in various assumptions at once. This motivates the introduction of OT combiners.

Oblivious Transfer combiners take as input a set of implementations of OT, and they produce a single OT protocol that is secure as long as sufficiently many of the original OT implementations are secure in the first place. Thus, when using OT combiners, the security of the combined OT protocol holds even in the case that a small set of OT implementations are insecure.

The literature on OT combiners deals extensively with the case where Alice holds $q = 2$ messages. In this master's thesis, we present an OT combiner that extends previous 1-out-of-2 constructions to the 1-out-of-q case for an arbitrary prime integer $q \geq 2$, and that is black-box and single-use. In the process, we study secret sharing schemes for particular families of adversary structures, and we provide secret sharing schemes and linear algebra results that are of independent interest.

We prove that our construction achieves a very strong notion of security called *perfect security against active* $(\mathcal{A}, \mathcal{B})$*-adversaries.* For large enough values of q, our construction is proved secure against a larger class of adversaries than in previous works. Furthermore, in the case that q is at least the number of used OT candidates, our construction is secure against adversaries that corrupt less than half the OT candidates. This improves on the previously studied $q = 2$ case.

Contents

To my family and friends.

Chapter 1

Introduction

In this chapter, we give a brief introduction to Oblivious Transfer (OT) protocols and OT combiners, providing an overview of the related literature. We also sketch the aims and results of this thesis.

This chapter is divided in four sections. In Section 1.1 we offer a high-level introduction of OT, which is the cryptographic primitive our work is directed at. Next, in Section 1.2 we introduce our main object of study, namely OT Combiners, and we survey the related literature in Section 1.3. Finally, we wrap up the chapter in Section 1.4, by commenting on the research directions we explore and by putting our work in the context of the existing research results.

1.1 Oblivious Transfer

Oblivious Transfer (OT) protocols were first introduced by Rabin [1] in 1981. Oblivious transfer protocols involve two parties, a *sender* and a *receiver*, which we respectively name Alice and Bob throughout all this work. 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.

The first example of an OT functionality, realized in the first OT protocol by Rabin [1] (described in Section 2.2.1), starts with Alice holding a single message. After the execution of the protocol, Bob learns this message with probability 1/2, and Alice is oblivious to whether or not Bob received it. Another flavor of OT is 1-*out-of*-2 *OT* [2] (see Section 2.2.2), in which the sender holds two messages and where the receiver chooses to receive one of the two messages from the sender. The security guarantees here state that the sender is oblivious to the message that was actually transferred to the receiver, and that the receiver gets information on one of the messages only. The type of OT that we study here is called 1-*out-of*-q *OT* (see Section 2.2.3). It is a generalization 1-out-of-2 OT that lets the sender hold $q \geq 2$ messages instead of just two, and which allows the receiver to fetch only one of these messages.

The relevance of OT protocols in cryptography lies in their role as a fundamental primitive in many cryptographic constructions. To put this into context, we give a brief account of the main functionalities OT has found an application to: Secure Multi-Party Computation (MPC), Zero-Knowledge Proofs (ZKP) and Bit Commitment (BC) schemes. We also comment on the related fields of Private Information Retrieval (PIR) and Oblivious Linear Function Evaluation (OLFE).

Secure Multi-Party Computation is an area of cryptography that considers a scenario with two or more parties. In this scenario, every party holds some private input data, and all of them wish to compute some function of their joint inputs. The aim of

MPC protocols is to enable them to jointly evaluate this function without revealing their inputs to the other parties. As an example application, one could imagine an e-voting scenario where electors hold their private ballot, and where the majority vote should be computed while keeping the ballots private.

The relation between MPC and OT was first shown by Rabin in [3], where he described a solution for MPC in the two-party setting that uses Rabin's OT protocol [1] as a primitive. Then, in [4] Kilian proved that 1-out-of-2 OT is *complete* for MPC. In other words, he proved that using OT as a primitive suffices to securely compute any polynomial-time computable function among any number of parties.

Zero-Knowledge Proof protocols involve two parties, respectively called the *prover* and the *verifier*. At the beginning of the protocol, the prover holds some private knowledge, which can consist of an integer value, a string, or a solution to some mathematical problem. A ZKP protocol allows the prover to attest to the verifier that she has this private knowledge, but without revealing it to the verifier. Moreover, ZKP protocols guarantee that the verifier is always convinced that the prover holds the private knowledge if she actually does, and that no cheating prover can convince the verifier that she holds the private knowledge in case she does not.

In [4], Kilian showed a method to build ZKP protocols by using exclusively 1-out-of-2 OT. Then, in [5], Kilian, Micali and Ostrovsky provided more efficient ZKP protocols that used OT only in a short pre-processing phase. Non-Interactive ZKP (NIZKs) can also be realized by using exclusively OT, and Bellare and Micali [6] show how to build NIZKs from non-interactive 1-out-of-2 OT (such as [7], see Figure 2.5 in Section 2.2.3).

A Bit Commitment Scheme consists of a pair of protocols executed by two parties, a *commiter* and a *receiver*. In the first protocol the commiter chooses a bit b, and she sends a *commitment* to this bit to the receiver. In the second protocol, which is executed at a later stage, the commiter reveals a bit b' to the receiver, and the receiver is able to verify whether $b = b'$ or not. The essential properties of BC schemes are the hiding and the binding property. The *hiding* property states that the commitment does not reveal any information about b to the receiver right after the execution of the first protocol. The *binding* property states that the commitment is bound to b, so that the commiter can not choose $b' \neq b$ and convince the receiver that $b = b'$ in the second protocol execution.

In [4], Kilian showed how to build BC schemes using exclusively 1-out-of-2 OT. We note that BC schemes, as OT protocols, are fundamental primitives that serve as building blocks for many other cryptographic constructions, such as two-party MPC protocols.

Another cryptographic construction that is closely related to OT is Private Information Retrieval (PIR) [8]. As in 1-out-of-q OT, in PIR a sender holds q messages, and a receiver may want to retrieve one (or possibly more) of them without letting the sender know which messages she received. However, unlike in OT, PIR does not introduce any privacy requirement on the sender, so the receiver could potentially learn arbitrary information on the messages held by the sender. In addition, the amount of information sent to the receiver should be significantly less than the whole set of messages. This rules out the trivial case where the sender just transmits all information to the receiver.

A generalization of OT, which was proposed by Naor and Pinkas in [9], is called Oblivious Polynomial Evaluation. As in OT, an OPE protocol involves two parties, a sender and a receiver. The sender chooses a polynomial function f over $\mathbb{F}_q$, and the receiver chooses a value $x \in \mathbb{F}_q$. At the end of the protocol, the receiver learns $f(x)$ and no other information about f, and the sender does not learn any information on

x. In the case that the function f is linear, OPE is called Oblivious Linear Function Evaluation (OLFE). In OLFE, the sender takes as input $a, b \in \mathbb{F}_q$, and Bob receives the value $ax + b$ as output. We note that OLFE is equivalent to 1-out-of-2 OT [10].

1.2 OT Combiners

Our work revolves around a central observation: perfectly secure OT protocols are impossible to achieve. This is so because, using the reduction from OT to MPC by Kilian [4], they would yield unconditionally-secure two-party computation, which is impossible to obtain for some functions (see [11, 12]). Hence, the security of OT protocols is necessarily conditional, and so OT protocols can only be built by imposing assumptions on security. These assumptions come in a variety of flavors, such as using hardware tokens [13], assuming the existence of a noisy channel between both parties [14], or restricting the storage [15] or the computational capabilities of the parties. In relation to this last assumption, there exist many computational hardness assumptions one can base OT protocols on, such as the hardness of RSA [1], the Decisional Diffie-Hellman assumption [6, 7], the assumptions used in the McEliece encryption scheme [16] and also some worst-case lattice assumptions [17].

While falling back to conditional security is a necessary step in order to build secure OT protocols, it implies that the security guarantees of OT can potentially be compromised. For example, at some point a hardware token may become corrupted, or a computational assumption may break due to the development of new cryptographic attacks. The standard method to mitigate this concern consists in grounding security on various assumptions at once, by simultaneously using several implementations. This motivates the introduction of *combiners* of OT candidates, or simply *OT combiners*.

On a more general note, suppose that we have at our disposal several implementations of a cryptographic functionality. We can think, for example, of a set of encryption schemes, one-way functions, or OT protocols. The notion of combiners consists of finding a way to blend all these implementations into a single one, so that the resulting combination is secure even if some of the original implementations are insecure. Combiners have been previously studied in many areas of cryptography. For example, the concept of combiners is applied in the familiar context of multi-factor authentication, where many authentication methods are used concurrently, as well as in cascading of block ciphers. Also, combiners for many cryptographic functionalities have been studied in the literature, for instance combiners of encryption schemes [18, 19], of PRGs [20], of hash functions [21] and, of course, OT combiners.

The study of OT combiners was initiated Harnik, Kilian, Naor, Reingold and Rosen [22] in 2005, and they were further studied in other articles [22–28]. Using an OT combiner, a set of n candidate implementations of OT can be merged to realize a single OT protocol, in such a way that the final protocol is secure as long as sufficiently many of the initial implementations were secure to start with. 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. Moreover, the resulting protocol stays secure even if the security of some of the OT candidates is flawed.

1.3 Related Work

The first OT combiners were presented by Harnik, Kilian, Naor, Reingold and Rosen in 2005 [22]. They defined 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 candidate implementation 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, and they prove the tightness of this bound. In particular, they show that such OT combiners cannot exist for $n = 2, t = 1$, and they build an OT combiner for $n = 3, t = 1$. They also introduce a second solution for the active (i.e. malicious) adversary model, but this variant has some efficiency and security flaws (e.g. see [29][Section 5.4]).

Meier, Przydatek and Wullschleger defined in [24] the notion of (n, δ)-*uniform OT combiner*. Such OT combiners implement the 1-out-of-2 OT functionality, and they 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 $t_A + t_B < n$. Their solution requires the roles of the sender and the receiver to be reversed during the protocol execution, and the corresponding combiner makes two calls to each OT candidate.

Later, Przydatek and Wullschleger [30] considered combiners that take a set of n OLFE candidate implementations and produce a 1-out-of-2 OT protocol. Their solution is also unconditionally secure for $t_A + t_B < n$. However, it requires the size of the message space to be greater than the number n of candidate implementations of OLFE to combine. Interestingly, we also consider this restriction in the analysis of our results (see Chapter 4).

In [25], Harnik, Ishai, Kusilevitz and Nielsen presented the first *single-use* OT combiner, meaning that only one black-box call is made to each of the n OT implementations per protocol execution. They study (n, t_A, t_B)-*OT combiners*, which are secure against passive adversaries that corrupt either Alice and t_A OT candidates, or Bob and t_B OT candidates. A statistically secure (n, t, t)-OT combiner is provided 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. They also provide constant production rate, meaning that the number of secure OT protocols produced is not just one, but a constant fraction $\Theta(n)$ of the number n of OT candidates.

Additionally, [25] gives a computationally secure OT combiner against active adversaries. Subsequently, Ishai, Prabhakaran and Sahai [29] show that this construction can be turned into an (n, t, t)-OT combiner that is statistically secure against active adversaries for $t = \Omega(n)$, while leaving unconditional security as an open problem.

Ishai, Maji, Sahai and Wullschleger present in [26] a single-use (n, t, t)-OT combiner in the 1-out-of-2 scenario. Their solution achieves statistical security against passive adversaries for $t = n/2 - \omega(\log \kappa)$, where κ is the security parameter.

Another variant of combiners for OT is that of *cross-primitive combiners*, studied by Meier and Przydatek in [23]. As in [30], here the combiner implements a different functionality than the candidates. They present a $(2, 1)$-*PIR-to-OT combiner*, which takes two Private Information Retrieval (PIR) schemes and produces a 1-out-of-2 OT protocol that is unconditionally secure for the sender, provided one of the two PIR schemes is also secure. This result comes in contrast with the impossibility result of [22]. Their construction only guarantees the privacy of Alice against a honest-but-curious adversary corrupting Bob and one of the two servers.

Following [26], Cascudo, Damgard, Farràs and Ranellucci [27, 28] achieve single-use 1-out-of-2 OT combiners, and their solution produces a single OT instance (see

Section 2.4). As for security, they are the first to provide perfect (unconditional, zero-error) security against active adversaries. They generalize the security notion of Harnik et al. [25] by defining the notion of *perfect security against active* $(\mathcal{A}, \mathcal{B})$-*adversaries*, which we also adopt in this thesis. This definition considers a malicious 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.

Our work requires the development of secret sharing and coding theory techniques that are of independent interest. In that regard, the works by Vaikuntanathan and Vasudevan [31] and by Beimel and Ishai [32] are related to ours, since some of the authorized sets of their studied secret sharing schemes (see Section 2.3) are also authorized in our construction. More concretely, given a language $L \in \{0,1\}^n$, they consider secret sharing schemes for $2n$ participants for which the sets $\{(1, v_1), \dots, (n, v_n)\}$ are minimally authorized for every $(v_1, \dots, v_n) \in L$. However, on top of those sets, they also consider minimally authorized sets of the form $\{(i, 0), (i, 1)\}$.

1.4 Our Work

The main references of this thesis are the works by Cascudo, Damgard, Farràs and Ranellucci [27, 28]. Our work can be seen as an extension of the OT combiner in [28], which in turn is based on the scheme by Ishai, Maji, Sahai and Wullschleger [26]. As in [27, 28], we prove that our OT combiners satisfy the notion of perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries.

In our work we present a 1-*out-of-q OT combiner* that extends previous 1-out-of-2 OT combiners to the 1-out-of-q case. In our setting, the underlying OT candidates and the produced OT protocol take q messages $m_0, \dots, m_{q-1}$ from Alice and an element $b \in \mathbb{F}_q$ from Bob, and they output the message m_b to Bob. All previous works in OT combiners have studied exclusively the 1-out-of-2 case, where the underlying OT candidates and the produced OT protocol take just two bit messages m_0, m_1 (or messages in $\mathbb{F}_{2^m}, \mathbb{F}_q$ in the case of [24, 30], respectively) from Alice and output one of them to Bob. Hence, our work studies OT combiners in a more general setting, and it allows to harness the combined security of 1-out-of-q OT protocols such as [7, 33–36].

Our OT combiner provides a security advantage with respect to the constructions of [27, 28]. Indeed, extending [28] to the general 1-out-of-q case has the effect of improving the security against adversaries that corrupt the sender, by resisting against the corruption of more OT candidates than in the $q = 2$ case. See Chapter 4 for more details.

As in [27, 28], we view OT combiners as *server-aided* OT protocols. This means that each of the n OT candidates is modeled as a server that implements the OT functionality, i.e. that receives q messages $m_0, \dots, m_{q-1} \in \mathbb{F}_q$ from Alice and an element $b \in \mathbb{F}_q$ from Bob, and outputs the message m_b to Bob. Hence, in the rest of the thesis we adopt this convention and refer to each of the n OT candidates as a *server*. In practice, a complete server transaction must be thought of as an OT protocol execution between Alice and Bob.

In the process of building our 1-out-of-q OT combiner, we study secret sharing schemes associated to affine spaces. Concretely, we study secret sharing schemes on n participants, for which the sets $\{(1, v_1), \dots, (n, v_n)\}$ are authorized for every $(v_1, \dots, v_n)$ in some affine subspace of $\mathbb{F}_q^n$. Hence, we also contribute to the study of secret sharing schemes, by studying the problem of building efficient schemes for these access structures.

1.5 Outline of the Thesis

This work is organized in four chapters. In Chapter 1, we have given a brief introduction to Oblivious Transfer (OT) protocols and to OT combiners, providing an overview of the related literature and of the various extensions and applications of OT. We have also stated the aims of this thesis, and we have described the previous results our work is based on.

In Chapter 2, we lay out the background theory and tools needed in the rest of this thesis. Here, we introduce Secret Sharing, a fundamental cryptographic primitive that is essential to our construction. We also give an account of OT, along with some examples and applications. Finally, we introduce OT combiners, which are the main objects of study in this work. We define OT combiners, we give examples, and we state the correctness and security definitions used to assess the properties of our construction.

Chapter 3 presents our 1-out-of-q OT combiner, which can be seen as an extension of the OT combiner in [28] to the 1-out-of-q scenario. Proofs of the consistency and of the security of our construction are also provided at the end of the chapter.

Finally, in Chapter 4 we conclude the thesis by commenting on the achieved results, and by stating some research lines following this work.

Chapter 2

Preliminaries

In this chapter, we lay out the background theory needed in the rest of the thesis. We divide this chapter in four sections. In Section 2.1, we introduce some basic definitions and notation used throughout this work. Section 2.2 presents the Oblivious Transfer (OT) primitive, along with some examples and applications. Next, in Section 2.3 we give an account of Secret Sharing, which is an essential primitive to our construction. Finally, in Section 2.4 we introduce OT combiners, defining them, giving examples and stating the correctness and security definitions used in this work.

2.1 Notation and Basic Definitions

All through this work, q denotes an arbitrary positive prime integer. We identify the set of representatives of the integer residue classes modulo $q-1$ with the set of non-negative integers smaller than q. Hence, by abuse of notation, we often denote $\mathbb{F}_q = \{0, \ldots, q-1\}$.

Given an integer $n \geq 2$, we denote by $\mathcal{P}_n$ the set of positive integers up to n, i.e. $\mathcal{P}_n := \{1, \ldots, n\}$. We also define

$$\mathcal{P}_{n,q} := \mathcal{P}_n \times \mathbb{F}_q = \{(i,j) \ : \ i \in \mathcal{P}_n,\, j \in \mathbb{F}_q\}\,.$$

We denote the power set of a set P by $2^P := \{A \ : \ A \subseteq P\}$.

In this work, we mainly deal with two-party protocols, and the aim of such protocols is to compute a certain functionality. The notion of functionality is formalized in the next definition, which is taken from [37].

Definition 1 ([37])**.** *A* functionality $\mathcal{F}$ *is a possibly random process* $\mathcal{F} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ *that takes a pair of inputs* $x, y \in \{0,1\}^*$ *and outputs a random variable* $(\mathcal{F}_1(x,y), \mathcal{F}_2(x,y))$.

We say that a protocol between two parties Alice and Bob *implements a functionality* $\mathcal{F}$ when, assuming Alice and Bob behave honestly and have input x and y respectively, at the end of the protocol Alice obtains $\mathcal{F}_1(x,y)$ and Bob obtains $\mathcal{F}_2(x,y)$.

2.2 Oblivious Transfer

Oblivious transfer protocols were introduced by Rabin [1] in 1981. An OT protocol runs between two parties, a sender Alice and a receiver Bob, which communicate without the help of any trusted third party and according to the protocol. Broadly, in OT protocols the sender owns some information, and she wants the receiver to have partial knowledge on that information while being oblivious to what was actually transferred. Moreover, the receiver should not have access to more information than intended as per the protocol.

This section is divided in three parts. In Section 2.2.1, we start by commenting on the first OT protocol by Rabin. Then, in Section 2.2.2, we describe 1-out-of-2 OT protocols, which are the ones studied by all the previous literature on OT combiners. Finally, in Section 2.2.3 we study 1-out-of-q OT, which is the cryptographic primitive our work focuses on.

We remark again that OT protocols are computationally secure by nature, since perfectly secure OT would directly yield unconditionally-secure two-party computation (see [11, 12]).

2.2.1 Rabin's OT protocol

As a first instance of OT, we describe the earliest OT protocol by Rabin [1] in Figure 2.1. In Rabin's OT protocol, the sender Alice holds a single message m. After the execution of the protocol, the receiver Bob learns this message m with probability $1/2$, and Alice is oblivious to whether or not Bob did receive the message. This protocol makes implicit use of the RSA encryption scheme [38], and its security against Bob rests on the hardness of RSA.

The Rabin OT Protocol

1. Alice chooses two large positive prime integers p, q at random and sets $n = pq$. She generates an exponent e relatively prime to $(p-1)(q-1)$, and computes a multiplicative inverse d of e in $\mathbb{Z}_{(p-1)(q-1)}$. She sends n and e to Bob.
2. Alice takes a nonzero message $m \in \mathbb{Z}_n$ and sends the RSA ciphertext m^e to Bob.
3. Bob picks a nonzero value $x \in \mathbb{Z}_n$ at random, computes $z = x^2$ in $\mathbb{Z}_n$, and sends z to Alice. Note that, with overwhelming probability, the square roots of z are $\{x, n-x, y, n-y\}$ for some $y \in \mathbb{Z}_n \setminus \{x, n-x\}$ (see [1] for details).
4. By using the factorization $n = pq$, Alice computes one of the four square roots t of z (see [1] for details). She sends t to Bob.
5. If Bob receives either $t = y$ or $t = n - y$, then he can factor $n = pq$ (since $\gcd(|x - t|, n)$ is either p or q). Using p and q, he raises the RSA ciphertext m^e to the secret key d to retrieve Alice's original message m. If Bob receives $t = x$ or $t = n - x$, it is computationally infeasible for him to retrieve the message m.

FIGURE 2.1: The Rabin OT Protocol π_{OT}.

The most relevant application of Rabin's OT protocol is Two-Party Computation (2PC). A 2PC protocol allows two users to jointly evaluate a polynomial time computable function on their private inputs. By agreeing on a particular function and by engaging in an elaborate protocol, 2PC allows them to learn the joint function evaluated on their inputs, while they learn nothing else about each other's inputs in the process. In 1982, Yao [3] described a solution for 2PC that uses Rabin's OT protocol as a primitive.

2.2.2 One-out-of-two OT

While Rabin's protocol is the original form of OT, an apparently more general flavor of OT has allowed many other implementations [2, 6, 14–17] and uses [4, 39]. This second formulation, provided by Even, Goldreich and Lempel [2] in 1985, is called

1-*out-of*-2 *OT*. In 1-out-of-2 OT protocols, the sender is in possession of two messages m_0, m_1, and the receiver wants to choose one of them by specifying a message index $b \in \mathbb{F}_2$. The corresponding functionality is illustrated in Figure 2.2.

We have to note that, while the OT functionality implemented by the Rabin OT protocol (fig. 2.1) is a particular case of the 1-out-of-2 OT functionality with bit messages (by letting Alice throw a fair coin $c \overset{\$}{\leftarrow} \{0,1\}$ and set $m_c = m$ and $m_{1-c} = \perp$), the two flavors are in fact equivalent as shown by Crépeau [40].

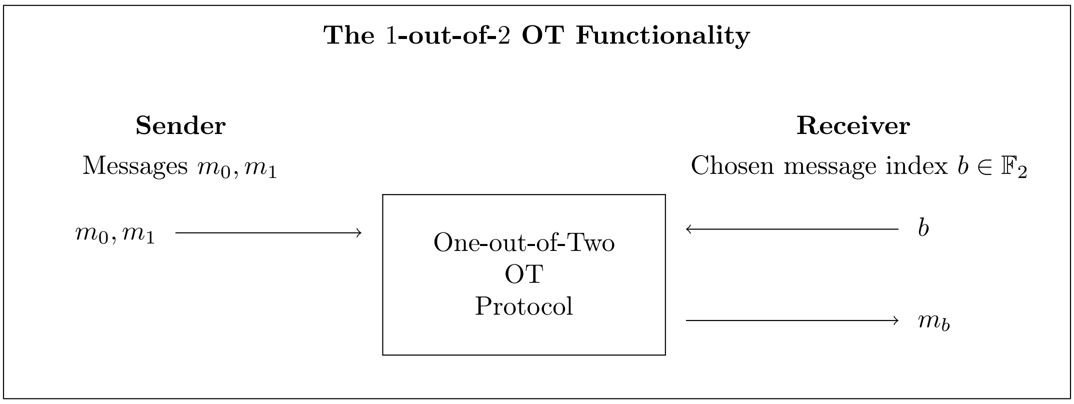
The 1-out-of-2 OT Functionality
Sender
Messages m_0, m_1
Receiver
Chosen message index $b \in \mathbb{F}_2$
m_0, m_1
One-out-of-Two OT Protocol
b
m_b

FIGURE 2.2: One-out-of-Two OT.

The security properties of 1-out-of-2 OT protocols are two-fold. On one hand, the security with respect to the receiver states that the sender must be oblivious to which message m_b was transferred to the receiver. On the other hand, the security with respect to the sender states that it is hard for the receiver to learn information on any message other than the one received after the execution of the protocol. Hence, if the receiver learns m_b after the execution of the protocol, then the message m_{1-b} must remain private to the sender.

The most relevant application of 1-out-of-2 OT protocols is Multi-Party Computation (MPC). Multi-party computation generalizes 2PC by allowing many users to jointly evaluate a function on their combined inputs. As in 2PC, users do not learn anything about each other's inputs in the process. In [4], Kilian proves that 1-out-of-2 OT is complete for MPC. That is, that using OT as a primitive suffices to securely compute any polynomial-time computable function among any number of parties.

Note that, given a protocol implementing 1-out-of-2 OT with bit messages, one can construct a 1-out-of-2 OT protocol supporting k-bit messages by invoking k runs of the original protocol and concatenating the obtained messages (see [41, 42]). Similarly, given a 1-out-of-2 OT protocol with message space $\mathcal{M}$, one can produce a 1-out-of-2 OT protocol with message space $\mathcal{M}^k$ by considering k runs of the same protocol. In this work, we often obviate the message space of OT protocols by making implicit use of this approach.

As an example, we describe the first 1-out-of-2 OT protocol by Even, Goldreich and Lempel [2] in Figure 2.3. The security of this protocol rests on the hardness of factoring the value $n = pq$ chosen by the sender Alice, and it makes an implicit use of the RSA encryption scheme [38].

The Even-Goldreich-Lempel OT Protocol

1. Alice chooses two large positive prime integers p, q at random and sets $n = pq$. She generates an exponent $e \in \mathbb{Z}^{\times}_{(p-1)(q-1)}$ (i.e., relatively prime to $(p-1)(q-1)$), and computes $d = e^{-1}$ (i.e., the multiplicative inverse d of e in $\mathbb{Z}_{(p-1)(q-1)}$). She sends n and e to Bob. Also, we assume she holds two messages $m_0, m_1 \in \mathbb{Z}_n$.
2. Alice generates two random nonces $x_0, x_1 \overset{\$}{\leftarrow} \mathbb{Z}_n$ and sends them to Bob.
3. Bob chooses a message index $b \in \mathbb{F}_2$ and picks a random value $k \overset{\$}{\leftarrow} \mathbb{Z}_n$ at random. Then, he computes $v = x_b + k^e$ in $\mathbb{Z}_n$, and sends v to Alice.
4. Alice computes the values
$$m'_0 = m_0 + (v - x_0)^d$$
$$m'_1 = m_1 + (v - x_1)^d$$
and sends them to Bob.
5. Knowing b and the previously chosen value $k \in \mathbb{Z}_n$, Bob recovers $m_b = m'_b - k$.

FIGURE 2.3: The Even-Goldreich-Lempel OT Protocol.

2.2.3 One-out-of-q OT

Here, we present the main functionality studied in this work, which is the 1-*out-of-q OT functionality.* This functionality was first presented by Crépeau, Brassard and Robert [42] in 1986, and it generalizes that of 1-out-of-2 OT by allowing Alice to hold multiple messages.

In the 1-out-of-q OT functionality, 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. That is, in the notation of Definition 1, the functionality $\mathcal{F}(x, y) = (\mathcal{F}_1(x, y), \mathcal{F}_2(x, y))$ implemented by 1-out-of-q OT protocols is described by

$$\begin{aligned} x &= (m_0, \ldots, m_{q-1}), & \mathcal{F}_1(x, y) &= \perp, \\ y &= b, & \mathcal{F}_2(x, y) &= m_y. \end{aligned}$$

where $\perp$ stands for the empty bit string. This functionality is illustrated in Figure 2.4.

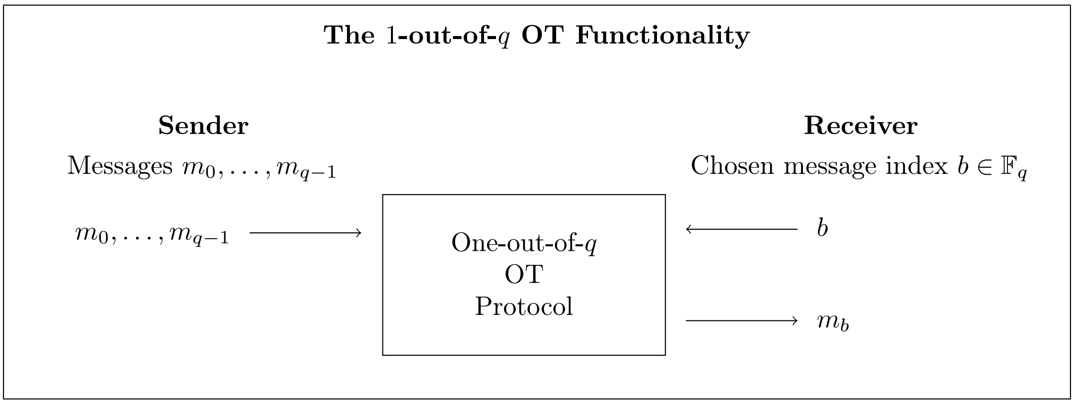
The 1-out-of-q OT Functionality
Sender
Messages $m_0, \ldots, m_{q-1}$
$m_0, \ldots, m_{q-1}$
Receiver
Chosen message index $b \in \mathbb{F}_q$
b
One-out-of-q OT Protocol
m_b

FIGURE 2.4: One-out-of-q OT.

This extension of OT allows other applications such as Private Set Intersection [35, 36], Private Information Retrieval [33] and Multi-Party Computation [39] (where 1-out-of-4 OT is necessary to securely evaluate arithmetic multiplication gates).

Note that, given a protocol implementing 1-out-of-q OT, it is possible to build a t-out-q OT protocol by invoking t runs of the original protocol [43]. In a t-out-q OT protocol, the receiver Bob recovers t messages out of the q that the sender Alice holds.

The original approach by Crépeau, Brassard and Robert [42] to build a 1-out-of-q OT protocol for bit messages consists in invoking a 1-out-of-2 OT protocol $q-1$ times. More concretely, the sender Alice, who holds bit messages $m_0, \ldots, m_{q-1}$, generates uniformly random bits $r_1, \ldots, r_{q-2}$ and invokes a 1-out-of-2 OT protocol with each pair of messages

$$(m_0, r_1), (m_1 \oplus r_1, r_1 \oplus r_2), \ldots, (m_{q-3} \oplus r_{q-3}, r_{q-3} \oplus r_{q-2}), (m_{q-2} \oplus r_{q-2}, m_{q-1} \oplus r_{q-2}).$$

If Bob wants to fetch the message indexed by $b \in \mathbb{F}_q$, then in the first b protocol executions he recovers the messages

$$r_1, r_1 \oplus r_2, \ldots, r_{b-1} \oplus r_b, m_b \oplus r_b$$

and he follows the protocol arbitrarily (but honestly) in later runs. Then, after all the 1-out-of-2 OT protocol executions have concluded, the message is reconstructed by computing the exclusive OR of the first b obtained messages. This approach can be optimized so that only $\log q$ runs are necessary [33].

By virtue of the previous procedure, there exists a reduction from 1-out-of-2 OT to 1-out-of-q OT. While this is indeed interesting, the proposed reductions introduce at least a logarithmic factor in the round interactivity overhead. To overcome this, 1-out-of-q OT protocols can also be built directly from other techniques, such as Additive Homomorphic Encryption [7], Public-Key Encryption Schemes [34] or Pseudo-Random Codes [35, 36]. As an example, we present the Aiello-Ishai-Reingold 1-out-of-q OT Protocol in Figure 2.5.

The Aiello-Ishai-Reingold OT Protocol

1. Given a public-key additive homomorphic scheme $\mathcal{S} = \{\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}\}$ with message space $\mathcal{M} = \mathbb{Z}_N$ (where we assume $q < N$), Bob generates a public-private key pair $(\mathrm{pk}, \mathrm{sk})$ and sends pk to Alice.
2. Bob chooses a message index $b \in \{0, \ldots, q-1\}$ and sends the ciphertexts $c = \mathsf{Enc}_{\mathrm{pk}}(b+1)$ to Alice.
3. Alice holds the messages $m_0, \ldots, m_{q-1} \in \mathcal{M}$. For each $j \in \{0, \ldots, q-1\}$, Alice generates a random nonce $r_j \overset{\$}{\leftarrow} \mathcal{M}$ and computes the ciphertext $m'_j = cr_j + \mathsf{Enc}_{\mathrm{pk}}(m_j - (j+1)r_j)$. She sends $m'_0, \ldots, m'_{q-1}$ to Bob.
4. Bob takes the message m'_b for his chosen message index b, and computes $\mathsf{Dec}_{\mathrm{sk}}(m'_b) = m_b$.

FIGURE 2.5: The Aiello-Ishai-Reingold OT Protocol.

2.3 Secret Sharing Schemes

In this section, we lay out the theory of secret sharing schemes necessary to develop our results. See [44] for more details.

Secret Sharing, introduced by Shamir [45] and Blakley [46] in 1979, is a cryptographic primitive that is used to protect a *secret value* by distributing it into *shares*. These shares are generated so that it is hard to recover the secret from individual shares, and so that the secret can be recovered by pooling together certain subsets of shares. Secret sharing is used to prevent both the disclosure and the loss of secrets, and it is applied as a primitive in many cryptographic applications, such as secure multi-party computation, attribute-based encryption and distributed cryptography.

In the typical scenario, a user called the *dealer* holds the secret value, and it generates a set of shares. Then, it sends each share privately to a different *participant*.

We next state a formal definition of secret sharing scheme, taken from [47]. This definition abstracts the previous concepts by using random variables, which in practice are sampled to produce secret values and shares.

Definition 2 ([47])**.** *Let $P = \{1, \dots, n\}$ be the set of participants, let $Q = P \cup \{0\}$ and let (Ω, p) be a finite probability distribution. A* Secret Sharing scheme *on P consists of a sequence $\Sigma = (X_i)_{i \in Q}$ of random variables on Ω that take values in some finite sets $E_0, \dots, E_n$ respectively, and which satisfy that, for every event $\{X_1 = x_1, \dots, X_n = x_n\}$ with*

$$p(X_1 = x_1, \dots, X_n = x_n) > 0,$$

there exists a unique $x_0 \in E_0$ such that

$$p(X_0 = x_0 \mid X_1 = x_1, \dots, X_n = x_n) = 1. \tag{2.1}$$

For every vector $\mathbf{r} \in \Omega$, the value $X_0(\mathbf{r})$ is called the *secret*, and the values $(X_i(\mathbf{r}))_{i \in P}$ are *shares* of the secret value $X_0(\mathbf{r})$.

We say that a subset $A \subseteq P$ is *authorized* for Σ when the shares of participants in A determine the secret value, that is, when $(X_i)_{i \in A}$ determines X_0. In other words, a subset $A \subseteq P$ is authorized when, for every event $\{X_1 = x_1, \dots, X_n = x_n\}$ with $p(X_1 = x_1, \dots, X_n = x_n) > 0$, there exists a unique $x_0 \in E_0$ such that

$$p(X_0 = x_0 \mid (X_i = x_i)_{i \in A}) = 1.$$

Similarly, we say that $A \subseteq P$ is *forbidden* for Σ when the shares of participants in A do not reveal any information on the secret value, that is, when $(X_i)_{i \in A}$ does not reveal any information on X_0. In other words, a subset $A \subseteq P$ is forbidden when, for every event $\{X_1 = x_1, \dots, X_n = x_n\}$ with $p(X_1 = x_1, \dots, X_n = x_n) > 0$, and for every pair of elements $x_0, x_0' \in E_0$, we have

$$p(X_0 = x_0 \mid (X_i = x_i)_{i \in A}) = p(X_0 = x_0' \mid (X_i = x_i)_{i \in A}). \tag{2.2}$$

Given a secret sharing scheme Σ, we denote by $\texttt{Reconstruct}_\Sigma$ an efficient algorithm that retrieves the secret value given a set of shares. Thus, if $A \subseteq P$ is authorized for Σ, and $(x_i)_{i \in A}$ are a set of shares of the secret value x_0,

$$x_0 = \texttt{Reconstruct}_\Sigma\left((x_i)_{i \in A}\right).$$

We say that Σ is *perfect* if every subset $A \subseteq P$ is either authorized or forbidden.

We define the *adversary* (resp. *access*) *structure* of Σ as the collection of all forbidden (resp. authorized) subsets $A \subseteq P$ for Σ.

Given a secret sharing scheme Σ on P, the *information ratio* $\sigma(\Sigma)$ of Σ is a quantity that measures the efficiency of secret sharing schemes. It is defined as the ratio of the maximum length in bits of the shares to the length of the secret value

$$\sigma(\Sigma) = \frac{\max_{1 \le i \le n} \log |E_i|}{\log |E_0|}.$$

The schemes Σ with information ratio $\sigma(\Sigma) = 1$ are called *ideal*.

Given an access structure Γ, we define the *minimal sets of* Γ by

$$\min \Gamma = \{A \in \Gamma : B \not\subset A \text{ for all } B \in \Gamma\}.$$

Similarly, given an adversary structure $\mathcal{A}$, we define the *maximal sets of* $\mathcal{A}$ by

$$\max \mathcal{A} = \{A \in \mathcal{A} : A \not\subset B \text{ for all } B \in \mathcal{A}\}.$$

If $\mathcal{A}, \mathcal{B} \subseteq 2^P$ are two adversary structures, we say that they are $\mathcal{R}_2$ when $A \cup B \neq P$ for every $A \in \mathcal{A}, B \in \mathcal{B}$.

We next present some properties of access structures and secret sharing schemes that are used later.

Lemma 3 ([28]). *Let $(\mathcal{A}, \mathcal{B})$ be an $\mathcal{R}_2$ pair of adversary structures, and Σ a perfect secret sharing scheme with $\mathcal{A}$ as its adversary structure. Then, for every $B \in \mathcal{B}$, its complement $\overline{B}$ is authorized in Σ.*

Proof. Assume on the contrary that $\overline{B} \in \mathcal{A}$ for some $B \in \mathcal{B}$. Since $(\mathcal{A}, \mathcal{B})$ is an $\mathcal{R}_2$ pair of adversary structures, the fact that $\overline{B} \cup B = P$ leads to contradiction. □

In this work, when defining a secret sharing scheme $\Sigma = (X_i)_{i \in Q}$, we do not exhibit the random variables X_i explicitly. Instead, given a secret value $x_0 \in E_0$ taken by X_0, we define an algorithmic procedure to sample $X_1, \dots, X_n$ conditioned on $X_0 = x_0$. Hence, we define a procedure to sample a vector $\mathbf{r} \in \Omega$ such that $X_0(\mathbf{r}) = x_0$, and we show how to produce a set of shares $X_1(\mathbf{r}), \dots, X_n(\mathbf{r})$ that satisfies Equation 2.1.

Given a secret sharing scheme Σ, a secret value $x_0 \in E_0$ and a vector $\mathbf{r} \in \Omega$ such that $X_0(\mathbf{r}) = x_0$, we denote by

$$[x_0]_\Sigma = (X_i(\mathbf{r}))_{i \in P} \in E_1 \times \cdots \times E_n$$

the concatenation of a set of shares of x_0 using Σ.

2.3.1 Linear Secret Sharing Schemes

Linear Secret Sharing schemes (LSSS) are a type of secret sharing schemes, and they are key to building our 1-out-of-q OT construction. From here on, we define LSSS, and we restate some of the previous properties for this case. We also provide a result on LSSS needed to prove the security of our construction.

We start with the definition of LSSS, which is taken from [47].

Definition 4 ([47]). *Let $\mathbb{K}$ be a finite field, $P = \{1, \dots, n\}$ and $Q = P \cup \{0\}$. Let (Ω, p) be a finite probability distribution, and let $\Sigma = (X_i)_{i \in Q}$ be a secret sharing scheme, where each random variable X_i on Ω takes values in the finite set E_i.*

Then Σ is called $\mathbb{K}$-linear *(or a* $\mathbb{K}$-Linear Secret Sharing scheme*, written $\mathbb{K}$-LSSS) if $\Omega, E_0, \dots, E_n$ are vector spaces of finite dimension over $\mathbb{K}$, the probability distribution p is uniform on Ω, and the random variables $X_0, \dots, X_n$ are surjective linear mappings.*

In this thesis we just consider $\mathbb{F}_q$-linear secret sharing schemes Σ where $\dim E_0 = 1$. That is, we may assume that $E_0 = \mathbb{F}_q$ and that, for each $i \in P$, the i-th share space is $E_i = \mathbb{F}_q^{\ell_i}$ for some positive integer ℓ_i.

As seen in [44], given a LSSS $\Sigma = (X_i)_{i \in Q}$, a subset $A \subseteq P$ is authorized when

$$\bigcap_{i \in A} \ker X_i \subseteq \ker X_0. \tag{2.3}$$

Similarly, a subset $A \subseteq P$ is forbidden when

$$\bigcap_{i \in A} \ker X_i + \ker X_0 = \Omega.$$

Since X_0 is surjective, if Ω is a non-trivial vector space then $\mathbb{K}$-LSSS are perfect.

The information ratio of a linear secret sharing scheme Σ can be rewritten as

$$\sigma(\Sigma) = \frac{\max_{i \in P} \dim E_i}{\dim E_0}.$$

Ito, Saito and Nishizeki [48] showed that every adversary structure admits an $\mathbb{F}_q$-LSSS for q large enough. However, the share size of these linear secret sharing schemes is generally exponential in n, and this has an impact on the efficiency of our OT combiner. Even worse, by [49] there exist adversary structures that require linear secret sharing schemes with shares of exponential size in n, while by [50, 51], the total share size is $\tilde{O}(2^{cn})$ for some constant $c < 0.95$. On a more positive note, many adversary structures (e.g. threshold) are known to admit ideal linear secret sharing schemes, and by [52] close adversary structures admit linear secret sharing schemes of similar share sizes. The characterization of adversary structures that admit $\mathbb{F}_q$-LSSS with small share sizes is an open problem in secret sharing, and finding efficient $\mathbb{F}_q$-LSSS is essential to the efficiency of our OT combiner.

Given a secret value $x_0 \in \mathbb{F}_q$, we have that $[x_0]_\Sigma \in \mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$. In this case, if we denote by V the set of all possible shares $[0]_\Sigma$ of $0 \in \mathbb{F}_q$, we have that

$$V = \mathrm{Im}_{X_1 \times \cdots \times X_n}(\ker(X_0))$$

is a vector subspace of $\mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$. Similarly, if we denote by W_b the set of all possible shares $[b]_\Sigma$ of a secret value $b \in \mathbb{F}_q$, we have that

$$W_b = [b]_\Sigma + V$$

is an affine subspace of $\mathbb{F}_q^{\ell_1} \times \cdots \times \mathbb{F}_q^{\ell_n}$, where $[b]_\Sigma$ denotes some share of b using Σ.

The following lemma is a consequence of the description of the access structure stated above.

Lemma 5. *Let Σ be an $\mathbb{F}_q$-LSSS consisting of random variables $X_0, X_1, \ldots, X_n$ defined on a non-trivial vector space Ω, and assume that X_0 takes values in $E_0 = \mathbb{F}_q$. Then, a subset $A \subseteq P$ is forbidden for Σ if and only if there exists a vector $\mathbf{r} \in \Omega$ such that $X_0(\mathbf{r}) = 1$ and $X_i(\mathbf{r}) = 0$ for every $i \in A$.*

Proof. For the first implication, since A is forbidden, by Equation 2.3 we see that $\bigcap_{i \in A} \ker X_i \not\subseteq \ker X_0$. Hence, we can consider $\mathbf{r}' \in \left(\bigcap_{i \in A} \ker X_i\right) \setminus \ker X_0$. The vector $\mathbf{r} = X_0(\mathbf{r}')^{-1} \cdot \mathbf{r}'$ satisfies the lemma.

Conversely, consider an arbitrary $\mathbf{r}' \in \Omega$, and let $\mathbf{r}_0 = \mathbf{r}' - X_0(\omega) \cdot \mathbf{r}$. Since the vectors $\mathbf{r}_0 \in \ker X_0$ and $\mathbf{r}_1 = X_0(\mathbf{r}') \cdot \mathbf{r} \in \bigcap_{i \in A} \ker X_i$ satisfy $\mathbf{r}_0 + \mathbf{r}_1 = \mathbf{r}'$, we have that A is forbidden, which concludes the proof. □

2.4 OT combiners

Here we lay out the fundamental theory of OT combiners. First, in Section 2.4.1 we define OT combiners, we name some of their properties and we fix some notation. In Section 2.4.2 we show the 1-out-of-2 OT construction of [28] that our work is based on. Then, in Sections 2.4.3 and 2.4.4 we formalize the correctness and security notions considered in this work.

Before proceeding further, and as in [28], 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, and not a functionality in the sense of Definition 1. Without loss of generality, in this work all 1-out-of-q OT protocols that are considered secure are assumed to follow the footprint of $\mathcal{F}_{OT}$. Figure 2.6 depicts the $\mathcal{F}_{OT}$ ideal functionality.

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 it 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.

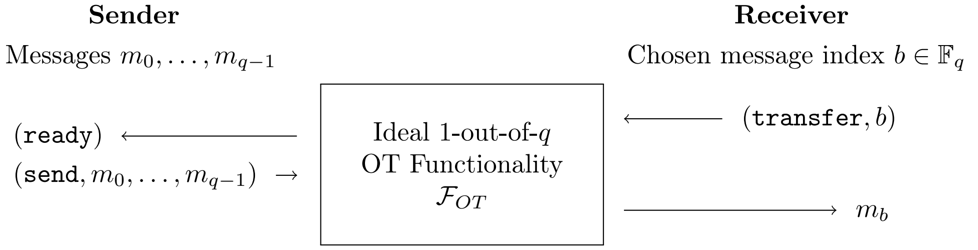
Sender
Messages $m_0, \dots, m_{q-1}$
Receiver
Chosen message index $b \in \mathbb{F}_q$
$(\texttt{ready})$
$(\texttt{send}, m_0, \dots, m_{q-1})$
Ideal 1-out-of-q OT Functionality $\mathcal{F}_{OT}$
$(\texttt{transfer}, b)$
m_b

FIGURE 2.6: The ideal 1-out-of-q Oblivious Transfer functionality.

2.4.1 Definition

Next, we formally define OT combiners.

Definition 6. *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 $S_1, \dots, S_n$, and which implements the OT functionality.*

We say that an OT combiner is 1*-out-of-q* if it implements the 1-out-of-q OT functionality. An OT combiner is *black-box* if, during the protocol, the candidate OT implementations are used in a black-box way, i.e. ignoring their internal workings and making oracle calls as in the ideal OT functionality. Under the black-box assumption, as in [27, 28], we refer to each of the OT candidate implementations as *servers*. An OT combiner is *single-use* if each server is used only once during the execution of the protocol.

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

Definition 7. *We define a* 1*-out-of-q, n-server, single-use and black-box* OT combiner *π by means of the following three polynomial-time algorithms:*

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

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

$m \leftarrow \pi.\mathsf{Reconstruct}(b, (u_1, \ldots, u_n))$: *Deterministic algorithm run by the receiver Bob and taking as input the chosen message index $b \in \mathbb{F}_q$ and n elements $u_1, \ldots, u_n$, where each u_i is received from server S_i. It returns a message m.*

Given an OT combiner $\pi = (\pi.\mathsf{Choose}, \pi.\mathsf{Send}, \pi.\mathsf{Reconstruct})$ and given n servers $S_1, \ldots, S_n$ implementing the 1-out-of-q OT functionality, we can 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 a related tuple $(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$ where $b_i \in \mathbb{F}_q$.
Bob sends $(\texttt{transfer}, b_i)$ to server S_i for $i = 1, \ldots, n$.

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

Sending Phase: The sender Alice chooses q messages $m_0, \ldots, m_{q-1}$.
Alice generates a related tuple $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(m_0, \ldots, m_{q-1})$.
After Alice has received $(\texttt{ready})$ from every server, she sends $(\texttt{send}, u_i^0, \ldots, u_i^{q-1})$ to S_i for $i = 1, \ldots, n$.

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

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

2.4.2 Example: Baseline OT Combiner

As a first example, we describe the baseline construction [28] that our OT combiner is based on. This OT combiner is 1-out-of-2, n-server, single-use and black-box. It satisfies a very strong security definition called *perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries* (cf. 2.4.4).

For the sake of clarity, we describe the OT combiner from [28] in two sections. First, we start with the simpler case where the adversary structure $\mathcal{A}$ admits an ideal $\mathbb{F}_2$-LSSS Σ. Hence, in this case the shares generated using Σ are bit-sized. Then, we comment on the general case where Σ may not be an ideal $\mathbb{F}_2$-LSSS, so the shares generated using Σ may not be bit-sized.

Ideal Case

Let $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ be the adversary structure given in the security definition (cf. 2.4.4), and let Σ be a $\mathbb{F}_2$-LSSS with adversary structure $\mathcal{A}$. Hence, the shares generated using Σ are assumed to be bit-sized. Denote by V the vector subspace of $\mathbb{F}_2^n$ formed by all possible shares $[0]_\Sigma$ of the secret value $0 \in \mathbb{F}_2$. Denote $W_0 = V$, and let $W_1 = [1]_\Sigma + V$ be the affine space formed by all possible shares $[1]_\Sigma$ of the value $1 \in \mathbb{F}_2$.

In order to describe the OT combiner in [28], we first need to define two $\mathbb{F}_2$-LSSS $\mathcal{S}_0, \mathcal{S}_1$, which have $\mathcal{P}_{n,2} = \mathcal{P}_n \times \mathbb{F}_2$ as their set of participants. See Figure 2.7 for a definition of the LSSS $\mathcal{S}_0, \mathcal{S}_1$.

The Secret Sharing Scheme $\mathcal{S}_k$ Defined in [28] in the Ideal Case

Let $V^\perp$ denote the orthogonal space to V, that is

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

To share a message $m \in \mathbb{F}_2$, first

- let $\mathbf{k} = (k_1, \dots, k_n) \in \mathbb{F}_2^n$ be a sharing of k using Σ
- sample $r_1, \dots, r_{n-1} \in \mathbb{F}_2$ uniformly at random, and let $r_n = m - \sum_{i=1}^{n-1} r_i$
- sample $\mathbf{h} = (h_1, \dots, h_n)$ uniformly at random from $V^\perp$

For every $i \in \mathcal{P}_n$ and for every $j \in \mathbb{F}_2$, define the (i,j)-th share as

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

FIGURE 2.7: The $\mathbb{F}_2$-LSSS $\mathcal{S}_k$ related to the affine subspace $W_k \subseteq \mathbb{F}_2^n$ (ideal case).

The main property of the LSSS $\mathcal{S}_k$, as proved in [28], is that their access structures are the families Γ_{W_k} defined by the minimal sets

$$\min \Gamma_{W_k} = \{\{(1, b_1), \dots, (n, b_n)\} \ : \ (b_1, \dots, b_n) \in W_k\}.$$

If $A \in \Gamma_{W_k}$, let $A' \subseteq A$ be an element of $\min \Gamma_{W_k}$ of the form $A' = \{(1, b_1), \dots, (n, b_n)\}$, where $\mathbf{b} = (b_1, \dots, b_n) \in W_k$. Then, we can define the function $\texttt{Reconstruct}_{\mathcal{S}_k}$ on the shares $(m_k^{(i,j)})_{(i,j) \in A}$ of the message m_k as

$$\texttt{Reconstruct}_{\mathcal{S}_k}\left((m_k^{(i,j)})_{(i,j)\in A}\right) = \sum_{i=1}^{n} m_k^{(i,b_i)}$$

To see that this function effectively retrieves m_k, note that

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

since $\sum_{i=1}^{n} r_i = m_k$, $\mathbf{k}, \mathbf{b} \in W_k$ (so $\mathbf{k} - \mathbf{b} \in V$) and $\mathbf{h} \in V^\perp$.

We are now in position to define the OT combiner in [28]. To choose a given message index b, the receiver Bob executes $\pi.\mathsf{Choose}(b)$ to generate a sharing of b for Σ, and he sends each share to a different server. Then, the sender Alice executes $\pi.\mathsf{Send}(m_0, m_1)$ to obtain a sharing of the message m_0 for $\mathcal{S}_0$ and of the message m_1 for $\mathcal{S}_1$. She concatenates these $4n$ shares in a particular way into $2n$ messages, and sends them in pairs to each of the n servers. Finally, Bob executes $\pi.\mathsf{Reconstruct}$ on the received shares, which retrieves m_b by internally using the $\texttt{Reconstruct}_{\mathcal{S}_b}$ function. This construction is, in fact, identical to ours for the case $q = 2$. See Figure 2.8 for an explicit description of this OT combiner.

Baseline OT Combiner [28] in the Ideal Case

π.Choose(b)**:** Given $b \in \mathbb{F}_2$, compute a sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ of b using Σ.
Note that each $b_i \in \mathbb{F}_2$ because Σ is ideal.
Output $(b_1, \ldots, b_n)$.

π.Send(m_0, m_1)**:** Compute a sharing m_0 for $\mathcal{S}_0$

$$[m_0]_{\mathcal{S}_0} = \left(m_0^{(i,j)}\right)_{(i,j)\in\mathcal{P}_{n,2}}$$

and a sharing m_1 for $\mathcal{S}_1$

$$[m_1]_{\mathcal{S}_1} = \left(m_1^{(i,j)}\right)_{(i,j)\in\mathcal{P}_{n,2}}.$$

Output $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}}$, where $u_i^j = m_0^{(i,j)} \| m_1^{(i,j)}$.

π.Reconstruct$(b, (u_1, \ldots, u_n))$**:** Parse each u_i as $u_i = m_0^{(i,b_i)} \| m_1^{(i,b_i)}$.
If $b = 0$, retrieve m_0 by evaluating

$$\texttt{Reconstruct}_{\mathcal{S}_0}\left((m_0^{(i,b_i)})_{i\in\mathcal{P}_n}\right).$$

If $b = 1$, retrieve m_1 by evaluating

$$\texttt{Reconstruct}_{\mathcal{S}_1}\left((m_1^{(i,b_i)})_{i\in\mathcal{P}_n}\right).$$

If the reconstruction fails at any step, output $\mathbf{0}$.

Otherwise, output the reconstructed message m_b.

FIGURE 2.8: The OT combiner by Cascudo et al. [28] (ideal case).

Non-Ideal Case

Now we briefly discuss the general case where Σ may not be an ideal $\mathbb{F}_2$-LSSS, so the shares generated using Σ may not be bit-sized. Hence, the i-th share space E_i of Σ may have dimension $\ell_i \geq 1$, and we denote $E_i = \mathbb{F}_2^{\ell_i}$. Let $\ell = \sum_{i=1}^{n} \ell_i$ be the complexity of Σ.

To generalize the previous construction to the non-ideal case, Cascudo et al. [28] think of the shares for Σ as ℓ-bit strings (instead of n-tuples of $\mathbb{F}_2^{\ell_1} \times \cdots \times \mathbb{F}_2^{\ell_n}$). Therefore, W_0 and W_1 are thought as subspaces of $\mathbb{F}_2^{\ell}$, and the $\mathbb{F}_q$-LSSS $\mathcal{S}_0, \mathcal{S}_1$ (which have $\mathcal{P}_{\ell,2}$ as their set of participants) and the OT combiner (which may use ℓ_i instances of each of the n OT candidates S_i, so it is ℓ-server) are built from Σ exactly as in the previous section. In this work, we also follow this approach (cf. Section 3.2). See [28][Appendix C] for more details.

2.4.3 Correctness Definition

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 that all servers correctly implement the OT functionality and that parties follow the protocol honestly, 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.

Definition 8. *An OT combiner π is* zero-error *if for every message index $b \in \mathbb{F}_q$ and for any q messages $m_0, \dots, m_{q-1}$ we have that*

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

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

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, from which she can see the inputs $(b_i)_{i\in A}$ of $\mathbb{B}$, and where she can also fix the messages $(z_i)_{i\in A}$ that $\mathbb{B}$ receives. Furthermore, she arbitrarily chooses inputs $(u_i^0, \dots, u_i^{q-1})_{i\in\overline{A}}$ for the non-corrupted servers in $\overline{A}$.

Here we have to show that, regardless of how the malicious sender generates input for each server, the obtained protocol is still an OT protocol. In particular, we must ensure that the message index b chosen by $\mathbb{B}$ determines 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 $\pi.\mathsf{Reconstruct}$, must 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 [37], 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 $i \in A$. Servers $i \in \overline{A}$ 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 9. *Let π be a* 1*-out-of-q, n-server OT combiner protocol, and let $\mathcal{F}_{OT}$ denote the ideal* 1*-out-of-q OT functionality. Let* Adv *denote the 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:*

```
$\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ :
    $b \leftarrow \mathbb{B}()$
    $(b_1, \ldots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$
    $\left((u_i^j)_{i \in \overline{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\right) \leftarrow \mathsf{Adv}\left((b_i)_{i \in A}\right)$
    $\textit{output } \pi.\mathsf{Reconstruct}\left(b, \left((u_i^{b_i})_{i \in \overline{A}}, (z_i)_{i \in A}\right)\right)$

$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ :
    $b \leftarrow \mathbb{B}()$
    $(\texttt{ready}) \leftarrow \mathcal{F}_{OT}(\texttt{transfer}, b)$
    $(b_i)_{i \in A} \leftarrow \mathsf{Sim}_1()$
    $\left((u_i^j)_{i \in \overline{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\right) \leftarrow \mathsf{Adv}((b_i)_{i \in A})$
    $(m_0, \ldots, m_{q-1}) \leftarrow \mathsf{Sim}_2\left((u_i^j)_{i \in \overline{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\right)$
    $(\texttt{sent}, m_b) \leftarrow \mathcal{F}_{OT}(\texttt{send}, m_0, \ldots, m_{q-1})$
    $\textit{output } m_b$
```

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*.*

2.4.4 Security Definition

In this section, we discuss the security definition used to capture the security properties of our OT combiner construction.

The security notion considered by Cascudo et al. [28] is called unconditional security. An OT combiner is *unconditionally secure* if its security rests solely on the security assumptions of the OT candidate implementations. That is, if, provided the security of sufficiently many 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 [28], an OT combiner is called *perfectly secure* if it is both unconditionally secure and zero-error (see Definition 8).

In order to capture the notion of unconditional security, we formalize it into a simulator-based security definition [37]. 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 [27, 28] and uses the Universal Composability Framework [53].

Given two adversary structures $\mathcal{A}, \mathcal{B}$, our security definition protects against two types of malicious 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 $A \in \mathcal{A}$ must give Adv no information on b.

This definition uses the simulation paradigm [37], 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 9.

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.

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, which includes the private inputs of $\mathbb{B}$ on the corrupted servers.

Definition 10. *Let π be a 1-out-of-q, n-server 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:*

```
$\mathsf{Real}^{\pi}_{\mathsf{Adv},\mathbb{B}}()$ :
    $b \leftarrow \mathbb{B}()$
    $(b_1, \dots, b_n) \leftarrow \pi.\mathsf{Choose}(b)$
    $\left((u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q}, (z_i)_{i\in A}\right) \leftarrow \mathsf{Adv}\left((b_i)_{i\in A}\right)$
    output $\left((b_i)_{i\in A}, (u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q}, (z_i)_{i\in A}\right)$

$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathsf{Adv},\mathbb{B},\mathsf{Sim}}()$ :
    $b \leftarrow \mathbb{B}()$
    $(\texttt{ready}) \leftarrow \mathcal{F}_{OT}(\texttt{transfer}, b)$
    $(b_i)_{i\in A} \leftarrow \mathsf{Sim}_1()$
    $\left((u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q}, (z_i)_{i\in A}\right) \leftarrow \mathsf{Adv}((b_i)_{i\in A})$
    output $\mathsf{Sim}_{\mathsf{out}}\left((u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q}, (z_i)_{i\in A}\right)$
```

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 $B \in \mathcal{B}$ 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 [37] 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 generates the input $u_i^0, \ldots, u_i^{q-1}$ that is sent to each server S_i. 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 $i \in \overline{B}$. Since the servers $i \in \overline{B}$ are assumed to behave as the ideal $\mathcal{F}_{OT}$ functionality, Adv receives $(u_i^{b_i})_{i \in \overline{B}}$ 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 11. *Let π be a* 1*-out-of-q, n-server 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:*

```
$\mathsf{Real}^{\pi}_{\mathbb{A},\mathsf{Adv}}()$ :
    $(m_0, \ldots, m_{q-1}) \leftarrow \mathbb{A}()$
    $(u_i^j)_{(i,j) \in \mathcal{P}_{n,q}} \leftarrow \pi.\mathsf{Send}(\boldsymbol{send}, m_0, \ldots, m_{q-1})$
    $(b_i)_{i \in \overline{B}} \leftarrow \mathsf{Adv}\left((u_i^j)_{i \in B, j \in \mathbb{F}_q}\right)$
    $\mathit{output}\ \left((u_i^j)_{i \in B, j \in \mathbb{F}_q}, (u_i^{b_i})_{i \in \overline{B}}, (b_i)_{i \in \overline{B}}\right)$

$\mathsf{Ideal}^{\mathcal{F}_{OT}}_{\mathbb{A},\mathsf{Adv},\mathsf{Sim}}()$ :
    $(u_i^j)_{i \in B, j \in \mathbb{F}_q} \leftarrow \mathsf{Sim}_1()$
    $(b_i)_{i \in \overline{B}} \leftarrow \mathsf{Adv}\left((u_i^j)_{i \in B, j \in \mathbb{F}_q}\right)$
    $\tilde{b} \leftarrow \mathsf{Sim}_2\left((b_i)_{i \in \overline{B}}\right)$
    $(\boldsymbol{ready}) \leftarrow \mathcal{F}_{OT}(\boldsymbol{transfer}, \tilde{b})$
    $(m_0, \ldots, m_{q-1}) \leftarrow \mathbb{A}()$
    $(\boldsymbol{sent}, m_{\tilde{b}}) \leftarrow \mathcal{F}_{OT}(\boldsymbol{send}, m_0, \ldots, m_{q-1})$
    $\mathit{output}\ \mathsf{Sim}_{\mathsf{out}}\left(\tilde{b}, m_{\tilde{b}}, (b_i)_{i \in \overline{B}}\right)$
```

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 make up the security definition considered in this work, namely perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries. We formally state this in the next definition.

Definition 12. *Let* π *be a* 1*-out-of-*q*,* n*-server 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 both perfectly secure for the sender against active* $\mathcal{B}$*-adversaries and for the receiver against active* $\mathcal{A}$*-adversaries.*

Finally, we state a result that characterizes the pairs $(\mathcal{A}, \mathcal{B})$ of adversary structures for which perfectly secure OT combiners are known to be impossible to attain.

Proposition 13 ([54])**.** *If* $(\mathcal{A}, \mathcal{B})$ *is not an* $\mathcal{R}_2$ *pair of adversary structures, then perfectly secure OT combiners against active* $(\mathcal{A}, \mathcal{B})$*-adversaries cannot exist.*

Chapter 3

One-out-of-q OT Combiners

In this chapter we present our research contributions. We introduce our 1-out-of-q OT combiner, which can be seen as an extension of the OT combiner in [28] to the 1-out-of-q scenario. In this scenario, the produced OT functionality takes $q \geq 2$ messages $m_0, \ldots, m_{q-1} \in \mathbb{F}_q$ from Alice and an element $b \in \mathbb{F}_q$ from Bob, and outputs the message m_b to Bob.

This chapter is divided in two sections. For the sake of simplicity, and following [28], in Section 3.1 we start by introducing a simplified version of our construction, where the adversary structure $\mathcal{A}$ of the security definition admits an ideal $\mathbb{F}_q$-linear secret sharing scheme. Next, in Section 3.2 we describe our construction in full generality, thus achieving an OT combiner with perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries, where $(\mathcal{A}, \mathcal{B})$ denotes an $\mathcal{R}_2$ pair of adversary structures.

3.1 Ideal Case

The described 1-out-of-q OT protocol is proven secure against any $(\mathcal{A}, \mathcal{B})$-adversary (see Definition 12), where $\mathcal{A}, \mathcal{B} \subseteq 2^{\mathcal{P}_n}$ is any pair of $\mathcal{R}_2$ adversary structures such that $\mathcal{A}$ admits an $\mathbb{F}_q$-LSSS. Throughout this section, we assume that the pair $(\mathcal{A}, \mathcal{B})$ of adversary structures is fixed, and that the adversary structure $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-LSSS Σ. The efficiency of our OT combiner is affected by the size of the shares of Σ, and it is best in this ideal case. We note that the characterization of adversary structures that admit $\mathbb{F}_q$-LSSS with small share sizes is an open problem in secret sharing. See Section 2.3.1 or [44] for more details.

This section is organized as follows. In Section 3.1.1 we develop the necessary tools to extend the previous scheme of [28] to suit our purposes. Then, in Section 3.1.2 we explicitly describe our 1-out-of-q OT combiner for the particular case where $\mathcal{A}$ admits a perfect ideal $\mathbb{F}_q$-LSSS. Finally, in Sections 3.1.3 and 3.1.4, we respectively provide proofs of the correctness and of the security of our construction.

3.1.1 OT-Compatible Secret Sharing Schemes

Let $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$ be an adversary structure on the set $\mathcal{P}_n = \{1, \ldots, n\}$, and let Σ be an ideal $\mathbb{F}_q$-LSSS for the set $\mathcal{P}_n$ of n participants with $\mathcal{A}$ as its adversary structure. As in [28], the scheme Σ are used by the receiver Bob to request the message with the selected index $b \in \mathbb{F}_q$, simply by generating a sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ of b under Σ and sending each share $b_i \in \mathbb{F}_q$ to the corresponding server S_i.

Denote by $V \subseteq \mathbb{F}_q^n$ the vector space consisting of all the sharings of 0 under the scheme Σ. Given any $b \in \mathbb{F}_q$, let $W_b \subseteq \mathbb{F}_q^n$ be the affine subspace of sharings of b for Σ. Note that, by this definition, $V = W_0$. Since Σ is an $\mathbb{F}_q$-LSSS, we can express $W_b = \mathbf{b} + V$, where $\mathbf{b} = [b]_\Sigma$ is a sharing of b for Σ. We can also express $\mathbb{F}_q^n$ as the disjoint union $\mathbb{F}_q^n = W_0 \cup \cdots \cup W_{q-1}$.

In order to let Alice send the messages $m_0, \ldots, m_{q-1}$ to each server, our construction follows the strategy of [28] and makes use of secret sharing schemes that are related to affine subspaces $W \subseteq \mathbb{F}_q^n$. All such schemes proposed here are defined on the set of nq participants $\mathcal{P}_{n,q} = \mathcal{P}_n \times \mathbb{F}_q$. 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 $i = 1, \ldots, n$.

The next definition associates an access structure Γ_W to each $W \subseteq \mathbb{F}_q^n$.

Definition 14. *Let $q > 1$ 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 sets*

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

This definition generalizes the access structures of the $\mathbb{F}_q$-LSSS $\mathcal{S}_0, \mathcal{S}_1$ of the OT combiner of [28] (see Section 2.4.2). In detail, recall that the first step of their OT combiner consists in Bob taking a selected message index $b \in \mathbb{F}_2$ and a sharing $[b]_\Sigma = (b_1, \ldots, b_n) \in W_b$ of b using an ideal $\mathbb{F}_2$-LSSS Σ, and then sending each share b_i to server S_i. In turn, Alice generates two sharings $[m_k]_{\mathcal{S}_0} = (m_k^{(i,0)})_{i \in \mathcal{P}_n}$ and $[m_k]_{\mathcal{S}_1} = (m_k^{(i,1)})_{i \in \mathcal{P}_n}$ of m_k for each $k \in \mathbb{F}_2$ by using the ideal $\mathbb{F}_2$-LSSS $\mathcal{S}_0, \mathcal{S}_1$ for Γ_{W_0} and Γ_{W_1}, and distributes the messages $m_0^{(i,0)} \| m_1^{(i,0)}, m_0^{(i,1)} \| m_1^{(i,1)}$ to each server S_i. At the end of the protocol, assuming Alice behaved honestly, Bob receives the shares $m_k^{(i,b_i)}$ for $i = 1, \ldots, n$. The reconstruction of m_b is then possible, since the legitimately received shares $m_b^{(1,b_1)}, \ldots, m_b^{(n,b_n)}$ of m_b correspond precisely to a minimal set of the access structure Γ_{W_b}.

In the 1-out-of-q scenario, the sender Alice holds q messages $m_0, \ldots, m_{q-1}$. To generalize the construction in [28] to this scenario, we would need to instantiate q $\mathbb{F}_q$-LSSS $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$ on the set of participants $\mathcal{P}_{n,q} = \mathcal{P}_n \times \mathbb{F}_q$, where $\mathcal{S}_k$ has access structure Γ_{W_k} for each $k \in \mathbb{F}_q$. Then, Alice would generate an independent sharing

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

of each message m_k, and she would send exactly q of these shares, $m_k^{(i,0)}, \ldots, m_k^{(i,q-1)}$, to each OT server S_i for each message m_k. Since this requires exactly q shares per server, we would need the $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ for Γ_{W_k} to be ideal for each $k \in \mathbb{F}_q$.

In [28] Cascudo et al. prove that, if $W \subseteq \mathbb{F}_2^n$ is an affine subspace, then the access structure Γ_W described above always admits an ideal $\mathbb{F}_2$-LSSS. 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. While $\mathbb{F}_q$-LSSS are guaranteed to exist for any such access structure thanks to [48], the ideality requirement may prove harder to obtain. Hence, we can not just take the course of action described above.

The main idea of this work is that, instead of aiming for $\mathbb{F}_q$-LSSS with access structures of the form Γ_W, it is possible to relax the conditions on the access structure and still be able to construct ideal schemes that fit our security needs. We accordingly propose the notion of *W-OT-compatibility.*

Definition 15. *Let $W \subseteq \mathbb{F}_q^n$. Let $\Delta \subseteq 2^{\mathcal{P}_{n,q}}$ be the family of subsets defined by*

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

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 $\mathbb{F}_q$-LSSS 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.

In particular, the obtained $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ satisfy the condition that the knowledge of any two distinct shares sent to server S_i leads to the knowledge of all q of them. Under this assumption, 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 to be used by Alice 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 it is W-OT-compatible.

We now give an example to make the last definitions clear.

Example 16. *Let* $n = q = 3$. *Then,*

$$\begin{aligned}
&P_1 = \{(1,0),(1,1),(1,2)\}\,,\\
&P_2 = \{(2,0),(2,1),(2,2)\}\,,\\
&P_3 = \{(3,0),(3,1),(3,2)\}\,,\\
&\mathcal{P}_{3,3} = P_1 \cup P_2 \cup P_3\\
&\Delta = \{A \subseteq \mathcal{P}_{3,3} \,:\, |A \cap P_i| = 0, 1 \text{ or } 3 \text{ for } i = 1,2,3\}.
\end{aligned}$$

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*

$$\begin{aligned}
&\mathbf{k} = (1,1,1)\\
&V = \langle (1,0,2) \rangle_{\mathbb{F}_3} = \{(0,0,0),(1,0,2),(2,0,1)\}\,,
\end{aligned}$$

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

$$\min \Gamma_W = \{\{(1,1),(2,1),(3,1)\},\{(1,2),(2,1),(3,0)\},\{(1,0),(2,1),(3,2)\}\}\,.$$

We note that Γ_W *is* W*-OT-compatible. Now, consider the access structures* $\Gamma_1, \Gamma_2, \Gamma_3$ *on* $\mathcal{P}_{3,3}$ *determined by the following minimal sets*

$$\begin{aligned}
&\min \Gamma_1 = \{\{(1,1),(2,1),(3,1)\},\{(1,2),(2,1),(3,0)\}\}\\
&\min \Gamma_2 = \{(2,1)\}\\
&\min \Gamma_3 = \min \Gamma_W \cup \{(1,1),(2,1),(3,2),(3,3)\}
\end{aligned}$$

Since $\{(1,0),(2,1),(3,2)\}$ *is in* $\Gamma_W \cap \Delta$ *but not in* Γ_1, *we have that* Γ_1 *is not* W*-OT-compatible. As for* Γ_2, *while* $\Gamma_W \subseteq \Gamma_2$, *we have sets of* $\Gamma_2 \cap \Delta$, *such as* P_2 *or* $\{(1,1),(2,1),(3,2)\}$, *that do not belong to* Γ_W. *In general, any* W*-OT-compatible access structure* Γ *must satisfy* $\min \Gamma_W \subseteq \min \Gamma$.

Lastly, we see that Γ_3 is W-OT-compatible. The reason is that, for any set $A \in \Gamma_3 \cap \Delta$ that contains $\{(1,1),(2,1),(3,2),(3,3)\}$, 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 $\Gamma_3 \cap \Delta \subseteq \Gamma_W \cap \Delta$. This demonstrates that W-OT-compatible access structures may have minimal sets outside of $\min \Gamma_W$.

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

Remark 17. *If an access structure $\Gamma \subseteq 2^{\mathcal{P}_{n,q}}$ is W-OT-compatible, then*

- $\{(1,b_1),\ldots,(n,b_n)\} \in \Gamma$ *for every* $\mathbf{b} \in W$ *(in fact,* $\min \Gamma_W \subseteq \min \Gamma$*),*
- $\{(1,v_1),\ldots,(n,v_n)\} \notin \Gamma$ *for every* $\mathbf{v} \notin \mathbb{F}_q^n \setminus W$*,*
- $\mathcal{P}_{n,q} \setminus P_i \notin \Gamma$ *for* $i = 1,\ldots,n$*,*
- *if* $A \in \Gamma$ *has size* $|A| = n$*, then* $A \in \min \Gamma_W$*,*
- *if* $A \in \mathcal{P}_{n,q}$ *has size* $|A| < n$*, then* $A \notin \Gamma$*.*

Given $k \in \mathbb{F}_q$, we instantiate the $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ associated to the affine subspace W_k in Figure 3.1. The scheme $\mathcal{S}_k$ is used by Alice to generate the input to each OT server for a single message. It is defined on the set of nq participants $\mathcal{P}_{n,q}$ and it is $\mathbb{F}_q$-linear and ideal.

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

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

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

FIGURE 3.1: The $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ related to the affine subspace $W_k \subseteq \mathbb{F}_q^n$.

As in [28] (see Section 2.4.2), if $A \subseteq \mathcal{P}_{n,q}$ contains a set $A' \in \min \Gamma_{W_k}$ of the form $A' = \{(1,b_1),\ldots,(n,b_n)\}$, where $\mathbf{b} = (b_1,\ldots,b_n) \in W_k$, we can then define the function $\texttt{Reconstruct}_{\mathcal{S}_k}$ on the shares $(m_k^{(i,j)})_{(i,j)\in A}$ of the message m_k as

$$\texttt{Reconstruct}_{\mathcal{S}_k}\left((m_k^{(i,j)})_{(i,j)\in A}\right) = \sum_{i=1}^{n} m_k^{(i,b_i)}$$

To see that this function effectively retrieves m_k, note that

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

since $\sum_{i=1}^{n} r_i = m$, $\mathbf{k}, \mathbf{b} \in W_k$ (so $\mathbf{k} - \mathbf{b} \in V$) and $\mathbf{h} \in V^\perp$.

The following proposition states that the $\mathbb{F}_q$-LSSS $\mathcal{S}_k$ satisfies the properties required for our purposes.

Proposition 18. *For every $k \in \mathbb{F}_q$, the secret sharing scheme $\mathcal{S}_k$ defined in Figure 3.1 is $\mathbb{F}_q$-linear, perfect, ideal and W_k-OT-compatible.*

Before we are able to prove this proposition, we need the following technical lemma.

Lemma 19. *Let $\mathbb{F}_q$ be a finite field with $q \geq 2$ and $V \subset \mathbb{F}_q^n$ be a vector subspace. Let $y_1, \dots, y_t \in \mathbb{F}_q$. 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. 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 $\langle y, \mathbf{h} \rangle = 1$.

Now, assume that $t < n$, and that we have $y_1, \dots, y_t \in \mathbb{F}_q$ such that

$$(y_1, \dots, y_t, x_{t+1}, \dots, x_n) \notin V \text{ for all } x_{t+1}, \dots, x_n \in \mathbb{F}_q.$$

By induction hypothesis, for every $x \in \mathbb{F}_q$, there exists an $\mathbf{h}^x = (h_1^x, \dots, h_n^x) \in V^\perp$ such that

$$\begin{aligned} &\sum_{i=1}^{t} y_i h_i^x + x h_{t+1}^x = 1 \\ &h_{t+2}^x = \cdots = h_n^x = 0. \end{aligned}$$

If $h_{t+1}^x = 0$, for some $x \in \mathbb{F}_q$, then $\mathbf{h}^x$ satisfies the lemma. Otherwise, let x and x' be two distinct elements of $\mathbb{F}_q$ such that $h_{t+1}^x = h_{t+1}^{x'} \neq 0$. Define

$$\mathbf{h} = \frac{\mathbf{h}^x - \mathbf{h}^{x'}}{h_{t+1}^x (x' - x)} \in V^\perp.$$

Since $\mathbf{h} = (h_1, \dots, h_n)$ satisfies $h_{t+1} = \cdots = h_n = 0$ and

$$\begin{aligned} y_1 h_1 + \cdots + y_t h_t &= \frac{1}{h_{t+1}^x (x' - x)} \left(\sum_{i=1}^{t} y_i h_i^x - \sum_{i=1}^{t} y_i h_i^{x'} \right) \\ &= \frac{1}{h_{t+1}^x (x' - x)} \left((1 - x h_{t+1}^x) - (1 - x' h_{t+1}^{x'}) \right) \\ &= 1 \end{aligned}$$

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

We next prove Proposition 18.

Proof. In order to share a secret $m \in \mathbb{F}_q$ in the considered scheme $\mathcal{S}_k$, the sender chooses $r_1, \dots, r_{n-1} \in \mathbb{F}_q$ uniformly at random, sets $r_n = m - \sum_{i=1}^{n-1} r_i$ and chooses $\mathbf{h} = (h_1, \dots, h_n) \in V^\perp$ uniformly at random. The share of participant (i, j) is, then

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

where $\mathbf{k} = (k_1, \dots, k_n)$ is a sharing of k using Σ, and we denote $W_k = \mathbf{k} + V$.

This scheme is ideal, since each participant in $\mathcal{P}_{n,q}$ is assigned a single share in $\mathbb{F}_q$, and it is $\mathbb{F}_q$-linear.

Now we prove that the access structure Γ of the considered secret sharing scheme is W_k-OT-compatible.

First, we prove that $\Gamma_{W_k} \cap \Delta \subseteq \Gamma \cap \Delta$. Let $\mathbf{w} = (w_1, \ldots, w_n) \in W$ and set $A = \{(1, w_1), \ldots, (n, w_n)\}$. Since $\mathbf{w} = \mathbf{k} + \mathbf{v}$ for some $\mathbf{v} = (v_1, \ldots, v_n) \in V$, we have

$$\sum_{(i,j)\in A} m^{(i,j)} = \sum_{i=1}^{n}(r_i + (k_i - w_i)h_i) = \sum_{i=1}^{n} r_i - \langle \mathbf{v}, \mathbf{h} \rangle = \sum_{i=1}^{n} r_i = m$$

and so $\{(1, w_1), \ldots, (n, w_n)\} \in \min \Gamma$ for every $\mathbf{w} \in W_k$. Hence, $\Gamma_{W_k} \subseteq \Gamma$.

To show that $\Gamma \cap \Delta \subseteq \Gamma_{W_k} \cap \Delta$ we see that, for every $A \in \Delta$, if $A \notin \Gamma_{W_k}$ then $A \notin \Gamma$. Assume, without loss of generality, that $A = \{(1, v_1), \ldots, (t, v_t)\} \cup P_{t+1} \cup \cdots \cup P_n$. Hence, we have that $(v_1, \ldots, v_t, x_{t+1}, \ldots, x_n) \notin W_k$ for every $x_{t+1}, \ldots, x_n \in \mathbb{F}_q$. By the previous lemma, there exist 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 $h_{t+1} = \cdots = h_n = 0$.

By considering such an $\mathbf{h} \in V^\perp$ and the following choice of randomness

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

we get a sharing of the message $m = 1$ such that $m^{(i,j)} = 0$ for every $(i, j) \in A$. The theorem follows by applying Lemma 5. □

3.1.2 Our One-out-of-q OT Combiner in the Ideal Case

Let Σ be an ideal $\mathbb{F}_q$-LSSS for n participants, with adversary structure $\mathcal{A} \subseteq 2^{\mathcal{P}_n}$. The shares generated with this scheme are used by Bob to query each server. Also, denote by $\mathcal{S}_k$ the ideal $\mathbb{F}_q$-LSSS defined previously in Figure 3.1 for $k \in \mathbb{F}_q$. Remind that the scheme $\mathcal{S}_k$ is attached to the affine subspace $W_k \subseteq \mathbb{F}_q^n$ determined by $W_k = \mathbf{k} + V$, where $\mathbf{k}$ is a sharing of k for the scheme Σ and $V \subseteq \mathbb{F}_q^n$ is the vector space consisting of all the sharings of 0 for the scheme Σ.

We are now in position to describe our 1-out-of-q OT combiner in the case that the adversary structure $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-LSSS. 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 2.6). The proposed construction is defined in Figure 3.2 below.

3.1.3 Correctness Proof

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

Proposition 20. *The OT combiner π_{OT} defined in Figure 3.2 is zero-error. That is, π_{OT} implements the 1-out-of-q OT functionality correctly, provided that both Alice and Bob are semi-honest.*

Proof. If Alice and Bob follow the protocol honestly, at the end of the protocol Bob receives the values $m_b^{(1,b_1)}, \ldots, m_b^{(n,b_n)}$ for some sharing $[b]_\Sigma = (b_1, \ldots, b_n) \in W_b$ of his input b. Since $\mathcal{S}_b$ is W_b-OT-compatible by Proposition 18, the set $\{(1, b_1), \ldots, (n, b_n)\}$ is authorized for $\mathcal{S}_b$, and thus Bob can use the algorithm $\texttt{Reconstruct}_{\mathcal{S}_b}$ to reconstruct the message m_b. □

Now, we consider the case of Definition 9, where Alice is controlled by an active adversary Adv.

Our 1-out-of-q OT Combiner π_{OT}

$\pi_{OT}.\mathsf{Choose}(b)$**:** 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 $(b_1, \ldots, b_n)$.

$\pi_{OT}.\mathsf{Send}(m_0, m_1)$**:** 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 every $(i, j) \in \mathcal{P}_{n,q}$, compute the values

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

Output $(u_i^j)_{(i,j)\in\mathcal{P}_{n,q}}$.

$\pi_{OT}.\mathsf{Reconstruct}(b, (u_1, \ldots, u_n))$**:** Parse each $u_i^{b_i}$ as

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

where $m_k^{(i,b_i)} \in \mathbb{F}_q$ for each $i \in \mathcal{P}_n$.
If $b = k$, retrieve m_b by evaluating

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

If the reconstruction fails at any step, output $\mathbf{0}$.

Otherwise, output the reconstructed message m_b.

FIGURE 3.2: Our 1-out-of-q OT combiner π_{OT} in the case where the access structure $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-LSSS Σ.

Proposition 21. *Let $(\mathcal{A}, \mathcal{B})$ be an $\mathcal{R}_2$ pair of adversary structures, and assume that the adversary structure $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-LSSS Σ. Then the OT combiner π_{OT} defined in Figure 3.2 implements the OT functionality correctly for the receiver against active $\mathcal{A}$-adversaries (see Definition 9).*

Proof. We start by defining the simulator appearing in Definition 9, 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 $0 \in \mathbb{F}_q$,

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

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

$\mathsf{Sim}_2((u_i^j)_{i\in\overline{A}, j\in\mathbb{F}_q}, (u_i)_{i\in A})$: Retrieve, from the state of Sim, the sharing $[0]_\Sigma = (b_i^0)_{i\in\mathcal{P}_n}$ that was generated in the previous execution of Sim_1.

Generate uniformly random sharings of every nonzero element of $\mathbb{F}_q$,

$$\begin{aligned} [1]_\Sigma =& (b_1^1, \ldots, b_n^1), \\ &\vdots \\ [q-1]_\Sigma =& (b_1^{q-1}, \ldots, b_n^{q-1}), \end{aligned}$$

subject to the restriction that $b_i^k = b_i^0$ for every $k \in \mathbb{F}_q \backslash \{0\}$ and for every $i \in A$. Note that these sharings exist by Equation 2.2, since A is forbidden for Σ. This may be done by choosing a random sharing $[0]_\Sigma$ first, and then generating the sharings of the other $k \in \mathbb{F}_q$ subject to the restriction that $b_i^k = b_i^0$ for all $i \in A$. In practice, this requires showing a solution of 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 the specified form (as it has been malformed by Alice), set $m_k = \mathbf{0}$ 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 $\mathbf{0}$ in the previous step, then try to reconstruct Alice's input by executing

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

If the reconstruction succeeds, let m_k be its output. Otherwise, set $m_k = \mathbf{0}$.

Output $(m_0, \ldots, m_{q-1})$.

In order to prove indistinguishability remind first that, in the real world, Bob generates a sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ 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 $\left((u_i^j)_{i \in \overline{A}, j \in \mathbb{F}_q}, (z_i)_{i \in A}\right)$ generated by Adv are identically distributed in both the real and the ideal world.

Also because of the previous observation, the sharing $[b]_\Sigma = (b_1, \ldots, b_n)$ generated in the real world and the sharing $[b]_\Sigma = (b_1^b, \ldots, b_n^b)$ generated by Sim are indistinguishable.

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

□

3.1.4 Security Proof

The following proposition states the security properties of our construction.

Proposition 22. *Let $(\mathcal{A}, \mathcal{B})$ be an $\mathcal{R}_2$ pair of adversary structures, and assume that the adversary structure $\mathcal{A}$ admits an ideal $\mathbb{F}_q$-LSSS Σ. Then the OT combiner π_{OT} defined in Figure 3.2 is perfectly secure against active $(\mathcal{A}, \mathcal{B})$-adversaries (see Definition 12).*

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

Lemma 23. *Let $m_0, \ldots, m_{q-1} \in \mathbb{F}_q$ 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 $B \subseteq \{1, \ldots, n\}$ and $(b'_1, \ldots, b'_n) \in \mathbb{F}_q^n$, and define the set $\mathcal{H} \subseteq \mathcal{P}_{n,q}$ by*

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

Fix $b \in \mathbb{F}_q$. Then, if the shares $(b'_i)_{i\in\overline{B}}$ 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, the only shares that could potentially give any information on m_b are $(m_b^{(i,j)})_{(i,j)\in\mathcal{H}}$. Hence, we need to 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 $i \in \overline{B}$, and this would mean that $(b'_i)_{i\in\overline{B}}$ belongs to a sharing $[b]_\Sigma$, a contradiction. □

We can now proceed to the proof of Proposition 22.

Proof. The proof is split in two parts, corresponding to Definitions 10 and 11. In each case, we define the simulators and compare the output of the ideal experiment to that of the real experiment.

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

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

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

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

$\mathsf{Sim}_{\mathsf{out}}((u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q},(z_i)_{i\in A})$: 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 $\Big((b_i^0)_{i\in A},(u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q},(z_i)_{i\in A}\Big)$

We prove indistinguishability in a similar fashion than in Proposition 21.

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. Because of this, these shares are distributed identically to the A-shares in a uniformly random sharing of any other $b' \neq b$ (in particular, of $0 \in \mathbb{F}_q$). Moreover, they do not carry any information on b, so the messages $\Big((u_i^j)_{i\in\overline{A},j\in\mathbb{F}_q},(z_i)_{i\in A}\Big)$ generated by Adv are 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 $m'_k \in \mathbb{F}_q$ at random and generate the sharing

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

Then, create the values $u_i^j = m'^{(i,j)}_0||\cdots||m'^{(i,j)}_{q-1}$ for every $(i,j) \in B \times \mathbb{F}_q$.

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

$\mathsf{Sim}_2((b_i)_{i\in\overline{B}})$: Try to reconstruct the input b of the adversary Adv by executing the $\mathtt{Reconstruct}_\Sigma$ function over the input to non-corrupted servers, i.e., by executing $\mathtt{Reconstruct}_\Sigma((b_i)_{i\in\overline{B}})$.

If the reconstruction succeeds, output the reconstructed message index $\tilde{b}$.

If the reconstruction fails, output $\perp$.

$\mathsf{Sim}_{\mathsf{out}}(\tilde{b}, m_{\tilde{b}}, (b_i)_{i\in\overline{B}})$: Retrieve, from the state of Sim and for every k, 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 $\tilde{m}_{\tilde{b}} = m_{\tilde{b}}$ and $\tilde{m}_k = m'_k$ for $k \in \mathbb{F}_q \setminus \{\tilde{b}\}$. Then, generate a sharing
$$[\tilde{m}_{\tilde{b}}]_{\mathcal{S}_{\tilde{b}}} = ({m'}_{\tilde{b}}^{(i,j)})_{(i,j)\in\mathcal{P}_{n,q}}$$
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 $\mathcal{S}_0, \ldots, \mathcal{S}_{q-1}$). For every $k \in \mathbb{F}_q \setminus \{\tilde{b}\}$, set
$$\tilde{m}_k^{(i,j)} = {m'}_k^{(i,j)} \quad \text{for every } (i,j) \in \mathcal{P}_{n,q}.$$

- If $\tilde{b} = \perp$ then, for every $k \in \mathbb{F}_q$, let
$$\begin{aligned}\tilde{m}_k &= {m'}_k \\ \tilde{m}_k^{(i,j)} &= {m'}_k^{(i,j)} \quad \text{for every } (i,j) \in \mathcal{P}_{n,q}.\end{aligned}$$

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

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

In order to prove indistinguishability we first note that, by Lemma 3, the set $\overline{B}$ 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 $(b_i)_{i\in\overline{B}}$ generated by the adversary Adv in the real world and in the ideal world are identically distributed.

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

In the first case, Sim_2 successfully reconstructs b. The set
$$\{(i, b_i) \;:\; i \in \overline{B}\} \cup (B \times \mathbb{F}_q)$$
is then authorized for $\mathcal{S}_b$ and, by Lemma 23, 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 23 shows that the shares output by $\mathsf{Sim}_{\mathsf{out}}$ give no information about m_b. Therefore, since here $\mathsf{Sim}_{\mathsf{out}}$ generates them from random messages, they obey the same distribution as in the real world, as required.

□

3.2 Non-Ideal Case

In this section, we show how our protocol π_{OT} from Section 3.1.2 extends to the general case where the adversary structure $\mathcal{A}$ does not necessarily admit an ideal $\mathbb{F}_q$-linear secret sharing scheme.

3.2.1 OT-Compatible Secret Sharing Schemes

Let Σ be an $\mathbb{F}_q$-linear secret sharing scheme 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 message index, and in this case Bob can not just send the share $\tilde{b}_i$ to each server S_i.

Instead, denote by $\ell = \sum_{i=1}^n \ell_i$ the complexity 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, if $\Sigma = (X_0, X_1, \ldots, X_n)$, we denote by $\Sigma' := (X'_0, X'_1, \ldots, X'_\ell)$ the ideal $\mathbb{F}_q$-LSSS defined by

$$\begin{aligned} X'_0 &= X_0, \\ X'_{i+j-1} &= \rho_j \circ X_i \quad \text{for } 1 \leq i \leq n \text{ and } 1 \leq j \leq \ell_i, \end{aligned}$$

where $\rho_j : \mathbb{F}_q^{\ell_i} \to \mathbb{F}_q$ is the j-th projection map. That is, $[b]_{\Sigma'} = \varphi([b]_\Sigma) = (b_1, \ldots, b_\ell)$ 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 $k \in \mathbb{F}_q$, we instantiate the $\mathbb{F}_q$-LSSS $\mathcal{S}'_k$ associated to the affine subspace W'_k in Figure 3.3. 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 $\mathcal{S}'_k$

To share a message $m \in \mathbb{F}_q$, first

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

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

FIGURE 3.3: The $\mathbb{F}_q$-LSSS $\mathcal{S}'_k$ related to the affine subspace $W'_k \subseteq \mathbb{F}_q^{\ell}$.

As in the previous case, if $A \subseteq \mathcal{P}_{\ell,q}$ contains a set $A' \in \min \Gamma_{W'_k}$ of the form $A' = \{(1, b_1), \ldots, (\ell, b_\ell)\}$, where $\mathbf{b} = (b_1, \ldots, b_\ell) \in W'_k$, we can then define the function

$\texttt{Reconstruct}_{\mathcal{S}'_k}$ on the shares $(m_k^{(i,j)})_{(i,j)\in A}$ of the message m_k as

$$\texttt{Reconstruct}_{\mathcal{S}'_k}\left((m_k^{(i,j)})_{(i,j)\in A}\right) = \sum_{i=1}^{\ell} m_k^{(i,b_i)}$$

To see that this function effectively retrieves m_k, note that

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

since $\sum_{i=1}^{\ell} r_i = m$, $\mathbf{k}, \mathbf{b} \in W'_k$ (so $\mathbf{k} - \mathbf{b} \in V'$) and $\mathbf{h} \in V'^{\perp}$.

As a direct consequence of Proposition 18 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.

3.2.2 Our One-out-of-q OT Combiner in the Non-Ideal Case

We now generalize the 1-out-of-q OT combiner presented previously to the case where Σ is not ideal. The obtained 1-out-of-q OT combiner is now ℓ-server (instead of n-server), and it is still single-use and black-box. We describe it in Figure 3.4.

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

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

π'_{OT}.Send(m_0, m_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

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

Output $(u_i^j)_{(i,j)\in\mathcal{P}_{\ell,q}}$.

π'_{OT}.Reconstruct($b, (u_1, \dots, u_n)$): Parse each $u_i^{b_i}$ as

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

where $m_k^{(i,b_i)} \in \mathbb{F}_q$ for each $i \in \mathcal{P}_\ell$.
If $b = k$, retrieve m_b by evaluating

$$\texttt{Reconstruct}_{\mathcal{S}'_k}((m_k^{(i,b_i)})_{i\in\mathcal{P}_\ell}).$$

If the reconstruction fails at any step, output $\mathbf{0}$.

Otherwise, output the reconstructed message m_b.

FIGURE 3.4: Our 1-out-of-q OT combiner π'_{OT} for a general access structure $\mathcal{A}$.

When considering this extension there is, however, a subtlety to take into account. We originally assumed that we have n OT implementations at our disposal, and an $\mathcal{R}_2$ pair $(\mathcal{A}, \mathcal{B})$ of adversary structures representing the capabilities of malicious readers and receivers. Now, the adversary structure $\mathcal{A}'$ is a family of subsets of $\mathcal{P}_\ell$. Hence, in practice, some of the ℓ servers may correspond to the same OT primitive (for example, the first ℓ_1 servers if $\ell_1 \geq 2$). Given $A \in \mathcal{A}$, if a malicious sender corrupts one of such servers, all of the servers implementing the same OT candidate should also be considered as corrupted and be placed into A. And conversely, if one of the servers is not corrupted by the sender, none of them should be placed into A. The same observation applies for the sets $B \in \mathcal{B}$ of servers corrupted by a malicious receiver.

More formally, note that the set of servers is $\mathcal{P}_\ell$, which is in bijection with

$$P' = \{(i,j) \ : \ i \in \mathcal{P}_n, \, j = 1, \ldots, \ell_i\}.$$

Given $i \in \mathcal{P}_n$ denote $P'_i = \{(i,j) \ : \ j = 1, \ldots, \ell_i\}$, so we can express the disjoint union $P' = P'_1 \cup \ldots \cup P'_n$. As stated earlier, we may assume that we have n OT candidates at our disposal, and that the servers in P'_i implement the i-th OT candidate. Since they implement the same OT candidate, they are either corrupted or noncorrupted. To account for this, we can replace the adversary structures in our security and consistency definitions by

$$\mathcal{A}'' = \{\cup_{i \in A} P'_i \ : \ A \in \mathcal{A}\}, \quad \mathcal{B}'' = \{\cup_{i \in B} P'_i \ : \ B \in \mathcal{B}\}.$$

Note that, while the actual adversary structure $\mathcal{A}'$ of Σ' depends on the share spaces $E_1, \ldots, E_n$ of Σ, we know that $\mathcal{A}'' \subseteq \mathcal{A}'$. Therefore, this is consistent with the use of Σ'. Moreover, since $(\mathcal{A}, \mathcal{B})$ is an $\mathcal{R}_2$ pair, so is $(\mathcal{A}'', \mathcal{B}'')$.

The notions of correctness and of security introduced earlier, and all their proofs, translate mutatis mutandis to the non-ideal case by replacing n with ℓ, $\mathcal{A}$ and $\mathcal{B}$ with the adversary structures $\mathcal{A}''$ and $\mathcal{B}''$, V with V', and W_b with W'_b for every $b \in \mathbb{F}_q$.

Chapter 4

Conclusions

Oblivious Transfer (OT) protocols are fundamental cryptographic primitives, as they are used to realize several cryptographic constructions such as Multi-Party Computation protocols, Zero-Knowledge Proofs and Bit Commitment schemes, among others. An OT protocol involves two parties, a sender Alice and a receiver Bob. In the flavor of OT protocols considered in this work, called *1-out-of-q OT*, Alice holds a set of $q \geq 2$ messages. These OT protocol allow Bob to choose and receive only one of these messages, while Alice is oblivious to which message he received.

As for the security of OT, perfectly secure OT protocols have been proved impossible to achieve, and all such protocols rely on some sort of security assumption. In order to mitigate this problem, one can employ OT combiners. Oblivious Transfer combiners are OT protocols that make internal use of various OT implementations. As for security, OT combiners are secure as long as sufficiently many of the used OT implementations are also secure.

The research in OT combiners deals extensively with 1-out-of-2 OT combiners, and constructions with increasing efficiency and security have been developed (see Section 1.3).

This thesis tackles OT combiners for 1-out-of-q OT protocols in the case that $q \geq 2$ is a prime integer. In this case, we build a 1-out-of-q OT combiner by extending the work of Cascudo, Damgård, Farràs and Ranellucci [28], which in turn is based in the construction by Ishai, Maji, Sahai and Wullschleger [26]. Our OT combiner is black-box and single-use. The construction in [28], as ours, is proved secure against malicious adversaries corrupting either one of the parties and a certain set of OT candidates.

The main obstacle when trying to extend the construction in [28] involves building an ideal $\mathbb{F}_q$-linear secret sharing scheme for an affine space $W \subseteq \mathbb{F}_q^n$. We circumvent this problem by introducing the notion of W-OT-compatible secret sharing schemes, and we describe one such a scheme that fits our needs.

We also extend the security and consistency notions of [28] to the 1-out-of-q case, and we present them in an explicit and formal form. The consistency and the security of our construction are proved according to these definitions. In particular, our construction uses the security notion of [27, 28], called perfect security against active $(\mathcal{A}, \mathcal{B})$-adversaries. However, it may provide stronger security than the works [27, 28], because the 1-out-of-q case allows to consider larger adversary structures $\mathcal{A}, \mathcal{B} \subseteq 2^{\mathcal{P}_n}$ for increasing values of q.

To understand this last claim, we analyze the particularly interesting case where adversaries are allowed to corrupt at most t servers for some $t < n/2$, or more generally, where $(\mathcal{A}, \mathcal{B})$ is an $\mathcal{R}_2$ pair of adversary structures satisfying $\binom{\mathcal{P}_n}{t} \subseteq \mathcal{A}, \mathcal{B}$. In the $q = 2$ case, by [27], we can choose $t = \lfloor 0.11n \rfloor$, so that there exists an ideal $\mathbb{F}_q$-LSSS Σ with access structure $\mathcal{A}$. In our construction, by choosing $q \geq n$ and $t = \lfloor n/2 - 1 \rfloor$, we can always take Σ as the $(t+1)$-threshold Shamir $\mathbb{F}_q$-LSSS. Hence, in this case, our construction is secure against adversaries that corrupt a minority of servers. This comes into contrast with [54], which prove that unconditionally secure single-use OT-combiners over small alphabets cannot exist when $t = n/2 - O(1)$. Letting the alphabet size vary allows to bridge this $O(1)$ gap.

As a research line in the direction of this work, we highlight how ideal $\mathbb{F}_2$-multilinear secret sharing schemes are used by Cascudo et al. in [27] to build a 1-out-of-2 OT combiner. By using multi-linear secret sharing schemes, the sender Alice creates a sharing of both messages (m_0, m_1) at the same time, which consists of $2n$ shares. In turn, our construction and [28] require that Alice generates independent sharings of each message, adding up to $4n$ shares. Extending their construction to the 1-out-of-q scenario would thus provide a reduction factor of q in the communication complexity of the sender, and also in the time complexity of the whole solution.

Bibliography

[1] Michael O. Rabin. How to exchange secrets with oblivious transfer, 2005. URL http://eprint.iacr.org/2005/187. Harvard University Technical, Report 81.

[2] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, June 1985.

[3] Andrew Chi-Chih Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA, 1982. IEEE Computer Society.

[4] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 20–31, New York, USA, 1988. ACM.

[5] Joe Kilian, Silvio Micali, and Rafail Ostrovsky. Minimum resource zero-knowledge proofs. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 545–546, New York, 1990. Springer.

[6] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 547–557, New York, 1990. Springer.

[7] Bill Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 119–135, Berlin, Heidelberg, 2001. Springer.

[8] Benny Chor, Oded Goldreich, and Eyal Kushilevitz. Private information retrieval. *Journal of the ACM*, pages 41–50, 1995.

[9] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC '99, pages 245–254, New York, USA, 1999. ACM.

[10] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. David & goliath oblivious affine function evaluation - asymptotically optimal building blocks for universally composable two-party computation from a single untrusted stateful tamper-proof hardware token. Cryptology ePrint Archive, Rep. 2012/135, 2012. URL https://eprint.iacr.org/2012/135.

[11] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 1–10, New York, USA, 1988. ACM.

[12] Benny Chor and Eyal Kushilevitz. A zero-one law for boolean privacy. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 62–72, New York, USA, 1989. ACM.

[13] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. *IACR Cryptology ePrint Archive*, 2010:153, 2010.

[14] Claude Crépeau and Joe Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science*, pages 42–52, 1988.

[15] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, pages 493–502, 12 1998.

[16] Rafael Dowsley, Jeroen van de Graaf, Jörn Müller-Quade, and Anderson C. A. Nascimento. Oblivious transfer based on the McEliece assumptions. In Reihaneh Safavi-Naini, editor, *Information Theoretic Security*, pages 107–117, Berlin, Heidelberg, 2008. Springer.

[17] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 554–571, Berlin, Heidelberg, 2008. Springer.

[18] Charles A. Asmuth and George R. Blakley. An efficient algorithm for constructing a cryptosystem which is harder to break than two other cryptosystems. *Computers & Mathematics with Applications*, 7(6):447–450, 1981.

[19] Yevgeniy Dodis and Jonathan Katz. Chosen-ciphertext security of multiple encryption. In Joe Kilian, editor, *Theory of Cryptography*, pages 188–209, Berlin, Heidelberg, 2005. Springer.

[20] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, March 1999.

[21] Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, April 2006. URL https://rfc-editor.org/rfc/rfc4346.txt.

[22] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 96–113, Berlin, Heidelberg, 2005. Springer.

[23] Remo Meier and Bartosz Przydatek. On robust combiners for private information retrieval and other primitives. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 555–569, Berlin, Heidelberg, 2006. Springer.

[24] Remo Meier, Bartosz Przydatek, and Jürg Wullschleger. Robuster combiners for oblivious transfer. In Salil P. Vadhan, editor, *Theory of Cryptography*, pages 404–418, Berlin, Heidelberg, 2007. Springer.

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

[26] Yuval Ishai, Hemanta K. Maji, Amit Sahai, and Jürg Wullschleger. Single-use OT combiners with near-optimal resilience. In *International Symposium on Information Theory*, pages 1544–1548. IEEE, 2014.

[27] Ignacio Cascudo, Ivan Damgård, Oriol Farràs, and Samuel Ranellucci. Resource-efficient OT combiners with active security. In *15th International Conference on Theory of Cryptography, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 461–486. Springer, 2017.

[28] Ignacio Cascudo, Ivan Damgård, Oriol Farràs, and Samuel Ranellucci. Server-aided two-party computation with minimal connectivity in the simultaneous corruption model. Cryptology ePrint Archive, Report 2014/809, 2014. URL https://eprint.iacr.org/2014/809.

[29] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Founding cryptography on oblivious transfer — efficiently. In *Proceedings of the 28th Annual Conference on Cryptology: Advances in Cryptology*, CRYPTO 2008, pages 572–591, Berlin, Heidelberg, 2008. Springer-Verlag.

[30] Bartosz Przydatek and Jürg Wullschleger. Error-tolerant combiners for oblivious primitives. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 461–472, Berlin, Heidelberg, 2008. Springer.

[31] Vinod Vaikuntanathan and Prashant Nalini Vasudevan. Secret sharing and statistical zero knowledge. In *Proceedings, Part I, of the 21st International Conference on Advances in Cryptology – ASIACRYPT 2015 - Volume 9452*, pages 656–680, New York, USA, 2015. Springer-Verlag, Inc.

[32] Amos Beimel and Yuval Ishai. On the power of nonlinear secret-sharing. *SIAM Journal on Discrete Mathematics*, 19(1):258–280, May 2005. ISSN 0895-4801.

[33] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, 2005.

[34] Sven Laur and Helger Lipmaa. A new protocol for conditional disclosure of secrets and its applications. In Jonathan Katz and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 207–225, Berlin, Heidelberg, 2007. Springer.

[35] Vladimir Kolesnikov, Ranjit Kumaresan, Mike Rosulek, and Ni Trieu. Efficient batched oblivious prf with applications to private set intersection. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 818–829, New York, USA, 2016. ACM.

[36] Michele Orrù, Emmanuela Orsini, and Peter Scholl. Actively secure 1-out-of-n OT extension with application to private set intersection. In *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, pages 381–396, 2017.

[37] Yehuda Lindell. How to simulate it - a tutorial on the simulation proof technique. In *Tutorials on the Foundations of Cryptography*, 2016.

[38] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[39] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 218–229, New York, USA, 1987. ACM.

[40] Claude Crépeau. Equivalence between two flavours of oblivious transfers. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pages 350–354, Berlin, Heidelberg, 1988. Springer.

[41] Gilles Brassard, Claude Crépeau, and Miklos Santha. Oblivious transfers and intersecting codes. *IEEE Transactions on Information Theory*, 42(6):1769–1780, 11 1996.

[42] Claude Crépeau, Gilles Brassard, and Jean-Marc Robert. Information theoretic reductions among disclosure problems. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)(FOCS)*, volume 00, pages 168–173, 10 1986.

[43] Wen-Guey Tzeng. Efficient 1-out-n oblivious transfer schemes. In David Naccache and Pascal Paillier, editors, *Public Key Cryptography*, pages 159–171, Berlin, Heidelberg, 2002. Springer.

[44] Carles Padró. Lecture notes in secret sharing. *IACR Cryptology ePrint Archive*, page 674, 2012. URL http://eprint.iacr.org/2012/674.

[45] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

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

[47] Oriol Farras. *Multipartite Secret Sharing Schemes*. PhD thesis, Universitat Politècnica de Catalunya, 7 2010. URL http://hdl.handle.net/2117/94536.

[48] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan, Part III*, 72 (9):56–64, 1989.

[49] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–415, 10 2016.

[50] Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 441–471. Springer International Publishing, 2019.

[51] Amos Beimel and Naty Peter. Secret-sharing from robust conditional disclosure of secrets. Cryptology ePrint Archive, Report 2019/522, 2019. URL https://eprint.iacr.org/2019/522.

[52] Oriol Farràs, Jordi Ribes-González, and Sara Ricci. Local bounds for the optimal information ratio of secret sharing schemes. *Designs, Codes and Cryptography*, 87(6):1323–1344, 2019.

[53] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. URL https://eprint.iacr.org/2000/067.

[54] Ignacio Cascudo Pueyo, Ronald Cramer, and Chaoping Xing. Bounds on the threshold gap in secret sharing and its applications. *IEEE Transactions on Information Theory*, 59(9):5600–5612, 2013.