Endemic Oblivious Transfer

Daniel Masny and Peter Rindal

VISA Research

Abstract

Oblivious Transfer has played a crucial role in the design of secure multi party computation. Nevertheless, there are not many practical solutions that achieve simulation based security and at the same time instantiable based on different assumptions.

In this work, we consider a simulation based security notion that we call endemic security. We show how to construct highly efficient oblivious transfer in the random oracle model that achieves endemic security under a wide range of assumptions, among them DDH, CDH, LWE and coding based assumptions. We construct a secure oblivious transfer based on DDH that takes only a single communication round which allows significant performance gains. We also instantiate our oblivious transfer with the Crystals.Kyber key agreement. Our implementation shows that both instantiations can be computed in under one millisecond.

Further, we revisit, correct and improve existing oblivious transfer extension techniques. We provide an implementation of an oblivious transfer extension protocol in the ideal cipher model that is actively secure, processing up to 23 million OTs per second and up to 10 times faster than previous secure implementations. We also show that our framework can compute endemically secure OT extension and the base OTs in just two rounds.

1 Introduction

An oblivious transfer (OT) [Rab81, EGL82] is a cryptographic primitive often used in the context of secure multi party computation, which allows to preserve the privacy during a joint computation. Among others, it solves the task of securely distributing cryptographic keys for garbled circuits, which can be seen as encrypted programs. The combination of garbled circuits and oblivious transfer gives a generic solution for securely computing any functionality between two parties [Yao82, Yao86, Kil88, IPS08, IKO+11] and multiple parties [CvT95, BL18, GS18].

In an OT, a sender and a receiver interact in a protocol and at the end of the protocol, the sender outputs two messages s_0, s_1 while the receiver outputs b, s_b for choice bit b. Security asks that the sender does not learn b and the receiver does not learn s_{1-b}. It is known that an OT implies key exchange and can be constructed from special types of public key encryption (PKE) [GKM+00, PVW08, FMV18], generalizations of dual mode PKE [GIS18] or certified trapdoor permutations [ORS15]. Though, all of these solutions come with some drawbacks when it comes to practical deployment. They either only achieve a weak security notion [GKM+00] or lack efficiency due to a special type of commitment protocol [Kil92, ORS15, FMV18] or dual-mode cryptosystem [PVW08, GIS18], which is less efficient than standard PKE and only known from DDH, QR and LWE with weaker parameter choices[1]. There exist also solutions tailored to specific assumptions.

[1] Peikert et al. require SIVP hardness for approximation factor $\tilde{O}(n^{3.5})$ while Regev's PKE [Reg05] only requires $\tilde{O}(n^{1.5})$.

Naor and Pinkas [NP01] constructed OTs from the DDH assumption and Brakerski and Döttling [BD18] from the LWE assumption that requires similar parameter choices as Peikert et al [PVW08].

In practice, a common approach is to use a critical amount of OTs in the random oracle model (ROM) [BR93] and then extend the amount of OTs to the desired amount of OTs using OT extension [Bea96, IKNP03, OOS17, ALSZ15, KOS15]. A random oracle is an ideal hash function that usually is instantiated with a concrete hash function in the implementation. The ROM brings major efficiency improvements and is therefore very common for practical cryptographic constructions, even though it might bring potential security weaknesses [CGH98].

In the ROM, Bellare & Micali [BM90] constructed OT based on the CDH assumption. Chou & Orlandi [CO15] claimed a more efficient OT construction which was proven with some caveats under the GapDH assumption [HL17]. Hauck & Loss improved the construction to base it on the CDH assumption [HL17]. Barreto et al. [BDD+17] constructed an CDH based OT in the global random oracle model.

A drawback of the more efficient constructions of Chou & Orlandi, Hauck & Loss, and Barreto et al. is that they require three or more rounds. The former also suffers technical issues with the ability to extract the input of the receiver [CO15, BPRS17]. Further, unlike the more generic constructions based on PKE, they are tailored to specific assumptions. A more generic construction for OT in the ROM would be preferable since it allows an easier transition to different assumption like LWE or LPN, which unlike CDH and DDH are assumed to offer security in presence of quantum computers. In this work, we therefore want to focus on the question:

How to construct a versatile, highly efficient and fully secure OT in the ROM?

For efficiency, we ask for a minimal round complexity, low computational complexity and compatibility with OT extension techniques.

1.1 Our Contribution

We start with a basic security definition which has previously been considered by Garg, Ishai and Srinivasan [GIS18] as OT correlations functionality. We call this security *endemic* security and denote an OT that is endemically secure with endemic OT. Endemic security allows to achieve a minimal round complexity, also denoted as non-interactive OT [BM90, GIS18] as well as analyze the security of optimized existing OT and OT extension protocols.

In Section 3, we compare endemic security notion with other notions and show that an endemic OT can efficiently be transformed such the other considered security notions are achieved but potentially at the cost of a higher round complexity. We also show that only endemic security permits a one-round or non-interactive OT.

In Section 4, we give a construction in the ROM that transforms any two message key agreement protocol, where the distribution of one of the messages is computationally close to uniform, into an endemically secure two message 1-out-of-n oblivious transfer[2]. Further, if the key agreement protocol is a one-round protocol, we obtain a one-round endemic OT. This implies that we get a one-round endemic OT from DDH, CDH and two round endemic OT from LWE, LPN, McEliece and Subset Sum. We emphasize that [GIS18] construct a one-round UC secure OT from LWE, DDH, QR in the CRS model, while we focus on stand-alone security in the random oracle model in favor of efficiency.

[2] In Appendix C, we show how the framework can be adapted to obtain an endemically secure $(n-1)$-out-of-n OT.

In Section 5, we show that our construction is compatible with OT extension techniques. Concretely, we show that endemic OTs can be extended to a larger amount of endemic OTs using only one additional round. This allows us to obtain $\mathsf{poly}(\kappa)$ OTs using only $O(\kappa)$ public key operations in only two rounds. We revisit the OT extension protocol of Keller et al., Orrù et al. and Asharov et al. [KOS15, OOS17, ALSZ17] under endemic security. It turns out that its uniform message security can be fully broken. We point out attacks and provide fixes such that classical, uniform and endemic security can be obtained. Finally, we observe that most OT extension protocols are implemented [Rin, Kel, WMK16] using an ideal cipher in place of a random oracle. However, these implementations have no security proofs and we show that they too can be fully broken. We give new protocols and proofs in the ideal cipher model which allows a 10 times speed up on the ROM when implemented.

In Section 6, we implement our construction based on the Diffie-Hellman key exchange and the Module LWE (MLWE) based Kyber key encapsulation [SAB+17]. We emphasize that it can also be instantiated with many of the other NIST post-quantum standardisation candidates and is to the best of our knowledge the first implementation of a quantum resistant OT.

1.2 Our Techniques

Endemic Security. When defining malicious security of an OT, one defines an ideal functionality $\mathcal{F}_{\mathsf{OT}}$. An OT is called secure, if for any adversary against the OT scheme, there exists an adversary interacting with $\mathcal{F}_{\mathsf{OT}}$ producing the same output. Classically, $\mathcal{F}_{\mathsf{OT}}$ either receives the OT strings s_0, s_1 as input from the sender or samples them uniformly at random and outputs them to the sender. But there are also OTs where the receiver can determine the OT strings or even both parties could influence how the OT strings are generated. We distinguish four main security notions.

Uniform Message Security: The ideal functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ samples the OT strings uniformly and outputs them to sender and one to the receiver.

Sender Chosen Message Security: The ideal functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ receives the OT strings from the sender and outputs one of the strings to the receiver.

Receiver Chosen Message Security: The ideal functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ receives one of the OT strings from the receiver, samples the other one uniformly at random and outputs the strings to the sender.

Endemic Security If the sender is malicious, it chooses both strings. If the receiver is malicious, it chooses one of the strings. All strings that are not chosen yet, are sampled uniformly by the functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. The sender obtains both strings and the receiver obtains one.

Notice that endemic security gives the weakest security guarantees, no matter whether the receiver or the sender is malicious, the malicious party can always determine the distribution of the OT messages. Uniform message security gives very strong security guarantees since a malicious party can never influence the distribution.

Relations Between Security Notions. We show on one hand that an OT with uniform message security is also secure with respect to all other security notions. On the other hand, uniform, sender and receiver chosen message security imply endemic security. Still, there are very simple transformations from an endemically secure OT to an OT that achieves any of the other security notions. Though we remark that uniform message security implies and therefore requires a secure coin tossing protocol. In Figure 1 we give an overview over these implications and transformations.

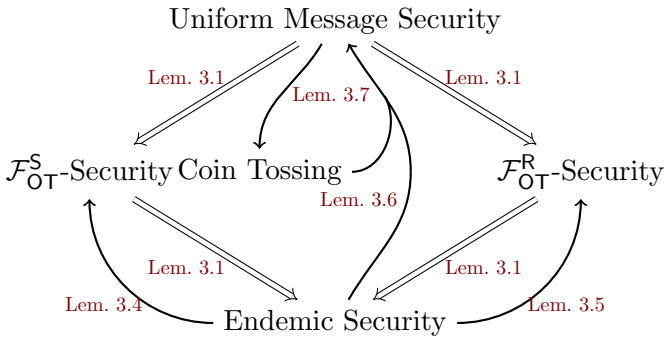

Figure 1: The figure depicts the different security notions of OT and their relations. $A \Rightarrow B$ denotes the implication. $A \to B$ denotes that there is an efficient transformation.

We also show that endemic OT is weaker than the other notions but at the same time this allows a minimal round complexity of a single round. More precisely, we show that there is no one round OT that achieves sender or receiver chosen message security.

From Key Agreement to OT A common strategy to construct OT from PKE is to use a PKE where the public keys form a group [PVW08, BDD+17], which we will denote with $(\mathcal{G}, \oplus)$. By giving a challenge c and forcing the receiver to generate two public keys pk_0 and pk_1 s.t. $c = \mathsf{pk}_0 \oplus \mathsf{pk}_1$, he can intuitively only decrypt ciphertexts with respect to one of them. But this does not actually follow from the standard notion of PKE since an adversary could generate pk_0 and pk_1 jointly given c. It requires a dual-mode cryptosystem [PVW08, GIS18] that is tailored towards this property. Dual-mode cryptosystems are known from DDH, QR and LWE [PVW08] but is not clear how to extend these results to other assumptions.

Another approach [ORS15, FMV18] uses specific commitment protocols which forces the receiver to commit to a public key before c is known. The drawback of this approach is that it requires four rounds and the known constructions of such a commitment protocol are not efficient [Kil92, ORS15].

We propose a different solution that uses a novel and simple technique to leverage the power of a random oracle. Rather than choosing two public keys, we ask the receiver to generate two strings r_0, r_1 in $\mathcal{G}$. From these strings a sender can generate the public keys $\mathsf{pk}_0 = r_0 \oplus \mathsf{H}(r_1)$, $\mathsf{pk}_1 = r_1 \oplus \mathsf{H}(r_0)$ under which he can encrypt the two OT messages s_0, s_1. In the actual protocol, the receiver can program $r_b \in \{r_0, r_1\}$ to a public key for his choice of $b \in \{0, 1\}$. He samples r_{1-b} and sets $r_b = \mathsf{pk} \ominus \mathsf{H}(r_{1-b})$.

This technique also allows to extract s_0, s_1 from a malicious sender by programming the random oracle such that secret keys for both, pk_0 and pk_1 are known. Further, one can extract b from a malicious receiver by programming the random oracle as well. Intuitively, a malicious receiver needs to query either r_0 or r_1 first. His choice will determine r_{1-b}, since all following random oracle queries q can be programmed such that $\mathsf{H}(q) = \mathsf{pk}' - r_{1-b}$ for a public key pk'. If a malicious adversary can learn s_{1-b}, he will decrypt a ciphertext for $\mathsf{pk}_{1-b} = r_{1-b} \oplus \mathsf{H}(q) = \mathsf{pk}'$ and be able to break the PKE scheme.

We optimize the protocol further by using a key agreement instead of a PKE scheme. In many settings, the OT messages don't need to be chosen, it is sufficient if they are pseudorandom. Hence, no ciphertext needs to be generated, only the exchanged keys need to be computed. This save in some settings a communication round, e.g. in case of the Diffie-Hellman key exchange [DH76].

In the main body of this paper, we only consider stand-alone security. We show UC security for some settings of our protocol in Appendix E.

Secure OT Extension. In Section 5 we explore the rich implications endemic security has on efficient 1-out-of-N OT extension along with presenting three new attacks and fixes of existing OT extension protocols[KOS15, OOS17] with Uniform Message security[3]. These protocols are derived from the seminal black-box protocol of Ishia, Kilian, Nissim and Petrank[IKNP03]. We note that in all cases the Sender Chosen Message variant of these protocols[IKNP03, KOS15, OOS17] are secure. The functionality of 1-out-of-N OT extension allows $n_{\mathcal{C}} \approx \kappa$ instances of 1-out-of-2 OTs to be transformed into $m = \mathsf{poly}(\kappa)$ instances of 1-out-of-N OTs. There are several advantages of this transformation. 1) m can be polynomial times larger than $n_{\mathcal{C}}$. 2) Only symmetric key cryptography is required which provides a larger performance improvement. 3) In some cases N can be exponential in the security parameter κ which we indicate with the use of capital N.

The 1-out-of-2 OTs that are being transformed are referred to as *base OTs*. Existing protocols [IKNP03, ALSZ15, KOS15, OOS17] have called for the use of base OTs with the sender chosen message security notion, e.g. $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$. However, we show that this requirement can be relaxed to allow the base OTs to only achieve endemic security. In both cases ($\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ or $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ base OTs) the OT extension protocol outputs messages that satisfy the endemic security notion. Traditional OT extension protocols, e.g. [IKNP03, ALSZ15, KOS15], then apply a simple transform ($\Pi_{1,N}^{\mathsf{S}}$, Figure 4) to realize the sender chosen message functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$. This observation suggests that more efficient OT extension can be realized by replacing Sender Chosen Message base OTs with Endemic OTs, e.g. our protocol. The authors of [KOS15, OOS17] suggest that the $\Pi_{1,N}^{\mathsf{S}}$ transform can be removed and resulting protocol would satisfy the uniform message security notion, but in Section 5.1 we show this to not be the case.

In particular, Section 5.1 detail three attacks where the first allows a malicious party to bias the OT messages that they output while the second and third attacks succeed even when base OTs with uniform message security are used. In all cases, the ability to bias the messages violates the ideal functionality which samples them uniformly at random. Therefore, we show that the protocol only achieves Endemic security.

We note that many protocols that utilize Uniform Message security can likely tolerate the weaker notion of Endemic security, e.g. [RR17a, RR17b]. However, other protocols such as the set inclusion protocol of [OOS17, Figure 5] are insecure[4] when Uniform Message security is not satisfied.

Uniform message security can be achieved from endemic OT extension in several ways. One solution is a black-box transformation $\Pi_{1,N}^{\mathsf{U}}$ (Figure 6) which lifts an OT protocol with endemic security to satisfy uniform message security. However, this would require additional rounds and significant communication. We demonstrate an alternative solution which replaces the base OTs with a protocol that satisfies uniform message, uniform selection security $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}u}$ and prove that this yields an OT extension protocol with uniform message security with minimal need to modify the extension protocol. More generally, Figure 2 shows the relation between different base OT security notions and the resulting OT extension security.

For example, the protocols of [IKNP03, ALSZ15, KOS15] perform

$$\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}} \xrightarrow{\Pi^{\text{ext}}} \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}} \xrightarrow{\Pi_{1,2}^{\mathsf{S}}} \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$$

[3] [KOS15, OOS17] refer to uniform OT as random OT $\mathcal{F}_{\mathsf{ROT}}^{m,\kappa}$

[4] The sender set all the OT messages to be the same value and force the receiver to conclude their item is in the sender's set.

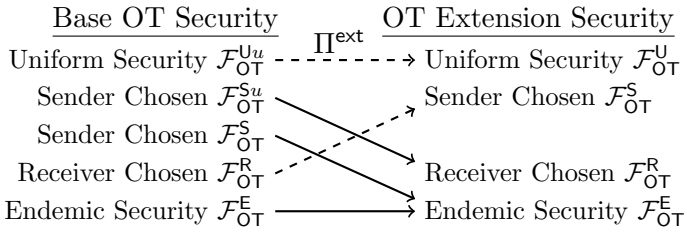

Figure 2: The figure shows the implications of the base OT security (Definition 2.6) when Π^{ext} from Figure 9 or a slight variation (dashed line) is applied. $\mathcal{F}_{\mathsf{OT}}^{*u}$ denotes $\mathcal{F}_{\mathsf{OT}}^{*}$ security where the receiver's selection is uniformly sampled by the functionality Definition 2.4 ,A.13.

where Π^{ext} is their respective extension protocol up to hashing.

In addition, Section 5.3 details new OT extension protocols that can efficiently be realized in the ideal cipher model. These protocol are inspired by existing implementation [Rin, Kel, WMK16] but are provably secure. Unfortunately, these existing implementation improperly apply the ideal cipher model which could be leverage by a malicious receiver to fully break *all* OT messages.

Implementation. We instantiate our OT protocol with the Diffie-Hellman key exchange. We show how the security loss can be reduced using the random self-reducibility of the DDH assumption. We also instantiate it based on the Kyber key exchange [BDK+17, SAB+17]. This is a proof of concept instantiation that shows that our framework is very agile in terms of assumptions and allows to obtain post-quantum security efficiently.

We give implementations and benchmarks for the two OT protocols as well as five implementations of OT extension protocols. Both our OT protocols can perform an OT in one millisecond. We compare our results with the OTs of Chou & Orlandi [CO15] and Naor & Pinkas [NP01]. In a WAN network setting we observe that our one-round DDH protocol is the fastest, requiring 110ms. The next fastest framework was the two round protocol of [CO15], requiring 210ms. In addition to being faster than [CO15] in the WAN setting, our protocols achieve full simulation based security without performing additional rounds as [CO15] requires.

Our communication optimized OT extension protocol achieving endemic security requires two rounds of communication and can processes 2 million OTs per second. This is on par with a throughput optimized version of [KOS15, Rin] in the LAN setting. In the WAN setting our protocol becomes 2.5 times faster than [KOS15] (which achieves stronger security). Our computation optimized protocol in the ideal cipher model can process 23 million OTs per second in the LAN over the course of 5 rounds and achieves full uniform message security.

2 Preliminaries

2.1 Notation

κ denotes the security parameter. For $n \in \mathbb{N}$, $[n] := \{1, \ldots, n\}$ and $(s_j)_{j \neq i}$ denotes ther ordered set $(s_j)_{j \in [n] \setminus \{i\}}$. We use $\Pi^{\mathsf{A},\mathsf{B}}$, Π when A and B are clear from the context, to denote a protocol between two parties A and B. $\langle \mathsf{A}, \mathsf{B} \rangle$ denotes as the transcript of the protocol, which consists of all the messages sent between them. We use $(\mathsf{A}(a), \mathsf{B}(b))_{\Pi}$ to denote the joint output distribution of A and B when interacting in protocol Π with inputs a and b.

Definition 2.1 (Random Oracle)**.** *A* random oracle *over a set of domains and an image is a*

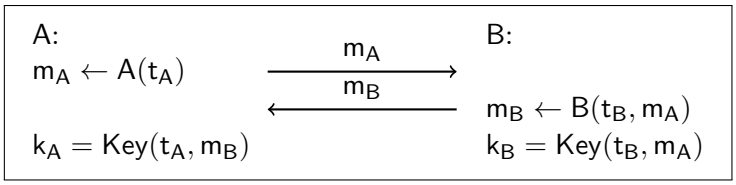

Figure 3: The figure shows a key agreement protocol between parties A and B with random tapes t_A and t_B. Correctness requires $\mathsf{k}_\mathsf{A} = \mathsf{k}_\mathsf{B}$.

collection of functions H *that map an element* q *within one of the domains to a uniform element* $\mathsf{H}(q)$ *in the image.*

Definition 2.2 (Ideal Cipher)**.** *An* ideal cipher *over a set of tuples of domains and an image is a collection of functions* π *such that for any element* k *of the first domain,* π_k *is a permutation that map an element* q *within the second domain to a uniform element* $\pi_\mathsf{k}(q)$ *in the image.*

2.2 Key Agreement

Definition 2.3 ((Two Message) Uniform Key Agreement (UKA))**.** *Let* $\mathcal{G}$ *be a group. We call a protocol* Π *between two ppt parties* A *and* B *(two message) uniform key agreement if* A *first sends a message* $\mathsf{m}_\mathsf{A} \in \mathcal{G}$ *to* B *and* B *responds with a final message* m_B *and in the end, both establish a common key* k *(see Figure 3) using a key establishing algorithm* Key*. Further, we require three properties:*

Correctness:

$$\Pr[\mathsf{k}_\mathsf{A} = \mathsf{Key}(\mathsf{t}_\mathsf{A}, \mathsf{m}_\mathsf{B}) = \mathsf{Key}(\mathsf{t}_\mathsf{B}, \mathsf{m}_\mathsf{A}) = \mathsf{k}_\mathsf{B}] \geq 1 - \mathsf{negl},$$

where $\mathsf{t}_\mathsf{A} \leftarrow \{0,1\}^*$*,* $\mathsf{t}_\mathsf{B} \leftarrow \{0,1\}^*$*,* $\mathsf{m}_\mathsf{A} \leftarrow \mathsf{A}(\mathsf{t}_\mathsf{A})$ *and* $\mathsf{m}_\mathsf{B} \leftarrow \mathsf{B}(\mathsf{t}_\mathsf{B})$*.*

Key-Indistinguishability: *For any ppt distinguisher* D *and any polynomial size auxiliary input* z*,*

$$|\Pr[\mathsf{D}(z, \langle \mathsf{A}, \mathsf{B} \rangle, \mathsf{k}) = 1] - \Pr[\mathsf{D}(z, \langle \mathsf{A}, \mathsf{B} \rangle, u) = 1]| = \mathsf{negl},$$

where k *is the established key between* A *and* B *and* u *is a uniform element from the key domain.*

Uniformity: *For any ppt distinguisher* D *and any polynomial size auxiliary input* z*,*

$$|\Pr[\mathsf{D}(z, \mathsf{m}_\mathsf{A}) = 1] - \Pr[\mathsf{D}(z, u) = 1]| = \mathsf{negl},$$

where u *is a uniform element from* $\mathcal{G}$ *and* $\mathsf{m}_\mathsf{A} \leftarrow \mathsf{A}(\mathsf{t}_\mathsf{A})$*.*

When A *and* B *can send their messages concurrently, we call it a one-round* UKA*.*

In Appendix A.1, we define multi-instance security notions when executing multiple instances of a key agreement. All the considered notions follow from the standard notions from above, but potentially with a polynomial security loss.

2.3 Oblivious Transfer

Definition 2.4 (Ideal k-out-of-n Oblivious Transfer)**.** *An* ideal k-out-of-n oblivious transfer *is a functionality that interacts with two parties, a sender* S *and a receiver* R*.* R *sends a set* $\mathbb{S} \subseteq [n]$ *of size* k *to the functionality.*

The functionality is publicly parameterized by one of the following message sampling methods:

SENDER CHOSEN MESSAGE: S *sends the messages* $(m_i)_{i\in[n]}$ *to the functionality who sets* $s_i := m_i$*.*

RECEIVER CHOSEN MESSAGE: R *sends the messages* $(m_i)_{i\in[k]}$ *to the functionality who sets* $s_{\mathbb{S}_i} := s'_i$ *for* $i \in [k]$ *and uniformly samples* $s_i \leftarrow \{0,1\}^\ell$ *for* $i \in [n] \setminus \mathbb{S}$*.*

UNIFORM MESSAGE: *The functionality uniformly samples* $(s_1, ..., s_n) \leftarrow \{0,1\}^{\ell\times n}$*.*

ENDEMIC: *If* S *is corrupt,* S *sends the messages* $(s_i)_{i\in[n]}$ *to the functionality. If* R *is corrupt,* R *sends the messages* $(s'_i)_{i\in[n]}$ *to the functionality who sets* $s_{\mathbb{S}_i} := s'_i$ *for* $i \in [k]$*. All remaining* s_i *for* $i \in [n]$ *are uniformly sampled* $s_i \leftarrow \{0,1\}^\ell$ *by the functionality.*

As specified by the message sampling method, the functionality constructs messages $(s_i)_{i\in[n]}$*. Thereafter, the functionality sends* $(s_i)_{i\in[n]}$ *to* S *and* $(s_i)_{i\in\mathbb{S}}$ *to* R*. We denote the ideal functionalities for sender chosen, receiver chosen, uniform message and endemic as* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*, respectively.*

Remark 2.5. *We generalize this definition for the case where* n *can be exponential. In addition, we consider the case when the set* $\mathbb{S}$ *is sampled uniformly by the functionality. We call this* uniform selection *as opposed to* receiver selection*. In this case, we denote the analogous oracles for* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ *as* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}u}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}u}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}u}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}u}$*, respectively. See Definition A.13.*

In our definition of OT, we use the simplified UC security definition that is sufficient for full UC security [CCL15]. We also use this definition for our stand-alone security analysis in the main body of this paper, but in that case, we allow adversary $\mathcal{A}$' to rewind adversary $\mathcal{A}$.

Definition 2.6 (k-out-of-n Oblivious Transfer ($\mathsf{OT}_{k,n}$))**.** *We call a protocol* Π *between two ppt parties, a sender* S *and a receiver* R*, a* k-out-of-n oblivious transfer *if at the end,* S *outputs* n *strings* $(s_i)_{i\in[n]}$ *and* R *outputs* $(s_i)_{i\in\mathbb{S}}$ *and a set* $\mathbb{S} \subset [n]$ *s.t.* $|\mathbb{S}| = k$*. For security, we require two properties with respect to a functionality* $\mathcal{F}_{\mathsf{OT}}$*.*

Security Against a Malicious Sender: *For any ppt adversary* $\mathcal{A}$*, there exists a ppt adversary* $\mathcal{A}$*' such that for any ppt environment* D *and any polynomial size auxiliary input* z

$$|\Pr[\mathsf{D}(z, (\mathcal{A}, \mathsf{R})_\Pi) = 1] - \Pr[\mathsf{D}(z, (\mathcal{A}', \mathcal{F}_{\mathsf{OT}})) = 1]| = \mathsf{negl},$$

where all algorithms receive input 1^κ*.* R *additionally receives input* $\mathbb{S}$*.*

Security Against a Malicious Receiver: *For any ppt adversary* $\mathcal{A}$*, there exists a ppt adversary* $\mathcal{A}$*' such that for any ppt distinguisher* D *and any polynomial size auxiliary input* z

$$|\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_\Pi) = 1] - \Pr[\mathsf{D}(z, (\mathcal{F}_{\mathsf{OT}}, \mathcal{A}')) = 1]| = \mathsf{negl},$$

where all algorithms receive input 1^κ*.*

We distinguish four different security notions.

Uniform Message Security: *The OT is secure with respect to* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$*, i.e.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$*-secure.*

Sender Chosen Message Security: *The OT is secure with respect to* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$*, i.e.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$*-secure.*

Receiver Chosen Message Security: *The OT is secure with respect to* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*, i.e.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*-secure.*

Endemic Security: *The OT is secure with respect to* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*, i.e.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*-secure.*

Remark 2.7. *It is important to notice that we distinguish the functionality an OT provides from the ideal functionality for which the OT is secure. Here,* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$*-security is the strongest security definition since a malicious party cannot tweak the distribution of the strings* $(s_i)_{i\in[n]}$*. Endemic security gives the weakest security guarantees since in both cases, the malicious receiver and malicious sender case, the adversary can potentially choose the strings* $(s_i)_{i\in\mathbb{S}}$*.*

Remark 2.8. *In the following, we assume that all messages from a sender or a receiver also contain a session identifier* sid *and that for every new session between a sender and a receiver, they receive access to a fresh random oracle that is unique to that session (local random oracle).*

3 Relations Between OT Security Notions

We show now how endemic security relates to the other security notions of uniform, receiver and sender chosen message security. For an overview, see Figure 1.

Lemma 3.1. *Let the distribution of OT strings be efficiently sampleable. Then* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$*-security implies* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *as well as* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*-security.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *or* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*-security imply* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*-security.*

Proof. In the first step, we show that uniform message security implies sender chosen message security and receiver chosen message security implies endemic security. These two implications result from the same simple fact that a malicious sender interacting with the ideal OT is easier to construct when it can choose the OT strings than when it receives the strings from the ideal OT. The following claim formalizes this fact.

Claim 3.2. *Let* Π *be an OT secure against a malicious sender with respect to an ideal OT* $\mathcal{F}_{\mathsf{OT}}^{*}$ *that sends the OT strings* $(s_i)_{i\in[n]}$ *to the sender, i.e. functionality* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ *and* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*, and the distribution of* $(s_i)_{i\in[n]}$ *is efficiently sampleable. Then* Π *is also secure against a malicious sender with respect to ideal OT* $\mathcal{F}_{\mathsf{OT}}$*, which receives the OT strings* $(s_i)_{i\in[n]}$ *from the sender, i.e functionality* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *and* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*.*

Proof. We show that if there is an adversary that breaks the security against a malicious security with respect to ideal OT $\mathcal{F}_{\mathsf{OT}}$ then there is also an adversary that breaks the security with respect to $\mathcal{F}_{\mathsf{OT}}^{*}$. More precisely, if there is a ppt adversary $\mathcal{A}_1$ such that for any ppt adversary $\mathcal{A}_1'$ there exists a ppt distinguisher D_1 and a polynomial size auxiliary input z with

$$|\Pr[\mathsf{D}_1(z,(\mathcal{A}_1,\mathsf{R})_\Pi)=1]-\Pr[\mathsf{D}_1(z,(\mathcal{A}_1',\mathcal{F}_{\mathsf{OT}}))=1]|=\epsilon,$$

where all algorithms receive input 1^κ and R additionally receives input $\mathbb{S}$. Then there is also a ppt adversary $\mathcal{A}_2$ such that for any ppt adversary $\mathcal{A}_2'$ there exists a ppt distinguisher D_2 and a polynomial size auxiliary input z with

$$|\Pr[\mathsf{D}_2(z,(\mathcal{A}_2,\mathsf{R})_\Pi)=1]-\Pr[\mathsf{D}_1(z,(\mathcal{A}_2',\mathcal{F}_{\mathsf{OT}}^{*}))=1]|=\epsilon,$$

where all algorithms receive input 1^κ and R additionally receives input $\mathbb{S}$.

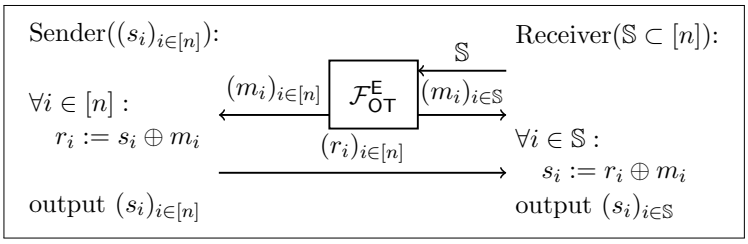

Figure 4: Sender chosen OT protocol $\Pi_{k,n}^{\mathsf{S}}$ in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ hybrid (Definition 2.4). For all $i \in [n]$, r_i, m_i and s_i are in $\{0,1\}^\ell$.

We set $\mathcal{A}_2 := \mathcal{A}_1$ and $\mathsf{D}_2 := \mathsf{D}_1$. Further, for any $\mathcal{A}_2'$, there is an $\mathcal{A}_1'$ such that the distribution of $(\mathcal{A}_2', \mathcal{F}_{\mathsf{OT}}^*)$ is identical with the distribution $(\mathcal{A}_1', \mathcal{F}_{\mathsf{OT}})$. This follows from the fact that $\mathcal{A}_1'$ could choose the OT strings $(s_i)_{i\in[n]}$ from the same distribution as $\mathcal{F}_{\mathsf{OT}}^*$ does and otherwise follow the description of $\mathcal{A}_2'$. Since D_1 is successful for any $\mathcal{A}_1'$ it will be also for any $\mathcal{A}_2'$, which can be seen as a subset of the set of all ppt adversaries $\mathcal{A}_1'$. □

The remaining two implications, from uniform security to receiver chosen message security and from sender chosen message security to endemic security follow in a similar fashion. Again it is easier to construct a malicious receiver interacting with the ideal OT when he can choose the OT strings rather than receiving them from the ideal OT.

Claim 3.3. *Let Π be an OT secure against a malicious receiver with respect to an ideal OT $\mathcal{F}_{\mathsf{OT}}^*$ that sends the learned OT strings $(s_i)_{i\in\mathbb{S}}$ to the receiver, i.e. functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$, and the distribution of $(s_i)_{i\in\mathbb{S}}$ is efficiently sampleable. Then Π is also secure against a malicious sender with respect to ideal OT $\mathcal{F}_{\mathsf{OT}}$, which receives the OT strings $(s_i)_{i\in\mathbb{S}}$ from the receiver, i.e. $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$.*

Proof. The proof is basically identical to the proof of Claim A.11. Again, the set of all ppt $\mathcal{A}_2'$ is a subset of the set of all ppt $\mathcal{A}_1'$ and identical with the set of all $\mathcal{A}_1'$ that sample $(s_i)_{i\in\mathbb{S}}$ from the same distribution as when sent by $\mathcal{F}_{\mathsf{OT}}^*$. □

□

Even though endemic security is implied by all the other security notions and could be seen as the weakest, it is still sufficient to obtain any of the other notions by using very simple transformations. In the following lemmas we show these transformations and sketch their security.

Lemma 3.4. *$\Pi_{k,n}^{\mathsf{S}}$ of Figure 4 realizes the sender chosen message ideal OT $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ (Definition 2.4) with unconditional security in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ hybrid.*

Proof. We construct a new adversary $\mathcal{A}_1'$ which interacts with functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and produces an identical output to $\mathcal{A}_1$. $\mathcal{A}_1'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and R in $\Pi_{k,n}^{\mathsf{S}}$ while running $\mathcal{A}_1$. $\mathcal{A}_1'$ receives the endemic OT strings $m_1, ..., m_n$ from $\mathcal{A}_1$, along with the strings $r_1, ..., r_n$. $\mathcal{A}_1'$ extracts the OT strings of $\mathcal{A}_1$ as $s_i := r_i \oplus m_i$. $\mathcal{A}_1'$ sends $(s_i)_{i\in[n]}$ to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and outputs whatever $\mathcal{A}_1$ outputs.

Observe that the output of the honest receiver is identical when $\mathcal{A}_1$ interacts with R in $\Pi_{k,n}^{\mathsf{S}}$ and when $\mathcal{A}_1'$ interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$.

Now consider a corrupt receiver $\mathcal{A}$. We construct a new adversary $\mathcal{A}_2'$ which interacts with the functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and produces an identical output distribution to $\mathcal{A}_2$. $\mathcal{A}_2'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and S in $\Pi_{k,n}^{\mathsf{S}}$ while running $\mathcal{A}_2$. $\mathcal{A}_2'$ receives the set $\mathbb{S} \subset [n]$ of size k and the endemic OT strings

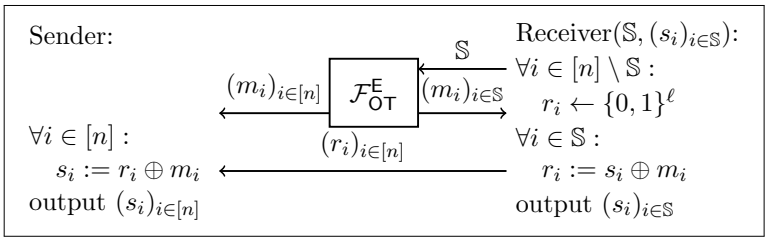

Figure 5: Receiver chosen OT protocol $\Pi_{k,n}^{\mathsf{R}}$ in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ hybrid (Definition 2.4). For all $i \in [n]$, r_i, m_i and s_i are in $\{0,1\}^\ell$.

$(m_i)_{i\in\mathbb{S}}$ from $\mathcal{A}_2$. $\mathcal{A}_2'$ sends $\mathbb{S}$ to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and receives $(s_i)_{i\in\mathbb{S}}$ in response. $\mathcal{A}_2'$ computes $r_i := s_i \oplus m_i$ for $i \in \mathbb{S}$ and uniformly samples $r_i \leftarrow \{0,1\}^\ell$ for $i \in [n] \setminus \mathbb{S}$. $\mathcal{A}_2'$ sends $r_1, ..., r_n$ to $\mathcal{A}_2$ and outputs whatever $\mathcal{A}_2$ outputs.

The transcripts of $\mathcal{A}_2$ in these two interactions are identical except for $(r_i)_{i\in[n]\setminus\mathbb{S}}$. Observe that in the real interaction for $i \in [n] \setminus \mathbb{S}$, $r_i := s_i \oplus m_i$, where m_i is sampled uniformly at random by functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and is independent of the transcript of $\mathcal{A}_2$ (conditioned on r_i). Therefore sampling r_i directly induces an identical distribution.

Therefore, any distinguishing advantage $\mathcal{A}_1$ or $\mathcal{A}_2$ produces in protocol $\Pi_{k,n}^{\mathsf{S}}$ implies that $\mathcal{A}_1'$ or $\mathcal{A}_2'$ would produce the same advantage against the instantiation of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$, i.e. negl. □

Lemma 3.5. *$\Pi_{k,n}^{\mathsf{R}}$ of Figure 5 realizes the receiver chosen message ideal OT $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ (Definition 2.4) with unconditional security in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ hybrid.*

Proof. First let us consider any corrupt sender $\mathcal{A}_1$. We construct a new adversary $\mathcal{A}_1'$ which interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and produces an identical output distribution as $\mathcal{A}_1$. $\mathcal{A}_1'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and R in $\Pi_{k,n}^{\mathsf{R}}$ while running $\mathcal{A}_1$. $\mathcal{A}_1'$ receives the endemic OT strings $m_1, ..., m_n$ from $\mathcal{A}_1$. $\mathcal{A}_1'$ invokes $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ as the sender and receives $s_1, ..., s_n$ in response. $\mathcal{A}_1'$ sends $r_1, ..., r_n$ to $\mathcal{A}_1$ where $r_i := m_1 \oplus s_i$ and outputs whatever $\mathcal{A}_1$ outputs.

The transcripts of $\mathcal{A}_1$ in these two interactions are identical except for $(r_i)_{i\in[n]\setminus\mathbb{S}}$. Observe that in the real interaction for $i \in [n] \setminus \mathbb{S}$, $r_i \leftarrow \{0,1\}^\ell$ while $\mathcal{A}_1'$ chooses $r_i := m_1 \oplus s_i$ where s_i is sampled uniformly at random by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$. Therefore, computing $r_i := m_1 \oplus s_i$ implies an identically distribution as when $r_i \leftarrow \{0,1\}^\ell$ given that $s_i \leftarrow \{0,1\}^\ell$ is independent of the transcript of $\mathcal{A}_1$.

Now let us consider a corrupt receiver $\mathcal{A}_2$. We construct a new adversary $\mathcal{A}_2'$ which interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and produces an identical output as $\mathcal{A}_2$. $\mathcal{A}_2'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and S in $\Pi_{k,n}^{\mathsf{R}}$ while running $\mathcal{A}_2$. $\mathcal{A}_2'$ receives the set $\mathbb{S} \subset [n]$ of size k and the endemic OT strings $(m_i)_{i\in\mathbb{S}}$ from $\mathcal{A}_2$. $\mathcal{A}_2'$ receives $r_1, ..., r_n$ from $\mathcal{A}_2$ and extracts $(s_i)_{i\in\mathbb{S}}$ as $s_i := r_i \oplus m_i$. $\mathcal{A}_2'$ sends $\mathbb{S}$ and $(s_i)_{i\in\mathbb{S}}$ to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and outputs whatever $\mathcal{A}_2$ outputs.

The output distribution of he honest sender is identical in these two interactions is identical except for $(s_i)_{i\in[n]\setminus\mathbb{S}}$. In the real interaction S outputs $s_i := m_i \oplus r_i$ where m_i is sampled uniformly by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ and independent of the transcript. Therefore $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ sampling $s_i \leftarrow \{0,1\}^\ell$ directly is identically distributed.

Therefore, any distinguishing advantage adversary $\mathcal{A}_1$ or $\mathcal{A}_2$ produces in protocol $\Pi_{k,n}^{\mathsf{R}}$ implies that $\mathcal{A}_1'$ or $\mathcal{A}_2'$ would produce the same advantage against the instantiation of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$, i.e. negl. □

Lemma 3.6. *Π^{U} of Figure 6 realizes the uniform message ideal OT $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ (Definition 2.4) with unconditional security in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{F}^{\mathsf{coin}}$ (Definition A.1)hybrid.*

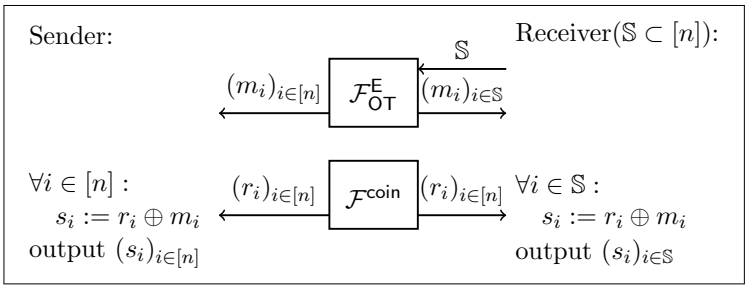

Figure 6: Uniform OT protocol $\Pi_{k,n}^{\mathsf{U}}$ in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{F}^{\mathsf{coin}}$ hybrid (Definition 2.4). For all $i \in [n]$, r_i, m_i and s_i are in $\{0,1\}^\ell$.

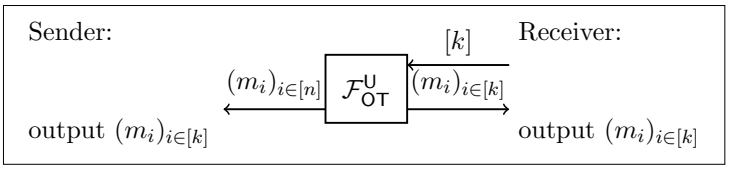

Figure 7: Coin flipping protocol Π^{coin} in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ hybrid. For all $i \in [n]$, m_i is in $\{0,1\}^\ell$.

Proof. First let us consider any corrupt sender $\mathcal{A}_1$. We construct a new adversary $\mathcal{A}_1'$ which interacts with functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and produces an identical output distribution as $\mathcal{A}_1$. $\mathcal{A}_1'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{F}^{\mathsf{coin}}$ and R in $\Pi_{k,n}^{\mathsf{U}}$ while running $\mathcal{A}_1$. $\mathcal{A}_1'$ receives the endemic OT strings $m_1, ..., m_n$ from $\mathcal{A}_1$.

$\mathcal{A}_1'$ invokes $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ as the sender and receives $s_1, ..., s_n$ in response. When $\mathcal{A}_1$ invokes $\mathcal{F}^{\mathsf{coin}}$, $\mathcal{A}_1'$ sends $r_1, ..., r_n$ to $\mathcal{A}_1$ on behalf of $\mathcal{F}^{\mathsf{coin}}$ where $r_i := m_1 \oplus s_i$ and outputs whatever $\mathcal{A}_1$ outputs. Observe that s_i is sampled uniformly at random by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and is independent of the transcript of $\mathcal{A}_1$ (conditioned on r_i). Therefore computing $r_i := m_1 \oplus s_i$ induces an identical distribution as $r_i \leftarrow \{0,1\}^\ell$.

Now let us consider a corrupt receiver $\mathcal{A}_2$. We construct a new adversary $\mathcal{A}_2'$ which interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and produces an identical output as $\mathcal{A}_2$. $\mathcal{A}_2'$ plays the role of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{F}^{\mathsf{coin}}$ and S in $\Pi_{k,n}^{\mathsf{U}}$ while running $\mathcal{A}_2$. $\mathcal{A}_2'$ receives the set $\mathbb{S} \subset [n]$ of size k and the endemic OT strings $(m_i)_{i\in\mathbb{S}}$ from $\mathcal{A}_2$.

$\mathcal{A}_2'$ invokes $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ as the receiver with input $\mathbb{S}$ and receives $(s_i)_{i\in\mathbb{S}}$ in response. When $\mathcal{A}_2$ invokes $\mathcal{F}^{\mathsf{coin}}\mathcal{A}_2'$ sends $r_1, ..., r_n$ to $\mathcal{A}_1$ on behalf of $\mathcal{F}^{\mathsf{coin}}$ where $r_i := m_1 \oplus s_i$ for $i \in \mathbb{S}$ and otherwise sets r_i as the output of $\mathcal{F}^{\mathsf{coin}}$. $\mathcal{A}_2'$ outputs whatever $\mathcal{A}_2$ outputs.

The transcripts of these two interactions are identical except for the messages $(r_i)_{i\in\mathbb{S}}$. In the real interaction r_i is sampled uniformly at random by $\mathcal{F}^{\mathsf{coin}}$ as opposed to $\mathcal{A}_2'$ computing $r_i := m_i \oplus s_i$. Observe that s_i is sampled uniformly at random by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and is independent of the transcript of $\mathcal{A}_2$ (conditioned on r_i). Therefore computing $r_i := m_1 \oplus s_i$ induces an identical distribution as $r_i \leftarrow \{0,1\}^\ell$.

Therefore, any distinguishing advantage adversary $\mathcal{A}_1$ or $\mathcal{A}_2$ produces in protocol Π^{U} implies that $\mathcal{A}_1'$ or $\mathcal{A}_2'$ would produce the same advantage against the instantiation of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ or $\mathcal{F}^{\mathsf{coin}}$, i.e. negl.

□

Lemma 3.7. *Π^{coin} of Figure 7 realizes an ideal coin flipping protocol (Definition A.1)with unconditional security in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ hybrid (Definition 2.4).*

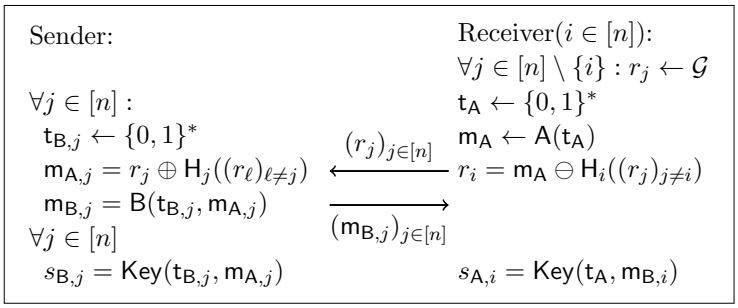

Figure 8: The figure depicts a 1 out of n OT using a $\mathsf{UKA} = (\mathsf{A}, \mathsf{B}, \mathsf{Key})$ and n random oracles, where for all $j \in [n]$, $\mathsf{H}_j : \mathcal{G}^{n-1} \to \mathcal{G}$ and $\mathcal{G}$ is a group with operations $\oplus$, $\ominus$. By the correctness of the UKA scheme, $\mathsf{k}_{\mathsf{A},i} = \mathsf{k}_{\mathsf{B},i}$ holds. In case of a one-round UKA the messages can be sent simultaneously.

Proof. Follows straigthforwardly from the definition of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and the ideal coin tossing functionality $\mathcal{F}^{\mathsf{coin}}$, which outputs a random string to both parties. □

As we have shown, endemic security allows to obtain any of the other notions efficiently. But as we show in the following lemmas, there can not be a one-round OT that achieves receiver or sender chosen message security. An adversary is at least able to tweak the distribution of the OT messages. As we will show in the upcoming section, there are OT protocols with a single round based on one-round key agreement.

Lemma 3.8. *There is no sender chosen message secure two message OT where the sender sends its message first.*

Lemma 3.9. *There is no receiver chosen message secure two message OT where the receiver sends its message first.*

Intuitively, if a malicious sender or receiver can choose its message in a two message protocol after seeing the message of the other party, he can bias the output distribution by resampling his message after observing what the learned OT message would be. We refer the reader for the formal argument to Appendix B.

4 From Key Agreement to Oblivious Transfer

In Figure 8, we present the generic construction from any two round key agreement to a two round OT. In Theorem 4.1, we show that both constructions yield an endemically secure OT. We emphasize that the protocol can be easily adapted to yield an all but one OT (see Appendix C).

In our security analysis for a malicious receiver, the simulator will rewind the malicious receiver. Hence we only obtain stand-alone security against a malicious receiver. In Appendix E, we show UC security against malicious receivers for our two-round construction and the one-round OT based on the Diffie-Hellman key agreement under a stronger variant of the DDH assumption.

Theorem 4.1. *Given a correct and secure* UKA *scheme, then the* 1 *out of* n *oblivious transfer in Figure 8 is an endemic* $\mathsf{OT}_{1,n}$ *with stand-alone security in the programmable random oracle model.*

Proof. Given a honest sender, receiver and the fact that UKA, an environment cannot distinguish the protocol from the ideal functionality.

We focus now on security against a malicious sender.

Claim 4.2. *Given a $(n-1)$-multi-instance ϵ-uniform* UKA *scheme, then it holds that in the programmable random oracle model for any ppt adversary $\mathcal{A}$, there exists a ppt adversary $\mathcal{A}'$ such that for any ppt distinguisher* D *and any polynomial size auxiliary input z,*

$$|\Pr[\mathsf{D}(z,(\mathcal{A},\mathsf{R})_\Pi)=1]-\Pr[\mathsf{D}(z,(\mathcal{A}',\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}))=1]|\leq\epsilon,$$

where all algorithms receive input 1^κ and R *additionally receives input* $\mathbb{S}$.

Proof. We define $\mathcal{A}'$ as follows. It generates $(r_j)_{j\in[n]}$ by sampling $r_1,\ldots,r_n\leftarrow\mathcal{G}$. Then, it samples for all $j\in[n]$, $\mathsf{t}_{\mathsf{A},j}\leftarrow\{0,1\}^*$ and $\mathsf{m}_{\mathsf{A},j}\leftarrow\mathsf{A}(\mathsf{t}_{\mathsf{A},j})$. Finally it programs the random oracle for all of the $j\in[n]$ points $(r_i)_{i\neq j}$ such that $r_i\oplus\mathsf{H}_i((r_i)_{i\neq j})=\mathsf{m}_{\mathsf{A},i}$. Now $\mathcal{A}'$ invokes $\mathcal{A}$, answers his random oracle queries straightforwardly, sends $(r_j)_{j\in[n]}$ and receives $(\mathsf{m}_{\mathsf{B},j})_{j\in[n]}$ from $\mathcal{A}$. It computes $s_{\mathsf{A},j}\leftarrow\mathsf{Key}(\mathsf{t}_{\mathsf{A},j},\mathsf{m}_{\mathsf{B},j})$ for all $j\in[n]$ and submits $(s_{\mathsf{A},j})_{j\in[n]}$ to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. $\mathcal{A}'$ outputs the output of $\mathcal{A}$.

We show, that if there is a distinguisher D that distinguishes the distribution $(\mathcal{A},\mathsf{R})_\Pi$ from $(\mathcal{A}',\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}})$, then there is an distinguisher $\mathsf{D}_{\mathsf{UKA}}$ against the n-multi-instance uniformity of the UKA scheme.

$\mathsf{D}_{\mathsf{UKA}}$ has access to an oracle $\mathcal{O}$ that either outputs uniform strings or messages of the UKA protocol. For all $j\in[n]\setminus\{i\}$, $\mathsf{D}_{\mathsf{UKA}}$ follows the description of $\mathcal{A}'$ with the difference that instead of sampling $\mathsf{m}_{\mathsf{A},j}\leftarrow\mathsf{A}(\mathsf{t}_{\mathsf{A},j})$, it samples $\mathsf{m}_{\mathsf{A},j}$ from $\mathcal{O}$. Given $(r_j)_{j\neq i}$, it samples $\mathsf{m}_{\mathsf{A},i}\leftarrow\mathsf{A}(\mathsf{t}_{\mathsf{A},i})$ and sets r_i such that $r_i\oplus\mathsf{H}_i((r_j)_{j\neq i})=\mathsf{m}_{\mathsf{A},i}$. As $\mathcal{A}'$, it computes $s_{\mathsf{A},i}\leftarrow\mathsf{Key}(\mathsf{t}_{\mathsf{A},i},\mathsf{m}_{\mathsf{B},i})$ which is R's output. It now invokes distinguisher D on R's output $s_{\mathsf{A},i}$ and the output of $\mathcal{A}$. In the end, it outputs the output of D.

We now analyze the distributions. First, notice that the distribution of $(r_i,\mathsf{m}_{A,i})$ when sampling $r_i\leftarrow\mathcal{G}$ and then programming the random oracle $r_i\oplus\mathsf{H}_i((r_j)_{j\neq i})=\mathsf{m}_{\mathsf{A},i}$ is identical to the distribution when sampling $\mathsf{H}_i((r_j)_{j\neq i})$ and choosing r_i such that $r_i\oplus\mathsf{H}_i((r_j)_{j\neq i})=\mathsf{m}_{\mathsf{A},i}$, both are the uniform distribution over $\mathcal{G}\times\mathcal{G}$ conditioned to their sum being $\mathsf{m}_{\mathsf{A},i}$. Therefore it follows straightforwardly from the definition of $\mathcal{O}$, R and $\mathcal{A}'$ that when $\mathcal{O}$ outputs uniform messages, the output of $\mathcal{A}$ is distributed as when interacting with R while when $\mathcal{O}$ outputs UKA messages, it is distributed as the output of $\mathcal{A}'$. Hence, if there is a distinguisher D for any z that distinguishes the output distribution of $\mathcal{A}$ given $s_{\mathsf{A},i}$, i.e.

$$\epsilon_{\mathsf{D}}\leq|\Pr[\mathsf{D}(z,(\mathcal{A},s_{\mathsf{A},i})_{\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_{\mathsf{A}}}})=1]-\Pr[\mathsf{D}(z,(\mathcal{A},s_{\mathsf{A},i})_{\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_u}})=1]|$$

then it implicitly breaks the $(n-1)$-multi-instance uniformity of the UKA protocol, i.e.

$$\begin{aligned}\epsilon &:= |\Pr[\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_{\mathsf{A}}}(z))=1]-\Pr[\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_u}(z)=1]|\\ &= |\Pr[\mathsf{D}(z,(\mathcal{A},s_{\mathsf{A},i})_{\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_{\mathsf{A}}}})=1]-\Pr[\mathsf{D}(z,(\mathcal{A},s_{\mathsf{A},i})_{\mathsf{D}_{\mathsf{UKA}}^{\mathcal{O}_u}})=1]|\\ &\geq \epsilon_{\mathsf{D}}.\end{aligned}$$

□

We finish the proof of the theorem by showing that the OT protocol is secure against a malicious receiver.

Claim 4.3. *Given a Q-multi-instance ϵ_u-uniform, $(Q,n-1)$-multi-instance ϵ_k-key indistinguishable* UKA *scheme, where Q upper bounds the amount of random oracle queries by an adversary then it holds that in the programmable random oracle model for any ppt adversary $\mathcal{A}$, there exists a ppt adversary $\mathcal{A}'$ such that for any ppt distinguisher* D *and any polynomial size auxiliary input z,*

$$|\Pr[\mathsf{D}(z,(\mathsf{S},\mathcal{A})_\Pi)=1]-\Pr[\mathsf{D}(z,(\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}},\mathcal{A}'))=1]|\leq\epsilon_u+\epsilon_k,$$

where all algorithms receive input 1^κ and $\mathcal{A}$' is expected to rewind $\mathcal{A}$ Q times.

Proof. Intuitively, we need to argue that all the $\mathsf{m}_{\mathsf{A},j}$ for which R does not learn $s_{\mathsf{A},j}$, $s_{\mathsf{A},j}$ is indistinguishable from uniform. To do this, we first exploit the uniformity of UKA to argue that $\mathsf{m}_{\mathsf{A},j}$ looks like an actual message of UKA. Afterwards, we can exploit the key-indistinguishability of UKA. To achieve this, we need to carefully program the random oracle.

We start by giving a description of $\mathcal{A}$'. $\mathcal{A}$' guesses a query index $\alpha \in [Q]$, where Q is an upper bound on the amount of oracle queries of $\mathcal{A}$. Then $\mathcal{A}$' invokes $\mathcal{A}$. If later this guess turns out to be incorrect, $\mathcal{A}$' aborts the current run with $\mathcal{A}$, rewinds $\mathcal{A}$ and makes a new guess.

When $\mathcal{A}$ makes an oracle query q to H_i for an $i \in [n]$ and the query number is less or equal to α, $\mathcal{A}$' responds with a random group element $\mathsf{H}_i(q) \leftarrow \mathcal{G}$. If the query number equals α, $\mathcal{A}$' stores $i^* := i$ and $(g_1^*, \ldots, g_{i^*-1}^*, g_{i^*+1}^*, \ldots, g_n^*) := q_\alpha$. For all following random oracle queries, i.e. the query number j is higher than α, $\mathcal{A}$' responds with a random group element $\mathsf{H}_i(q) \leftarrow \mathcal{G}$ if $i = i^*$ or for all $g \in \mathcal{G}$ $q_j \neq (g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^*$. Otherwise $\mathcal{A}$' samples random tape $\mathsf{t}_j \leftarrow \{0,1\}^*$ and computes $\mathsf{m}_j \leftarrow \mathsf{A}(\mathsf{t}_j)$. It responds with $\mathsf{H}_i(q_j) := \mathsf{m}_j \ominus g_i^*$. When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, $\mathcal{A}$' aborts if $q_\alpha \neq (r_1, \ldots, r_{i^*-1}, r_{i^*+1}, \ldots, r_n)$. $\mathcal{A}$' sends i^* to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. $\mathcal{A}$' computes for all $i \in [n]$ $\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i})$, $\mathsf{t}_{\mathsf{B},i} \leftarrow \{0,1\}^*$ and $\mathsf{m}_{\mathsf{B},i} \leftarrow \mathsf{B}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. It also computes $s_{\mathsf{B},i^*} := \mathsf{Key}(\mathsf{t}_{\mathsf{B},i^*}, \mathsf{m}_{\mathsf{A},i^*})$. $\mathcal{A}$' sends s_{B,i^*} to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$, $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ and outputs the output of $\mathcal{A}$. In case of one-round OT, $\mathcal{A}$' generates and sends $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ in the very beginnning. This concludes the description of $\mathcal{A}$'. In total, $\mathcal{A}$' is expected to rewind $\mathcal{A}$ Q times.

Let there be a distinguisher D with

$$\epsilon_{\mathsf{D}} := |\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_\Pi) = 1] - \Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A}')) = 1]|,$$

where $(s_{\mathsf{B},i})_{i\in[n]}$ are the outputs of $\mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. Then there is a distinguisher D_u breaking the Q-multi-instance uniformity of the UKA protocol. D_u gets access to an oracle $\mathcal{O}$ which either outputs uniform messages, i.e. $\mathcal{O}_u$ or messages of the form $\mathsf{m}_\mathsf{A} \leftarrow \mathsf{A}(\mathsf{t}_\mathsf{A})$ for $\mathsf{t}_\mathsf{A} \leftarrow \{0,1\}^*$. D_u invokes D and creates its input as follows. It invokes $\mathcal{A}$ and interacts with him as $\mathcal{A}$' does with the difference that m_j are requested from $\mathcal{O}$ rather than computing them. After receiving the output, D_u uses it as input for D together with $(s_{\mathsf{B},i})_{i\in[n]}$, where $s_{\mathsf{B},i} \leftarrow \mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. D_u outputs the output of D.

If $\mathcal{O}$ is oracle $\mathcal{O}_u$, all m_j are uniform and hence all random oracle queries q are answered with a uniformly random $\mathsf{H}_i(q) \in \mathcal{G}$. Otherwise, $\mathcal{A}$' is identical with S as well as $(s_{\mathsf{B},i})_{i\in[n]}$ are identical with the output of S. Hence

$$\begin{aligned}
\epsilon_u &= |\Pr[\mathsf{D}_u^{\mathcal{O}_\mathsf{A}}(z)] = 1] - Pr[\mathsf{D}_u^{\mathcal{O}_u}(z) = 1]| \\
&= |\Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_u^{\mathcal{O}_\mathsf{A}}}) = 1] \\
&\quad - \Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_u^{\mathcal{O}_u}}) = 1]| \\
&\geq \epsilon_{\mathsf{D}}.
\end{aligned}$$

Next, we assume that there is a distinguisher D with

$$\epsilon_{\mathsf{D}} := |\Pr[\mathsf{D}(z, (s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A}) = 1] - \Pr[\mathsf{D}(z, \mathbb{S}_{\mathsf{B},i^*}, \mathcal{A}) = 1]|,$$

where for all $i \in [n]$, s_i is sampled uniformly from the key space of UKA and $\mathbb{S}_{\mathsf{B},i^*} := (s_1, \ldots, s_{i^*-1}, s_{\mathsf{B},i^*}, s_{i^*+1}, \ldots, s_n$ Then there is a distinguisher D_k that breaks the $(Q, n-1)$-multi-instance key-indistinguishability of the UKA protocol. D_k has access to oracles $\mathcal{O}_{\langle\mathsf{A},\mathsf{B}\rangle}$ and $\mathcal{O}$ which is either $\mathcal{O}_u$ or $\mathcal{O}_k$. D_k invokes D and creates its input as follows. D_k invokes $\mathcal{A}$ and interacts with it as $\mathcal{A}'$ does with the difference, that D_k generates m_j by querying a transcript $\langle\mathsf{A},\mathsf{B}\rangle = (\mathsf{m}'_{\mathsf{A},j}, \mathsf{m}'_{\mathsf{B},j})$ from $\mathcal{O}_{\langle\mathsf{A},\mathsf{B}\rangle}$ and setting $\mathsf{m}_j = \mathsf{m}'_{\mathsf{A},j}$. $\mathcal{A}$' computes for all $i \in [n] \setminus \{i^*\}$

$$\mathsf{m}'_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i}) = \mathsf{m}'_{\mathsf{A},j}$$

where there exists a $j \in [Q]$ such that the last equality holds. It also uses oracle $\mathcal{O}$ to query for all $i \in [n] \setminus \{i^*\}$ the $n-1$ corresponding keys k_i that match with the transcripts containing $\mathsf{m}_{\mathsf{A},i}$. D_k sets $\mathsf{m}_{\mathsf{B},i} := \mathsf{m}'_{\mathsf{B},j}$ and $s_{\mathsf{B},i} := \mathsf{k}_i$. It creates $\mathsf{m}_{\mathsf{B},i^*}$ and s_{B,i^*} as $\mathcal{A}$' does. It sends $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ to receive its output which it uses together with $(s_{\mathsf{B},i})_{i\in[n]}$ as input for D. D_k outputs D's output.

$$\begin{aligned}
\epsilon_k &= |\Pr[\mathsf{D}_k^{\mathcal{O}_k}(z)] = 1] - Pr[\mathsf{D}_k^{\mathcal{O}_u}(z) = 1]| \\
&= |\Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_k^{\mathcal{O}_k}}) = 1] \\
&\quad -Pr[\mathsf{D}(z, (\mathbb{S}_{\mathsf{B},i^*}, \mathcal{A})_{\mathsf{D}_k^{\mathcal{O}_u}}) = 1]| \\
&\geq \epsilon_{\mathsf{D}}.
\end{aligned}$$

We conclude with:

$$\begin{aligned}
\epsilon_{\mathsf{OT}} &= |\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_\Pi) = 1] - \Pr[\mathsf{D}(z, (\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{A})) = 1]| \\
&\leq \epsilon_u + |\Pr[\mathsf{D}((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A}) = 1] - \Pr[\mathsf{D}(\mathbb{S}_{\mathsf{B},i^*}, \mathcal{A}) = 1]| \\
&\leq \epsilon_u + \epsilon_k,
\end{aligned}$$

□

□

5 OT Extension

Next we review the OT extension protocol of [KOS15, OOS17, ALSZ17] which we describe in Figure 9. The base OTs are performed on inputs that are sampled uniformly at random where the roles of the sender and receiver are reversed with respect to the OTs that are output by the extension. That is, S will receive $(b_j, \boldsymbol{t}^j_{b_j}) \in \mathbb{F}_2 \times \mathbb{F}_2^m$ while R will receive $(\boldsymbol{t}^j_0, \boldsymbol{t}^j_1) \in \mathbb{F}_2^m \times \mathbb{F}_2^m$ for $j \in [n_{\mathcal{C}}]$.

R forms two matrices $T_0, T_1 \in \mathbb{F}_2^{m \times n_{\mathcal{C}}}$ by concatenating the base OT messages as column vectors, i.e. $T_i := (\boldsymbol{t}^1_i ... \boldsymbol{t}^{n_{\mathcal{C}}}_i)$. Similarly, S forms the matrix $T_{\mathbf{b}} := (\boldsymbol{t}^1_{b_1} ... \boldsymbol{t}^{n_{\mathcal{C}}}_{b_{n_{\mathcal{C}}}})$. R then encodes their 1-out-of-N selections $\boldsymbol{w}_1, ..., \boldsymbol{w}_m$ into a matrix $C \in \mathbb{F}_2^{m \times n_{\mathcal{C}}}$. Each row $\boldsymbol{c}_i$ is the codeword $\mathcal{C}(\boldsymbol{w}_i)$, where $\mathcal{C}$ is a binary code of length $n_{\mathcal{C}}$, dimension $k_{\mathcal{C}} = \log_2 \kappa$ and minimum distance $d_{\mathcal{C}} \geq \kappa$. R sends the matrix $U = T_0 + T_1 + C$ to S. Observe that U encodes the selections of R but the selection is perfectly masked/encrypted due to the j-th column of U being masked by the column $\boldsymbol{t}^j_{1-b_j}$ which is uniformly distributed in the view of S. Upon receiving U, S computes $Q \in \mathbb{F}^{m \times n_{\mathcal{C}}}$ where the j-th column is defined as $\boldsymbol{q}^j := b_j \cdot \boldsymbol{u}^j + \boldsymbol{t}^j_{b_j} = b_j \cdot \boldsymbol{c}^j + \boldsymbol{t}^j_0$. It holds that

$$\boldsymbol{q}_i = \boldsymbol{c}_i \odot \boldsymbol{b} + \boldsymbol{t}_i$$

where $\odot$ is bitwise multiplication, $\boldsymbol{t}_i, \boldsymbol{q}_i$ is the i-th row of T_0, Q, respectively, and $\boldsymbol{b} := (b_1, ..., b_{n_{\mathcal{C}}}) \in \mathbb{F}_2^{n_{\mathcal{C}}}$. R will output $\boldsymbol{v}_{i,\boldsymbol{w}_i} := \mathsf{H}(i, \boldsymbol{t}_i)$. S can then generate any OT message by computing $\boldsymbol{v}_{i,\boldsymbol{w}} := \mathsf{H}(i, \boldsymbol{q}_i + \mathcal{C}(\boldsymbol{w}) \odot \boldsymbol{b})$. Correctness of this operation follows from

$$\begin{aligned}
\boldsymbol{q}_i + \mathcal{C}(\boldsymbol{w}) \odot \boldsymbol{b} &= (\mathcal{C}(\boldsymbol{w}_i) \odot \boldsymbol{b} + \boldsymbol{t}_i) + \mathcal{C}(\boldsymbol{w}) \odot \boldsymbol{b} \\
&= (\mathcal{C}(\boldsymbol{w}_i) + \mathcal{C}(\boldsymbol{w})) \odot \boldsymbol{b} + \boldsymbol{t}_i.
\end{aligned}$$

Let $\delta = \mathcal{C}(\boldsymbol{w}_i) + \mathcal{C}(\boldsymbol{w})$. In the event that $\boldsymbol{w}_i = \boldsymbol{w}$, then $\delta = 0$ and S computes the same $\boldsymbol{v}_{i,\boldsymbol{w}_i}$ value as R. Otherwise the hamming distance $\mathsf{HD}(\delta) \geq d_{\mathcal{C}} \geq \kappa$ by construction of $\mathcal{C}$. For R to generate

any other OT message $\boldsymbol{v}_{i,\boldsymbol{w}}$ s.t. $\boldsymbol{w} \neq \boldsymbol{w}_i$, R must guess the value $\delta \odot \boldsymbol{b} \in \mathbb{F}_2^{nc}$ given δ, which can be done with probability $2^{-\mathsf{HD}(\delta)} = O(2^{-\kappa})$.

Traditionally, two additional steps are specified to realize the ideal sender chosen message functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ [IKNP03, KOS15, OOS17, ALSZ17]:

1. A proof that all rows in C can be decoded. Ishai et al. [IKNP03] proposed a cut-and-choose approach while the more recent schemes [KOS15, OOS17] improve on the efficiency of these proofs by making R send random linear combinations of $\boldsymbol{t}_i, \boldsymbol{w}_i$ and having S check they are consistent with same combination of U. [OOS17] follows a slightly different strategy. We defer the details behind these proofs to [KOS15, OOS17, OOS17].

2. The parties apply the sender chosen OT transformation $\Pi_{1,N}^{\mathsf{S}}$ from Figure 4 which reduces sender chosen to endemic OT. That is, S must send their chosen messages $(x_{i,1}, ..., x_{i,N})_{i\in[m]}$ encrypted under the corresponding key $(\boldsymbol{v}_{i,1}, ..., \boldsymbol{v}_{i,N})_{i\in[m]}$, e.g. S sends $e_{i,j} := x_{i,j} + \boldsymbol{v}_{i,j}$ to R who outputs $x_{i,\boldsymbol{w}_i} = e_{i,\boldsymbol{w}_i} + \boldsymbol{v}_{i,\boldsymbol{w}_i}$. Note, this step is not included in Figure 9. Next we will show without this step the protocol only achieves endemic security.

5.1 OT Extension Attacks

The authors of [KOS15, ALSZ17] and [OOS17] provide protocol descriptions that are intended to (respectively) satisfy the sender and uniform chosen message security notion, Definition 2.6, but we show this to not be the case. These protocols can be summarized as the previous protocol description where the $\Pi_{1,N}^{\mathsf{S}}$ transformation is not applied, i.e. output $\boldsymbol{v}_{i,x_i}$. For the rest of this work we will refer to the protocol of [OOS17] as defined in Definition 5.1 but note that the attacks by a malicious R apply to [KOS15, Figure 6, 7] and [ALSZ17, Protocol 10]. In particular, we detail three attacks where the first (Lemma F.1) allows a malicious R to bias the OT messages that they output while the second and third attacks (Lemma F.2, F.3) succeed even when base OTs with stronger security are used. In all cases, the ability to bias the messages violates the ideal functionality which samples them uniformly at random.

Definition 5.1. *Let Π^{OOS} be the protocol of Figure 9 where $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$.*

Remark 5.2. *[OOS17] is inconsistent which type of base OTs should be used, switching between standard Sender Chosen Message OT ($\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}} = \mathcal{F}_{\text{2-OT}}^{\kappa,nc}$) in the protocol description, theorem statements and Uniform Message OT ($\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}} = \mathcal{F}_{\text{2-ROT}}^{\kappa,nc}$) in their proof. Lemma F.1 only applies to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}} = \mathcal{F}_{\text{2-OT}}^{\kappa,nc}$ while Lemma F.2 and F.3 apply even with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}} = \mathcal{F}_{\text{2-ROT}}^{\kappa,nc}$ base OTs. All three attacks apply to [KOS15] which uses $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}} = \mathcal{F}_{\text{2-OT}}^{\kappa,nc}$.*

Lemma F.1 details an attack which allows R to bias the output $\boldsymbol{v}_{i,x_i}$ to be $\mathsf{H}(i,x)$ for any $x \in \mathbb{F}_2^{nc}$. The core idea behind this attack is that R has complete control over the matrix T_0 since they input it to the base OTs. As such, R can choose their output messages to be $\boldsymbol{v}_{i,x_i} = \mathsf{H}(i, \boldsymbol{t}_i)$ for any $\boldsymbol{t}_i$. For example, let $\boldsymbol{t}_1 = \boldsymbol{t}_i$ for all $i \in [m]$. Then the distinguisher can compare the output of S and outputs 1 if $\boldsymbol{v}_{1,x_1} = \boldsymbol{v}_{i,x_i}$.

Lemma 5.3. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{OOS}}) = 1] - \Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1]| = 1 - 2^{-\kappa}$$

where Π^{OOS} is the protocol in Definition 5.1. All algorithms also receive input 1^κ.

PARAMETERS: κ is the computational security parameter. m denotes the number of OTs. N denotes the number of messages each OT has. $\mathcal{C}$ is an $[n_{\mathcal{C}}, k_{\mathcal{C}}, d_{\mathcal{C}}]$ binary linear code such that $k_{\mathcal{C}} = \log_2 N$ and $d_{\mathcal{C}} \geq \kappa$. A bijective map $map : [N] \to \mathbb{F}^{k_{\mathcal{C}}}$.
REQUIREMENTS: $\mathsf{H} : [m] \times \mathbb{F}_2^{n_{\mathcal{C}}} \to \mathbb{F}_2^{\kappa}$ is a random oracle. Let $m' = m + s$ where s is defined in Step 4. $\mathcal{F}_{\mathsf{OT}}$ is an 1-out-of-2 OT oracle with output messages in $\mathbb{F}_2^{m'}$.

EXTEND: On input (EXTEND) from S and (EXTEND, $(x_1, ..., x_m) \in [N]^m$) from R.

1. Both parties invoke $n_{\mathcal{C}}$ instances of $\mathcal{F}_{\mathsf{OT}}$ where S takes the role of the receiver. If $\mathcal{F}_{\mathsf{OT}}$ has inputs, the corresponding party locally samples them uniformly from the input domains. S receives $(\boldsymbol{b}' \in \{1,2\}^{n_{\mathcal{C}}}, \{\boldsymbol{t}^j_{b_j}\}_{j\in[n_{\mathcal{C}}]})$ where $\boldsymbol{b}_i = (\boldsymbol{b}'_i - 1) \in \{0,1\}$. R receives $\{(\boldsymbol{t}^j_0, \boldsymbol{t}^j_1)\}_{j\in[n_{\mathcal{C}}]}$. Let $T_i \in \mathbb{F}_2^{m'\times n_{\mathcal{C}}}$ denote the matrix formed by concatenating the column vectors $\boldsymbol{t}^1_i||...||\boldsymbol{t}^{n_{\mathcal{C}}}_i$.

2. R defines $\boldsymbol{w}_i := map(x_i)$ for $i \in [m]$ and samples random $\boldsymbol{w}_{m+\ell} \leftarrow \mathbb{F}_2^{k_{\mathcal{C}}}$, for $\ell \in [s]$. Then constructs a matrix $C \in \mathbb{F}_2^{m'\times n_{\mathcal{C}}}$ such that each row $\boldsymbol{c}_i$ is the codeword $\mathcal{C}(\boldsymbol{w}_i)$. Then, R sends to S the values
$$\boldsymbol{u}^j := \boldsymbol{t}^j_0 + \boldsymbol{t}^j_1 + \boldsymbol{c}^j, \qquad \forall j \in [n_{\mathcal{C}}],$$
where $\boldsymbol{c}^j$ is the j-th column of C.

3. S receives $\boldsymbol{u}^j \in \mathbb{F}_2^{m'}$ and computes
$$\boldsymbol{q}^j := b_j \cdot \boldsymbol{u}^j + \boldsymbol{t}^j_{b_j} = b_j \cdot \boldsymbol{c}^j + \boldsymbol{t}^j_0, \qquad \forall j \in [n_{\mathcal{C}}]$$
that form the columns of an $(m' \times n_{\mathcal{C}})$ matrix Q. Denoting the rows of T_0, T_1, Q by $\boldsymbol{t}_i, \boldsymbol{t}_{1,i}, \boldsymbol{q}_i$, R now holds $\boldsymbol{c}_i, \boldsymbol{t}_i$ and S holds $\boldsymbol{b}, \boldsymbol{q}_i$ so that
$$\boldsymbol{q}_i = \boldsymbol{c}_i \odot \boldsymbol{b} + \boldsymbol{t}_i, \qquad \forall i \in [m'].$$

4. *Consistency check:* R proves in zero knowledge that
$$\forall i \in [m], \exists w \in \mathbb{F}_2^{k_{\mathcal{C}}} \mid 0 = \boldsymbol{b} \odot (\boldsymbol{u}_i + \boldsymbol{t}_i + \boldsymbol{t}_{1,i} + \mathcal{C}(w))$$
Note: $\boldsymbol{b} \in \mathbb{F}_2^{n_{\mathcal{C}}}$ is distributed uniformly in the view of R.
For example, the proof of [KOS15] for $N = 2$ or [OOS17] otherwise. $s \geq 0$ is specified by the proof protocol.

5. R outputs $\boldsymbol{v}_{i,x_j} := \mathsf{H}(i, \boldsymbol{t}_i)$ for all $i \in [m]$.

OUTPUT: On input (OUTPUT, (i, x)) from S. If $i \in [m], j \in [N]$, then S outputs $\boldsymbol{v}_{i,x} := \mathsf{H}(i, \boldsymbol{q}_i + \mathcal{C}(map(x)) \odot \boldsymbol{b})$.

Figure 9: 1-out-of-N OT Extension.

Proof. For simplicity let $N = 2$ and $m = 1$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of R and replaces the input to base OTs, the sender input, with strings $\boldsymbol{t}^j_0, \boldsymbol{t}^j_1 \in \{0\}^{m'}$ and then completes the protocol as normal.

We define D as follows. D executes S and $\mathcal{A}$ with input $x_1 = 1$. S outputs $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2})$ and D outputs 1 if $\boldsymbol{v}_{1,1} = \mathsf{H}(1, \{0\}^{n_{\mathcal{C}}})$ and 0 otherwise. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{\mathsf{OOS}}}) = 1] = 1$. In the ideal interaction the honest S will output a uniformly distributed $\boldsymbol{v}_{1,1} \in \{0,1\}^{\kappa}$ which was sampled by $\mathcal{F}^{\mathsf{U}}_{\mathsf{OT}}$ and therefore $\Pr[\mathsf{D}((\mathcal{F}^{\mathsf{U}}_{\mathsf{OT}}, \mathcal{A}')) = 1] = 2^{-\kappa}$. □

We now focus our attention to a second class of adversary that can distinguish even when the base OTs output uniformly distributed messages, i.e. $\mathcal{F}^{\mathsf{U}}_{\mathsf{OT}}$. [OOS17] is inconsistent which type of

base OTs should be used, switching between $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ in the protocol and $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ in the proof. Regardless the next two attacks apply.

Definition 5.4. *Let Π^{OOS+} be the protocol of Figure 9 where $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$.*

The core idea behind the Lemma F.2 attack against $\Pi^{\mathsf{OOS+}}$ is that R can choose their selection *after* seeing their output message, i.e. $H(i, \boldsymbol{t}_i)$. This allows R to correlate their selection x_i with their message $H(i, \boldsymbol{t}_i)$ and there by distinguish. For example, let $v = \mathsf{H}(i, \boldsymbol{t}_i) \mod N$ and then R makes their selection be $x_i = v + 1$. This can not happen when interacting with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$.

Lemma 5.5. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{OOS+}}) = 1] - \Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1]| = 1 - 2^{-\kappa}$$

where Π^{OOS+} is the protocol in Definition 5.4 and all algorithms additionally receive input 1^κ.

Proof. For simplicity let $N = 2$ and $m = \kappa$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of R and receives the strings $\boldsymbol{t}_0^j, \boldsymbol{t}_1^j \in \{0\}^{m'}$ from $\mathcal{F}_{\mathsf{OT}}$. $\mathcal{A}$ redefines the selection values $x_1, ..., x_m \in [2]$ of R such that $x_i := \text{LSB}(\mathsf{H}(i, \boldsymbol{t}_i)) + 1$. That is, x_i equals the least significant bit of $\boldsymbol{v}_{i,x_i} = \mathsf{H}(i, \boldsymbol{t}_i)$ plus 1. $\mathcal{A}$ executes the rest of the protocol as R would and outputs $(x_i)_{i \in [m]}$.

We define D as follows. D executes S and $\mathcal{A}$. S outputs $(\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2})_{i \in [m]}$ and D outputs 1 if $\forall i \in [m], \text{LSB}(\boldsymbol{v}_{i,x_i}) + 1 = x_i$ and 0 otherwise. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{\mathsf{OOS+}}}) = 1] = 1$. In the ideal interaction the honest S will output a uniformly distributed $\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2} \in \{0,1\}^\kappa$ which are independent of x_i and therefore $\Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1] = 2^{-\kappa}$. □

Lemma F.3 details another attack where a malicious S sets the base OT selection values to be $\widehat{\boldsymbol{b}} := (1, ..., 1) \in \{1,2\}^{nc}$. As such S learns the matrix T_0 in full. Therefore S can always output the same message $\mathsf{H}(i, \boldsymbol{t}_i) = \boldsymbol{v}_{i,\boldsymbol{w}_i}$ as R. For sender chosen message or endemic security a viable simulation strategy is to extract $H(i, \boldsymbol{t}_i)$ and define $\boldsymbol{v}_{i,j} := \mathsf{H}(i, \boldsymbol{t}_i)$ for all j. However, there is no valid strategy for the receiver chosen or uniform message security where the oracle samples some of the messages uniformly. This attack breaks the security of the set inclusion protocol described by [OOS17, Figure 5].

Lemma 5.6. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathcal{A}, \mathsf{R})_{\Pi^{OOS+}}) = 1] - \Pr[\mathsf{D}((\mathcal{A}', \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}})) = 1]| = 1 - \mathsf{negl}$$

where Π^{OOS+} is the protocol in Definition 5.4 and all algorithms additionally receive input 1^κ.

Proof. For simplicity let $N = 2$ and $m = \kappa$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of S and replaces the input to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$, the receiver input, with the string $\boldsymbol{b} := \{0\}^{nc}$. $\mathcal{A}$ outputs the matrix Q.

We define D as follows. D samples the selection bits $x_1, ..., x_m \leftarrow [2]$ and sends them to R. D executes $\mathcal{A}$ who outputs Q and R outputs $\boldsymbol{v}_{1,x_1}, ..., \boldsymbol{v}_{m,x_m}$. If $\boldsymbol{v}_{i,x_i} = \mathsf{H}(i, \boldsymbol{q}_i)$ for all $i \in [m]$, output 1, otherwise 0. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathcal{A}, \mathsf{R})_{\Pi^{\mathsf{OOS+}}}) = 1] = 1$ since $\boldsymbol{q}_i = \boldsymbol{t}_i$.

By definition the input of $\mathcal{A}'$ is independent of x_i and receives no output from $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ (apart from their input $(\boldsymbol{v}_{0,i}, \boldsymbol{v}_{1,i})_{i \in [m]}$). Therefore, it must hold that $\Pr[\mathsf{D}((\mathcal{A}', \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}})) = 1] = 2^\kappa$. □

5.2 OT Extension with a Random Oracle

We now give a new security proof (Lemma 5.8) of the [KOS15, OOS17] protocols with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ ideal functionality. We then give new enhancements (Definition 5.14, 5.17, F.6) to this protocol that provider stronger notions of security at a modest overhead, e.g. $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$. Note that in

this section we used the generalize definition of the OT functionality, Definition A.13,where a circuit specifying the inputs are send instead of strings. We also give a data flow diagram in Figure 16 showing the various instantiations and their round complexity.

Definition 5.7. *Let* $\Pi^{\mathsf{ext\text{-}E}}$ *be the protocol of Figure 9 where* $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$.

Lemma 5.8. *The* $\Pi^{\mathsf{ext\text{-}E}}$ *protocol (Definition 5.7) is a 1-out-of-N OT* ($\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$) *satisfying endemic and receiver selection Security.*

Proof. Correctness of the protocol was demonstrated by [OOS17].

Claim 5.9 (Malicious Sender Security)**.** $\Pi^{\mathsf{ext\text{-}E}}$ *satisfies security against a malicious sender (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ *functionality.*

Proof. Consider the following hybrids which will define the simulator $\mathcal{A}'$.

Hybrid 1. $\mathcal{A}'$ internally runs $\mathcal{A}$ while plays the role of R and base OT oracle $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. For $j \in [n_{\mathcal{C}}]$, $\mathcal{A}'$ receives $(\boldsymbol{b}'_j, \boldsymbol{t}^j_{\boldsymbol{b}_j}) \in [2] \times \mathbb{F}_2^{m'}$ from $\mathcal{A}$ in Step 1 where $\boldsymbol{b}_j := \boldsymbol{b}'_j - 1$. $\mathcal{A}'$ uniformly samples $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ as $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ would. $\mathcal{A}'$ sends $(\boldsymbol{b}'_j, \{\boldsymbol{t}^j_{\boldsymbol{b}_j}\})$ to $\mathcal{A}$ on behalf of $\mathcal{F}_{\mathsf{OT}}$. $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs.

Hybrid 2. For Step 2 $\mathcal{A}'$ does not sample $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ and instead uniformly samples $U \leftarrow \mathbb{F}_2^{m' \times n_{\mathcal{C}}}$. $\mathcal{A}'$ sends U to $\mathcal{A}$ and then computes Q as S would. The view of $\mathcal{A}$ is identically distributed. This follows from the fact that $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ is uniformly distributed in the view of $\mathcal{A}$ and masks the j-th column of U in the previous hybrid.

Hybrid 3. For each row $\boldsymbol{q}_i$, $\mathcal{A}'$ defines the circuit $\mathcal{M}_i : [N] \to \{0,1\}^{\kappa}$ such that on input $j \in [N]$ it outputs $\mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(j)))$. $\mathcal{A}'$ sends $\mathcal{M}_i$ to the ideal functionality $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ as the input to the i-th OT instance. This change allows the ideal functionality to output the same distribution as the real protocol. The view of $\mathcal{A}$ is unmodified. Note, $\mathcal{A}$ can influence $\mathcal{M}_i(j) = \mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(j))) = H(i, (c_i + \mathcal{C}(map(j)) \odot \boldsymbol{b} + \boldsymbol{t}_i)$ by choosing $\boldsymbol{b}$ and the bits $\{\boldsymbol{t}_i[j] \mid \boldsymbol{b}_j = 0\}$.

Hybrid 4. For Step 4 $\mathcal{A}'$ simulates the consistency proof.

Hybrid 5. $\mathcal{A}'$ does not take the input of R since it was not used. R only interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. This change is identically distributed.

□

Claim 5.10 (Malicious Receiver Security)**.** $\Pi^{\mathsf{ext\text{-}E}}$ *satisfies security against a malicious receiver (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ *functionality.*

Proof. Consider the following hybrids which will define the simulator $\mathcal{A}'$.

Hybrid 1. $\mathcal{A}'$ internally runs $\mathcal{A}$ while plays the role of S and base OT functionality $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. $\mathcal{A}'$ receives $\{\boldsymbol{t}^j_0, \boldsymbol{t}^j_1\}_{i \in n_{\mathcal{C}}}$ from $\mathcal{A}$ in Step 1. $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. The view of $\mathcal{A}$ is unmodified.

Hybrid 2. In Step 2 $\mathcal{A}'$ receives U from $\mathcal{A}$, computes $C := T_0 + T_1$ and uniformly samples $\boldsymbol{b} \leftarrow \mathbb{F}_2^{n_{\mathcal{C}}}$. Let $B_i := \{j \mid \boldsymbol{b}_j = i\}$. For all $i \in [m]$, $\mathcal{A}'$ attempts to erasure decode $\boldsymbol{c}_i$ with erasures indexed by B_0. If $\boldsymbol{c}_i$ failed to decode, then there does not exist a $w \in \mathbb{F}_2^{k_{\mathcal{C}}}$ s.t. $0 = \boldsymbol{b} \odot (\boldsymbol{c}_i + \mathcal{C}(w))$ and $\mathcal{A}'$ aborts in Step 4 as S would. Otherwise, let $\boldsymbol{c}_i$ decode to $\boldsymbol{w}_i$ and $\mathcal{A}'$ computes x_i s.t. $\boldsymbol{w}_i = map(x_i)$

$\mathcal{A}'$ defines the circuit $\mathcal{S}_i[1] \to \{0,1\}$ with support $\{x_i\}$ and $\mathcal{M}_i : [1] \to \mathbb{F}_2^\kappa$ s.t. $\mathcal{M}_i(1) = \mathsf{H}(i, \boldsymbol{t}_i)$. $\mathcal{A}'$ sends $\mathcal{S}_i$ and $\mathcal{M}_i$ to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ as the receiver's input to the i-th OT instance. The view of $\mathcal{A}$ is unmodified and the ideal-real output agree on $\boldsymbol{v}_{i,x_i}$.

Hybrid 3. Assuming $\mathcal{A}'$ did not abort in Step 4, let $E = \{j \mid \exists i \in [m], (\boldsymbol{c}_i \oplus \mathcal{C}(\boldsymbol{w}_i))_j = 1\}$ index the columns of C where $\mathcal{A}$ added an error to any codeword $\boldsymbol{c}_i$ (w.r.t $\boldsymbol{w}_i$). By the correctness of Step 4, it holds that $E \subseteq B_0$, otherwise the consistency proof would have failed. By passing the consistency proof, $\mathcal{A}$ learns what $\boldsymbol{b}_j = 0$ for all $j \in E$. Similarly, the probability of passing the check and $\Pr[|E| = d] = \Pr[\boldsymbol{b}_j = 0 \mid \forall j \in E] = 2^{-d}$ due to the proof being independent of $\boldsymbol{b}$. We will see that this is equivalent to $\mathcal{A}$ simply guessing E (which is correct with the same probability) and then being honest.

For all $w \neq \boldsymbol{w}_i$, $\mathcal{A}$ has negl probability of computing $g = \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w)$. If this was not the case, then $\mathcal{A}$ could compute

$$g + \boldsymbol{t}_i = (\boldsymbol{c}_i + \mathcal{C}(w)) \odot \boldsymbol{b} = (\mathcal{C}(\boldsymbol{w}_i) + \mathcal{C}(w)) \odot \boldsymbol{b}$$

This last equality holds due to $\mathcal{A}'$ aborting if $(\boldsymbol{c}_i + \mathcal{C}(\boldsymbol{w}_i)) \odot \boldsymbol{b} \neq 0$. Recall that $\mathcal{C}$ has minimum distance $d_{\mathcal{C}} \geq \kappa$ and therefore computing g is equivalent $\mathcal{A}$ guessing $d_{\mathcal{C}} \geq \kappa$ bits of $\boldsymbol{b}$ which happens with probability $2^{-d_{\mathcal{C}}} \leq 2^{-\kappa}$. As such, the probability that $\mathcal{A}$ has made a query of the form $\mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w))$ for $w \neq \boldsymbol{w}_i$ is also negligible. If such as query does happen $\mathcal{A}'$ aborts. This hybrid is indistinguishably distributed from the previous.

Hybrid 4. When S makes an H query of the form $\mathsf{H}(i, h)$ which has not previously be queries, $\mathcal{A}'$ must determine if there is a unique $w \in \mathbb{F}_2^{k_{\mathcal{C}}}$ such that $h = \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w)$. First, let us assume there exists two distinct $w, w' \in \mathbb{F}_2^{k_{\mathcal{C}}}$ that result in h. That is,

$$\begin{aligned} \boldsymbol{b} \odot (c_i + \mathcal{C}(w)) &= \boldsymbol{b} \odot (c_i + \mathcal{C}(w')) \\ 0 &= \boldsymbol{b} \odot (\mathcal{C}(w) + \mathcal{C}(w')) \end{aligned}$$

Recall that $\mathcal{C}$ by construction has minimum distance $d_{\mathcal{C}} \geq \kappa$ and that $\boldsymbol{b}$ is uniformly distributed. Let $\delta = \mathcal{C}(w) + \mathcal{C}(w')$ and $E = \{i \mid \delta_i = 1\}$, then $|E| \geq d_{\mathcal{C}} \geq \kappa$ and for the above to hold we require $\boldsymbol{b}_i = 0 \mid \forall i \in E$ which occurs with probability $\Pr[\boldsymbol{b}_i = 0 \mid \forall i \in E] = 2^{-|E|} \leq 2^{-d_{\mathcal{C}}} \leq 2^{-\kappa}$. In such an event the simulations fails but this occurs with negligible probability.

$\mathcal{A}'$ checks that $(h + \boldsymbol{q}_i)_\ell = 0$ for all $\ell \in \{i \mid \boldsymbol{b}_i = 0\}$ and if so uses Gaussian elimination to determine if there exists a w such that $h + \boldsymbol{q}_i$ erasure decodes to w where the erasures are index by $B_0 = \{i \mid \boldsymbol{b}_i = 0\}$. If so, $\mathcal{A}'$ computes x s.t. $map(x) = w$ and sends $(\textsc{Output}, x)$ to the i-th instance of $\mathcal{F}_{\mathsf{OT}}^E$ and receives $\boldsymbol{v}_{i,x} \leftarrow \{0,1\}^\ell$ in response. $\mathcal{A}'$ programs H to output $\boldsymbol{v}_{i,x}$ on this query. All other H queries are answered as normal. The distribution of H after being programmed is identical since the input has not previously been queried and in both cases the result is uniformly distributed.

Hybrid 5. $\mathcal{A}'$ does not take the input of S and does not program H in Hybrid 5.2. S only interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. This change is identically distributed.

□

□

Definition 5.11. *Let $\Pi^{\mathsf{ext}\text{-}S}$ be the protocol of Figure 9 where $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$.*

Lemma 5.12. *The* $\Pi^{\mathsf{ext\text{-}S}}$ *protocol realizes 1-out-of-N* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ *security.*

Proof. Follows directly from Lemma 3.1 and Lemma 5.8. □

Lemma 5.13. *The* $\Pi^{\mathsf{ext\text{-}S}}$ *protocol does not realizes 1-out-of-N* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *or* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ *security.*

Proof. Follows directly from Lemma 3.1 with Lemma F.2 for $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and F.3 for $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$. □

Definition 5.14. *Let* $\Pi^{\mathsf{ext\text{-}R}}$ *be the protocol of Figure 9 where* $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{Su}}$.

Lemma 5.15. *The* $\Pi^{\mathsf{ext\text{-}R}}$ *protocol realizes 1-out-of-N* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ *security.*

Proof. Security against a malicious receiver follows from Lemma 3.1 and Lemma 5.8.

Claim 5.16 (Malicious Sender Security)**.** $\Pi^{\mathsf{ext\text{-}E}}$ *satisfies security against a malicious sender (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ *functionality.*

Proof. The general security of this claim also follows from Lemma 3.1 and Lemma 5.8. What remains is programming the random oracle. Observe that the honest receiver uniformly chooses $\boldsymbol{t}_0^j$ and $\boldsymbol{b} \in \{0,1\}^{nc}$ is uniformly sampled by the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{Su}}$ functionality. Now observe that all of the outputs strings are of the form

$$\begin{aligned}\boldsymbol{v}_{i,j} &= \mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(x))) \\ &= \mathsf{H}(i,\ \boldsymbol{t}_i + \boldsymbol{b} \odot (c_i + \mathcal{C}(map(x)))).\end{aligned}$$

Prior to receiving the output of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{Su}}$, $\boldsymbol{t}_i$ is uniformly distributed in the view of $\mathcal{A}$. As such, $\mathcal{A}$ has negligible probability of querying strings of this form. Therefore, H can be programmed to return the ideal output of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ when $\mathcal{A}$ queries it.

The second concern is there does not exist distinct $x, x' \in [N]$ s.t.

$$\boldsymbol{b} \odot \mathcal{C}(map(x)) = \boldsymbol{b} \odot \mathcal{C}(map(x'))$$

as this would result in $\mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(x))) = \boldsymbol{v}_{i,x} = \boldsymbol{v}_{i,x'} = \mathsf{H}(i, \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(x')))$ and thereby allow $\mathcal{A}$ to distinguish. However, this happens with negligible probability as described in claim 2, hybrid 4 of Lemma 5.8. □

□

Definition 5.17. *Let* $\Pi^{\mathsf{ext\text{-}U}}$ *be the protocol of Definition 5.14 where the random oracle* H *is redefined as follows.*

1. *Let* $\mathsf{H}' : \{0,1\}^{[m] \times \mathbb{F}_2^{nc}} \to \{0,1\}^{\kappa}$ *be a random oracle.*
2. *In Step 1,* S *samples* $k \leftarrow \mathbb{F}_2^{nc}$ *sends an extractable commitment (Definition A.2)of* k *to* R.
3. *After receiving* U *in Step 2,* S *decommits to* k *to* R *who aborts is the decommitment fails.*
4. *Both parties define* $\mathsf{H}(i, x) = \mathsf{H}'(i, x + k)$.

Lemma 5.18. *The* $\Pi^{\mathsf{ext\text{-}U}}$ *protocol realizes 1-out-of-N* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ *security.*

Proof.

Claim 5.19 (Malicious Sender Security)**.** $\Pi^{\mathsf{ext\text{-}Su+}}$ *satisfies security against a malicious sender (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ *oracle.*

Proof. The simulation follows the same strategy as Lemma 5.15 except now $\mathcal{A}$ is allowed to sample k and have the parties output messages of the form $\boldsymbol{v}_{i,x} := \mathsf{H}(i, k + \boldsymbol{t}_i + \boldsymbol{b} \odot (\boldsymbol{c}_i + \mathcal{C}(map(x))))$. The simulator $\mathcal{A}'$ samples $\boldsymbol{t}_i$ uniformly at random after $\mathcal{A}$ is bound to their choice of k and therefore its easy to verify that $\mathcal{A}$ has negligible probability of querying H on such an input before receiving k. □

Claim 5.20 (Malicious Receiver Security)**.** $\Pi^{\textit{ext-Su+}}$ *satisfies security against a malicious receiver (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ *oracle.*

Proof. The simulation also follows the same strategy as Lemma 5.15 with a few key differences.

1. $\mathcal{A}'$ sends a dummy commitment in place of the commitment to k, i.e. a uniform string from the same distribution.

2. Then $\mathcal{A}'$ runs the normal simulation described by Lemma 5.15 up to the point that S would decommit to k except that $\mathcal{A}'$ does not program H as described.

3. At this point $\mathcal{A}'$ has received U in step Step 2 and $\mathcal{A}$ send a valid proof for Step 4 (by assumption or $\mathcal{A}'$ would have aborted). $\mathcal{A}'$ now uniformly samples $k \leftarrow \mathbb{F}_2^{n_\mathcal{C}}$ and programs the commitment random oracle to decommit to k. $\mathcal{A}'$ then programs H' to output the ideal output $\boldsymbol{v}_{i,x_i}$ of R for the query $\mathsf{H}'(i, k + \boldsymbol{t}_i)$. Since $k \in \mathbb{F}_2^{n_\mathcal{C}}$ is uniformly distributed in the view of $\mathcal{A}$, it follows that $\mathcal{A}$ has probability at most $q2^{-n_\mathcal{C}} \leq q2^{-\kappa} = \mathsf{negl}$ probability of querying the oracle at this point, where q is the number of queries that $\mathcal{A}$ has made.

4. $\mathcal{A}'$ then sends the decommits of k to $\mathcal{A}$ and completes the simulation as Lemma 5.15 does.

□

□

5.3 OT Extension with an Ideal Cipher

We now discuss how to efficiently implement OT extension by restricting the input domain of the random oracle H to be $\{0,1\}^{n_\mathcal{C}}$. In particular, we are interested in the 1-out-of-2 OT case where $n_\mathcal{C} = \kappa = 128$. The core motivation for OT extension in this setting is the pervasive support for hardware based implementations of AES, which we will then use as an ideal cipher to hash the output messages. In this model we design new protocols that satisfy $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-security and achieve better concrete performance than the protocols analyzed in Section 5.2. These previous protocols have required a random oracle with input domain $[m] \times \mathbb{F}_2^{n_\mathcal{C}}$ which we reduce to $\mathbb{F}_2^{n_\mathcal{C}}$ while maintaining security.

Existing implementations[Rin, Zoh16, Kel, WMK16] have either instantiated H as a strong hash function such as SHA-256 or using AES. However, in most cases[5] that we observed[Rin, Zoh16, Kel], these instantiation incorrectly reduce the input domain to $\mathbb{F}_2^{n_\mathcal{C}}$ before applying H which can lead to full loss of security. In most cases[5] the instantiation of H effectively follows the form $\mathsf{H}(i, x) = \mathsf{H}'(c_1 x + c_2 i) + c_1 x + c_2 i$ for constants $c_1 \in \mathbb{Z}_3^*, c_2 \in \mathbb{Z}_2$ where H' is either a strong cryptographic hash function or AES with a fixed and public key. Regardless of c_1, c_2, H', it is trivial to find collisions in such an instantiation. For example, let $x \in \mathbb{F}_2^{n_\mathcal{C}}$ and then it holds that $\forall i, i' \in [m], \mathsf{H}(i, x + i) = \mathsf{H}(i', x + i')$. In the context of the OT extension protocols in the previous section, this

[5]The authors of [WMK16, GKWY19] independently identified the same implementation issue concurrently to us. [WMK16] securely implement $\mathsf{H}(i, x)$ but requires twice the number of ideal cipher calls. See [GKWY19].

Protocol	Security†	Rounds	ASM	m							
				1	32	128	512	1	32	128	512
				LAN				WAN			
[CO15]	GapDH, RO→ "Rand. OT"	2	No	5	70	230	662	106	179	301	581
			Yes	2	**6**	**18**	**64**	104	115	210	272
[NP01]	DDH, RO $\rightarrow \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$	3	No	5	67	203	573	155	185	304	593
This	IDDH, RO $\rightarrow \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$	1	No	3	46	148	480	54	135	240	550
			Some	**1**	11	28	111	**53**	**75**	**110**	**225**
	LWE, RO $\rightarrow \mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ or $\mathcal{F}_{\mathsf{OT}}^{\mathsf{Su}}$	2	Yes	**1**	**6**	24	105	101	108	154	481

Figure 10: Running times in milliseconds of our OT protocols and [CO15, NP01]. ASM indicates if the implementation was written in assembly (better performance). †We emphasize that both, [NP01] and [CO15] do not give full simulation based malicious security, but only weaker security guarantees.

attack translates into a malicious receiver being able to fully break the security. For example, let $c_1 = c_2 = 1$, then R can choose T_0 such that $\boldsymbol{t}_i + i = \boldsymbol{t}_{i'} + i'$. It then holds that all the output messages of the receiver will be the same value. That is, for all $i, i' \in [m], x \in [N]$, it holds that $\boldsymbol{v}_{i,x} = \boldsymbol{v}_{i',x}$.

One solution is to implement H directly as a random oracle as opposed to first adding the input together. However, in the case of 1-out-of-2 OT this would prevent the efficient use of AES based hashing. We take a different approach by removing the requirement for inputting i into the hash function, i.e. R outputs $\mathsf{H}(\boldsymbol{t}_i)$ as opposed to $\mathsf{H}(i, \boldsymbol{t}_i)$. We prove this approach secure given that T_0 is sampled uniformly. Intuitively, this condition is sufficient due to collisions on the input to H being negligible, i.e. the set $\{\boldsymbol{t}_i + \boldsymbol{b} \oplus \mathcal{C}(x) \mid \forall i, x\}$ does not collide[6].

See Appendix F.6 for details and proofs.

6 Implementation

We give a detailed description of how to instantiate the OT protocols based on Diffie-Hellman key exchange under tigther security loss and based on Kyber in Appendix D. We implement and benchmark the optimized DH based (Appendix D.2)and the Kyber based protocol along with five implementations of our OT extension protocols . See [Rin] for source code. We then compare these two several other implementation including the Chou & Orlandi [CO15] and Naor & Pinkas [NP01] OT protocols and the chosen string variant of Keller, Orsini & Scholl [KOS15]. All protocols are in the random oracle model.

All protocols are implemented using the elliptic curve implementation of Relic Took-kit[AG] and the assembly based curve25519 of [CO15, CO]. For the OT protocol based on Kyber we adapt the [SAB+17] key exchange implementation. For our protocol we instantiate the Random Oracle using Blake2 or the hashing to curve implementation of [AG] and the Ideal Cipher using AES.

We perform experiments on a multi-core Intel Xeon processor at 2.7GHz and 256GB of RAM. Each party is given a single thread to execute on. The parties communicate over a network loopback device. We consider two settings, LAN where the parties have a 10Gbp connection and sub millisecond latency and a WAN setting where an artificial latency of 50 ms and throughput of 100Mbps is imposed on the loopback device. We consider computation security parameter $\kappa = 128$ and statistical security of $\lambda = 40$. Some of the OT protocols take advantage of code written in assembly which can significantly outperform the other c++ implementations.

We begin with the performance results for our OT protocols. These are detailed in Figure 10.

[6] Assuming $\boldsymbol{b}$ is uniformly sampled.

Protocol	Security	Total Rounds	n	m							
				2^{12}	2^{16}	2^{20}	2^{24}	2^{12}	2^{16}	2^{20}	2^{24}
				LAN				WAN			
$\Pi^{\text{ext-U}}$	$\mathcal{F}_{\text{OT}}^{\text{Su}}$,RO $\rightarrow \mathcal{F}_{\text{OT}}^{\text{U}}$	4	2^{76}	20	151	1,612	24,060	345	833	7,003	103,481
[KOS15]	$\mathcal{F}_{\text{OT}}^{\text{S}}$,RO $\rightarrow \mathcal{F}_{\text{OT}}^{\text{S}}$	5	2	28	84	640	8,361	865	1769	7,504	85,077
$\Pi^{\text{ext-R}}$	$\mathcal{F}_{\text{OT}}^{\text{Su}}$,RO $\rightarrow \mathcal{F}_{\text{OT}}^{\text{R}}$	**2**	2	14	76	610	8,224	406	700	2,488	32,315
$\Pi^{\text{ext-U}}$	$\mathcal{F}_{\text{OT}}^{\text{Su}}$,RO $\rightarrow \mathcal{F}_{\text{OT}}^{\text{U}}$	4	2	18	70	547	7,429	407	708	2,666	32,856
$\Pi^{\text{ext-R}\pi}$	$\mathcal{F}_{\text{OT}}^{\text{Uu}}$,IC $\rightarrow \mathcal{F}_{\text{OT}}^{\text{R}}$	3	2	14	22	174	1,158	**300**	**530**	**2,097**	**25,701**
$\Pi^{\text{ext-U}\pi}$	$\mathcal{F}_{\text{OT}}^{\text{Uu}}$,IC $\rightarrow \mathcal{F}_{\text{OT}}^{\text{U}}$	5	2	**6**	**24**	**101**	**720**	395	645	2,128	26,256

Figure 11: Running times in milliseconds of our 1-out-of-n OT extension protocols and [KOS15] as implemented by [Rin]. Base OT running times are *not* included. RO indicates that a random oracle is used to has while IC *additionally* indicates an ideal cipher was used in the Davie-Meyer compression function, see Section 5.2. Rounds includes the rounds required for base OTs.

Interestingly, our two protocols are each more efficient than the other depending on the network setting. The Kyber based protocol protocol is most efficient in the LAN setting. This is due to the highly efficient operations which essentially comprise of linear algebra. However, the public keys and encryptions that are send in the Kyber based protocol are 40 times larger than the DH based OT. For example, a single OT using Kyber requires a total of 5,934 bytes while the DH protocol requires 145 bytes. In the LAN setting this added communication has little impact. To perform 128 OT the Kyber implementation requires 24 milliseconds while our DH based approach takes 25 milliseconds (when using an assembly based implementation). We note that one would rarely perform a different number of OTs than 128 which are used as a seed for OT extension.

In the WAN setting the decreased round complexity and communication of the DH approach allows it to achieve the smallest running times. To perform 128 OTs the DH protocol requires 130 milliseconds while the Kyber protocol requires 154 milliseconds. However, this increased performance comes at the expense of only achieving $\mathcal{F}_{\text{OT}}^{\text{E}}$-security.

We also compare against the protocol of [CO15] and [NP01]. In the LAN setting the fast protocol is that of [CO15] which requires each party to performance an amortized three exponentiation per OT while our DH protocol requires four. Both require five in the worst case. However, the [CO15] approach suffers from a technical issue in the proof where the input of the receiver can not be extracted at the appropriate time. As a result, to compose this protocol with OT extension requires additional computation and rounds of communication which we do not consider in our comparison, e.g. see [DKLs18, Appendix A]. In addition, our hash to group implementation which takes up the majority of the running time difference is *not* written in assembly. We suspect the gap between us and [CO15] narrow significantly if ours was fully optimized. We also note that our protocol achieve full endemic security in just one round or sender chosen message in two rounds.

Regardless, in the WAN setting with 128 OT our protocols achieve the best performance of 110ms (DH) and 154ms (Kyber) compared to 210ms by [CO15]. To achieve simulation bases security, at least one more round of communication is requires which would bring their time to at least 260ms, a 2.3× increase compared to our protocol. To perform a single OT, our DDH protocol requires 145 bytes of communication, our Kyber protocol requires 5,934 bytes, [CO15] requires the least with 112 bytes and [NP01] requires 165 bytes.

We now turn our attention to the OT extension performance result as shown in Figure 11. We compare our protocols to the fastest implementation [KOS15, Rin]. In particular, we update the [Rin] implementation of [KOS15] to allow the sender to specify the output messages and include the index i in the call to the random oracle H, as specified by [KOS15]. . We implement five variants of our proposals where the output strings are sampled by the protocol (or possibly a malicious

party). The 1-out-of-2 $\Pi^{\mathsf{ext\text{-}R}}, \Pi^{\mathsf{ext\text{-}R}\pi}$ protocols in the random oracle and ideal cipher model require 2 and 3 rounds of communication, respectively, and achieves $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$-security. To reduce the rounds we apply the Fiat-Shamir transformation [FS87] to the sigma protocol of [KOS15] for Step 4. We also implement the $\Pi^{\mathsf{ext\text{-}U}}$ and $\Pi^{\mathsf{ext\text{-}U}\pi}$ which both achieve $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-security.

In the LAN setting where communication and round complexity has little impact the fastest protocol is our $\Pi^{\mathsf{ext\text{-}U}\pi}$ protocol which achieves our strongest security notion. The performance of this approach derives from the exclusive use of AES in the protocol which has extremely fast hardware support. For example, $\Pi^{\mathsf{ext\text{-}U}\pi}$ is $10\times$ faster compared to the $\Pi^{\mathsf{ext\text{-}U}}$ protocol which achieves the same $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-security in the random oracle model. The next fastest protocol is $\Pi^{\mathsf{ext\text{-}R}\pi}$ which achieves $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$-security in the ideal cipher model. This protocol only requires 3 rounds of communication which allows it to be the most efficient in the WAN setting. However, the improved round complexity requires hashing the transcript of the protocol which decrease the running time in the LAN setting compared to $\Pi^{\mathsf{ext\text{-}U}\pi}$.

In the random oracle model we implement the $\Pi^{\mathsf{ext\text{-}R}}$ protocol which only requires 2 rounds of communication, including the base OTs. In particular, the OT extension sender sends the first base OT message and the receiver sends the second base OT message, the extension matrix U and Fiat-Shamir proof of consistency. One short coming of this approach is that $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$-security is achieved. However, we argue that this level of security could be sufficient for special purpose protocols which require both high performance and low round complexity. We also implement the $\Pi^{\mathsf{ext\text{-}U}}$ protocol which requires two more rounds and does not apply the Fiat-Shamir transformation which allows improved performance in the LAN setting at the expense of worse performance in the WAN setting. This protocol is also implemented for 1-out-of-2^{76} OT where the sender computes three of the OT strings.

For a point of comparison we benchmark the [KOS15] protocol and find that our protocols are between 3 and 8 times more efficient, depending on the network setting. Our performance improvements stem from the use of of the Ideal Cipher model (AES) and that fact that our protocols output random strings where as the secure version of [KOS15] requires the sender to send encrypted strings. This effectively triples the communication overhead and adds an additional round to the protocol. In particular, all of our extension protocols require an amortized κ bits of communication per 1-out-of-2 OT while [KOS15] requires 3κ.

Acknowledgements

We thank Jonathan Katz and Mike Rosulek for pointing out flaws with the UC security in the original paper. In this version, we fixed flaws and removed wrong claims. This mainly affects UC security of our one-round OT constructions. It is an open question to obtain UC secure one-round endemic OT from uniform key agreement. For our one-round UC OT from cyclic group based assumptions, we seem to require significantly stronger assumptions than previously claimed.

We thank Sebastian R. Verschoor for pointing out that we do not need the correctness of the key exchange protocol to prove security against a malicious receiver or sender. This is in particular significant for LWE based OTs where malicious parties can cause decryption errors.

References

[AG] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic.

[ALSZ15] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions with security for malicious adversaries. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 673–701. Springer, Heidelberg, April 2015.

[ALSZ17] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer extensions. *Journal of Cryptology*, 30(3):805–858, July 2017.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistically sender-private OT from LWE. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 370–390. Springer, Heidelberg, November 2018.

[BDD+17] Paulo S. L. M. Barreto, Bernardo David, Rafael Dowsley, Kirill Morozov, and Anderson C. A. Nascimento. A framework for efficient adaptively secure composable oblivious transfer in the ROM. Cryptology ePrint Archive, Report 2017/993, 2017. http://eprint.iacr.org/2017/993.

[BDK+17] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. CRYSTALS – kyber: a CCA-secure module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634, 2017. http://eprint.iacr.org/2017/634.

[Bea96] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *28th ACM STOC*, pages 479–488. ACM Press, May 1996.

[BL18] Fabrice Benhamouda and Huijia Lin. k-round multiparty computation from k-round oblivious transfer via garbled interactive circuits. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 500–532. Springer, Heidelberg, April / May 2018.

[BM90] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 547–557. Springer, Heidelberg, August 1990.

[BPRS17] Megha Byali, Arpita Patra, Divya Ravi, and Pratik Sarkar. Fast and universally-composable oblivious transfer and commitment scheme with adaptive security. Cryptology ePrint Archive, Report 2017/1165, 2017. https://eprint.iacr.org/2017/1165.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 320–335. Springer, Heidelberg, August 2002.

[CCL15] Ran Canetti, Asaf Cohen, and Yehuda Lindell. A simpler variant of universally composable security for standard multiparty computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 3–22. Springer, Heidelberg, August 2015.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CO] Tung Chou and Claudio Orlandi. The Simplest Oblivious Transfer Protocol. http://users-cs.au.dk/orlandi/simpleOT/.

[CO15] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In Kristin E. Lauter and Francisco Rodr´iguez-Henr´iquez, editors, *LATINCRYPT 2015*, volume 9230 of *LNCS*, pages 40–58. Springer, Heidelberg, August 2015.

[CvT95] Claude Crépeau, Jeroen van de Graaf, and Alain Tapp. Committed oblivious transfer and private multi-party computation. In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 110–123. Springer, Heidelberg, August 1995.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DKLs18] Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Secure two-party threshold ECDSA from ECDSA assumptions. In *2018 IEEE Symposium on Security and Privacy*, pages 980–997. IEEE Computer Society Press, May 2018.

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 205–210. Plenum Press, New York, USA, 1982.

[FMV18] Daniele Friolo, Daniel Masny, and Daniele Venturi. Secure multi-party computation from strongly uniform key agreement. Cryptology ePrint Archive, Report 2018/473, 2018. https://eprint.iacr.org/2018/473.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

[GIS18] Sanjam Garg, Yuval Ishai, and Akshayaram Srinivasan. Two-round MPC: Information-theoretic and black-box. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 123–151. Springer, Heidelberg, November 2018.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335. IEEE Computer Society Press, November 2000.

[GKWY19] Chun Guo, Jonathan Katz, Xiao Wang, and Yu Yu. Efficient and secure multiparty computation from fixed-key block ciphers. *IACR Cryptology ePrint Archive*, 2019:74, 2019.

[GS18] Sanjam Garg and Akshayaram Srinivasan. Two-round multiparty secure computation from minimal assumptions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 468–499. Springer, Heidelberg, April / May 2018.

[HL17] Eduard Hauck and Julian Loss. Efficient and universally composable protocols for oblivious transfer from the cdh assumption. Cryptology ePrint Archive, Report 2017/1011, 2017. https://eprint.iacr.org/2017/1011.

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 145–161. Springer, Heidelberg, August 2003.

[IKO+11] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, Manoj Prabhakaran, and Amit Sahai. Efficient non-interactive secure computation. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 406–425. Springer, Heidelberg, May 2011.

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

[Kel] Keller, Marcel and Orsini, Emmanuela and Scholl, Peter. APRICOT OT Extension. https://github.com/bristolcrypto/apricot.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *20th ACM STOC*, pages 20–31. ACM Press, May 1988.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

[KOS15] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure OT extension with optimal overhead. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 724–741. Springer, Heidelberg, August 2015.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *12th SODA*, pages 448–457. ACM-SIAM, January 2001.

[OOS17] Michele Orrù, Emmanuela Orsini, and Peter Scholl. Actively secure 1-out-of-N OT extension with application to private set intersection. In Helena Handschuh, editor, *CT-RSA 2017*, volume 10159 of *LNCS*, pages 381–396. Springer, Heidelberg, February 2017.

[ORS15] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 339–358. Springer, Heidelberg, August 2015.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, Heidelberg, August 2008.

[Rab81] Michael O. Rabin. How to exchange secrets by oblivious transfer. Technical report, Harvard University, 1981.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Rin] Peter Rindal. libOTe: an efficient, portable, and easy to use Oblivious Transfer Library. https://github.com/osu-crypto/libOTe.

[RR17a] Peter Rindal and Mike Rosulek. Improved private set intersection against malicious adversaries. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 235–259. Springer, Heidelberg, April / May 2017.

[RR17b] Peter Rindal and Mike Rosulek. Malicious-secure private set intersection via dual execution. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17*, pages 1229–1242. ACM Press, October / November 2017.

[SAB+17] Peter Schwabe, Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehle. Crystals-kyber. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[Win84] R. S. Winternitz. A secure one-way hash function built from des. In *1984 IEEE Symposium on Security and Privacy*, pages 88–88, April 1984.

[WMK16] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, 2016.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd FOCS*, pages 160–164. IEEE Computer Society Press, November 1982.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986.

[Zoh16] Michael Zohner. Encrypto Group OTExtension . https://github.com/encryptogroup/OTExtension, 2016.

A Additional Preliminary Definitions and Lemmata

Definition A.1 (Coin Tossing)**.** *An* ideal coin tossing *is a functionality denoted with* $\mathcal{F}^{\mathsf{coin}}$ *that interacts with two parties* A *and* B*, samples a uniform string* $r \in \{0,1\}^*$ *and sends* r *to* A *and* B*.*

Definition A.2 (Extractable Commitments)**.** *An* extractable commitment scheme *consists of three algorithms.*

$\mathsf{com}(x, r)$**:** *Commits to* x *using randomness* r*.*

$\mathsf{open}(\mathsf{com}, x, r)$**:** *Outputs* 1 *if commitment* $\mathsf{com} \in \mathsf{com}(x, r)$*.*

$\mathsf{ext}(\mathsf{com}, \mathsf{aux})$**:** *Given some auxiliary information, it extracts committed value* x*.*

For security, we ask that it is hiding, i.e. for any x, m*,* $x, \mathsf{com}(x, r)$ *is indistinguishable from* $m, \mathsf{com}(x, r)$ *and that it is binding, i.e. for any* x*,* $\mathsf{ext}(\mathsf{com}(x, r))$ *outputs* x*.*

An extractable commitment can easily constructed using a random oracle by defining $\mathsf{com}(x, r) := H(x, r)$, open simply evaluates H and checks equality and the ext algorithm observes the random oracle queries from which x can be learned.

A.1 Key Agreement

We give the following additional security definition for key agreement protocols.

Definition A.3 (One-Round Uniform Key Agreement)**.** *We call a* UKA one-round uniform key agreement *if the function* $\mathsf{m_B} \leftarrow \mathsf{B}(\mathsf{t_B}, \mathsf{m_A})$ *does not depend on* m_A *and can be computed solely using input* $\mathsf{t_B}$*. More precisely, there is a function* B' *such that for any* $\mathsf{m_A}$, $\mathsf{B}'(\mathsf{t_B}) = \mathsf{B}(\mathsf{t_B}, \mathsf{m_A})$*, which we will in the following refer to with* B *as well.*

Definition A.4 (Multi-Instance Uniformity)**.** *We call a* UKA Q-multi-instance ϵ-uniform *if for any ppt distinguisher* D *and any polynomial size auxiliary input* z*,*

$$|\Pr[\mathsf{D}^{\mathcal{O}_\mathsf{A}}(z)] = 1] - Pr[\mathsf{D}^{\mathcal{O}_u}(z) = 1]| \leq \epsilon,$$

where $\mathcal{O}_\mathsf{A}$ *outputs* $\mathsf{m_A} \leftarrow \mathsf{A}(\mathsf{t_A})$ *for fresh randomness* $\mathsf{t_A}$ *and* $\mathcal{O}_u$ *outputs* $u \leftarrow \mathcal{G}$ *and* Q *is a bound on the amount of queries to* $\mathcal{O}_u$*,* $\mathcal{O}_\mathsf{A}$*.*

Definition A.5 (Multi-Instance Key-Indistinguishability)**.** *We call a* UKA (Q, n)-multi-instance ϵ-key-indistinguishable *if for any ppt distinguisher* D *and any polynomial size auxiliary input* z*,*

$$|\Pr[\mathsf{D}^{\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle},\mathcal{O}_\mathsf{k}}(z)] = 1] - Pr[\mathsf{D}^{\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle},\mathcal{O}_u}(z) = 1]| \leq \epsilon,$$

where $\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle}$ *outputs on the* i*-th query a transcript* $\mathsf{T}_i := \langle \mathsf{A}_i, \mathsf{B}_i \rangle$*,* $\mathcal{O}_\mathsf{k}$ *outputs on query* j*, key* $\mathsf{k}_j = \mathsf{Key}(\mathsf{t_{A},}_j, \mathsf{m_{B},}_j) = \mathsf{Key}(\mathsf{t_{B},}_j, \mathsf{m_{A},}_j)$ *that matches transcript* T_i*.* $\mathcal{O}_u$ *outputs a uniform element* u *from the key domain.* $\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle}$ *uses fresh random tapes* $\mathsf{t_{A},}_i, \mathsf{t_{B},}_i \leftarrow \{0,1\}^*$ *for every query.* Q *is a bound on the amount of queries to* $\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle}$*, where* n *bounds the amount of queries to* $\mathcal{O}_\mathsf{k}$*,* $\mathcal{O}_u$*.*

In case of a one-round UKA, we define a stronger version of the multi-instance key-indistinguishability, which we call one-round multi-instance key-indistinguishability.

Definition A.6 (One-Round Multi-Instance Key-Indistinguishability)**.** *We call a one-round* UKA one-round (Q, n)-multi-instance ϵ-key-indistinguishability *if for any ppt distinguisher* D *and any polynomial size auxiliary input* z*,*

$$|\Pr[\mathsf{D}^{\mathcal{O}_\mathsf{A},\mathcal{O}_\mathsf{k}}(z, \mathsf{m_B})] = 1] - Pr[\mathsf{D}^{\mathcal{O}_\mathsf{A},\mathcal{O}_u}(z, \mathsf{m_B}) = 1]| \leq \epsilon,$$

where $\mathsf{m_B} \leftarrow \mathsf{B}(\mathsf{t_B})$ *for uniform* $\mathsf{t_B}$*.* $\mathcal{O}_\mathsf{A}$ *outputs on the* i*-th query* $\mathsf{m_{A},}_i \leftarrow \mathsf{A}(\mathsf{t_{A},}_i)$ *for uniform* $\mathsf{t_{A},}_i$*.* $\mathcal{O}_\mathsf{k}$ *outputs on query* j*, key* $\mathsf{k}_j = \mathsf{Key}(\mathsf{t_{A},}_j, \mathsf{m_B}) = \mathsf{Key}(\mathsf{t_B}, \mathsf{m_A})$*.* $\mathcal{O}_u$ *outputs a uniform element* u *from the key domain.* Q *is a bound on the amount of queries to* $\mathcal{O}_\mathsf{A}$*, where* n *bounds the amount of queries to* $\mathcal{O}_\mathsf{k}$*,* $\mathcal{O}_u$*.*

In the next lemmata, we show that all the security notions are implied by the standard definitions, but potentially with a polynomial security loss.

Lemma A.7. *Let* UKA *be* ϵ*-uniform, then it is* Q*-multi-instance* $Q\epsilon$*-uniform.*

Proof. This follows straightforwardly from using a simple hybrid argument. Hybrid hyb_i samples $\mathsf{m_{A},}_j$ for $j \leq i$ from $\mathcal{O}_u$ and for $j > i$ from $\mathcal{O}_\mathsf{A}$. If there is an adversary that distinguishes hyb_i from hyb_{i+1} for any i, then we can break the uniformity of UKA by distinguishing $\mathsf{m_{A},}_{i+1}$ from uniform. □

Lemma A.8. *Let* UKA *be* ϵ*-key-indistinguishable, then it is* (Q, n)*-multi-instance* $Q\epsilon$*-key-indistinguishable.*

Proof. Again, we use a hybrid argument over hybrids hyb_i. In hyb_i, $(\mathsf{m}_{\mathsf{A},j}, \mathsf{m}_{\mathsf{B},j}), \mathsf{k}_j$ is sampled from $\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle} \times \mathcal{O}_u$ for $j \leq i$ and from $\mathcal{O}_{\langle \mathsf{A},\mathsf{B}\rangle} \times \mathcal{O}_{\mathsf{k}}$ for $j > i$. If one distinguishes hyb_i from hyb_{i+1} for some i, one breaks the key-indistinguishability. □

Lemma A.9. *Let a one-round* UKA *be* ϵ*-key-indistinguishable, then it is one-round* (Q, n)*-multi-instance* $Q\epsilon$*-key-indistinguishable.*

Proof. This lemma follows for the same reason as Lemma A.8. □

A.2 Oblivious Transfer

Lemma A.10 (Repeat of Lemma 3.1)**.** *Let the distribution of OT strings be efficiently sampleable. Then* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$*-security implies* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *as well as* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*-security.* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *or* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*-security imply* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*-security.*

Proof. In the first step, we show that uniform message security implies sender chosen message security and receiver chosen message security implies endemic security. These two implications result from the same simple fact that a malicious sender interacting with the ideal OT is easier to construct when it can choose the OT strings than when it receives the strings from the ideal OT. The following claim formalizes this fact.

Claim A.11. *Let* Π *be an OT secure against a malicious sender with respect to an ideal OT* $\mathcal{F}_{\mathsf{OT}}^{*}$ *that sends the OT strings* $(s_i)_{i\in[n]}$ *to the sender, i.e. functionality* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ *and* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$*, and the distribution of* $(s_i)_{i\in[n]}$ *is efficiently sampleable. Then* Π *is also secure against a malicious sender with respect to ideal OT* $\mathcal{F}_{\mathsf{OT}}$*, which receives the OT strings* $(s_i)_{i\in[n]}$ *from the sender, i.e functionality* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *and* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$*.*

Proof. We show that if there is an adversary that breaks the security against a malicious security with respect to ideal OT $\mathcal{F}_{\mathsf{OT}}$ then there is also an adversary that breaks the security with respect to $\mathcal{F}_{\mathsf{OT}}^{*}$. More precisely, if there is a ppt adversary $\mathcal{A}_1$ such that for any ppt adversary $\mathcal{A}_1'$ there exists a ppt distinguisher D_1 and a polynomial size auxiliary input z with

$$|\Pr[\mathsf{D}_1(z, (\mathcal{A}_1, \mathsf{R})_\Pi) = 1] - \Pr[\mathsf{D}_1(z, (\mathcal{A}_1', \mathcal{F}_{\mathsf{OT}})) = 1]| = \epsilon,$$

where all algorithms receive input 1^κ and R additionally receives input $\mathbb{S}$. Then there is also a ppt adversary $\mathcal{A}_2$ such that for any ppt adversary $\mathcal{A}_2'$ there exists a ppt distinguisher D_2 and a polynomial size auxiliary input z with

$$|\Pr[\mathsf{D}_2(z, (\mathcal{A}_2, \mathsf{R})_\Pi) = 1] - \Pr[\mathsf{D}_1(z, (\mathcal{A}_2', \mathcal{F}_{\mathsf{OT}}^{*})) = 1]| = \epsilon,$$

where all algorithms receive input 1^κ and R additionally receives input $\mathbb{S}$.

We set $\mathcal{A}_2 := \mathcal{A}_1$ and $\mathsf{D}_2 := \mathsf{D}_1$. Further, for any $\mathcal{A}_2'$, there is an $\mathcal{A}_1'$ such that the distribution of $(\mathcal{A}_2', \mathcal{F}_{\mathsf{OT}}^{*})$ is identical with the distribution $(\mathcal{A}_1', \mathcal{F}_{\mathsf{OT}})$. This follows from the fact that $\mathcal{A}_1'$ could choose the OT strings $(s_i)_{i\in[n]}$ from the same distribution as $\mathcal{F}_{\mathsf{OT}}^{*}$ does and otherwise follow the description of $\mathcal{A}_2'$. Since D_1 is successful for any $\mathcal{A}_1'$ it will be also for any $\mathcal{A}_2'$, which can be seen as a subset of the set of all ppt adversaries $\mathcal{A}_1'$. □

The remaining two implications, from uniform security to receiver chosen message security and from sender chosen message security to endemic security follow in a similar fashion. Again it is easier to construct a malicious receiver interacting with the ideal OT when he can choose the OT strings rather than receiving them from the ideal OT.

Claim A.12. *Let Π be an OT secure against a malicious receiver with respect to an ideal OT $\mathcal{F}^*_{\mathsf{OT}}$ that sends the learned OT strings $(s_i)_{i\in\mathbb{S}}$ to the receiver, i.e. functionality $\mathcal{F}^{\mathsf{U}}_{\mathsf{OT}}$ and $\mathcal{F}^{\mathsf{S}}_{\mathsf{OT}}$, and the distribution of $(s_i)_{i\in\mathbb{S}}$ is efficiently sampleable. Then Π is also secure against a malicious sender with respect to ideal OT $\mathcal{F}_{\mathsf{OT}}$, which receives the OT strings $(s_i)_{i\in\mathbb{S}}$ from the receiver, i.e. $\mathcal{F}^{\mathsf{R}}_{\mathsf{OT}}$ and $\mathcal{F}^{\mathsf{E}}_{\mathsf{OT}}$.*

Proof. The proof is basically identical to the proof of Claim A.11. Again, the set of all ppt $\mathcal{A}'_2$ is a subset of the set of all ppt $\mathcal{A}'_1$ and identical with the set of all $\mathcal{A}'_1$ that sample $(s_i)_{i\in\mathbb{S}}$ from the same distribution as when sent by $\mathcal{F}^*_{\mathsf{OT}}$. □

□

In the following, we give a generalized definition of OT.

Definition A.13 (Generalize Ideal k-out-of-n Oblivious Transfer)**.** *A (generalized)* ideal k-out-of-n oblivious transfer *is a functionality that interacts with two parties, a sender S and a receiver R. Let $\mathbb{S}\subseteq[n]$ of size k and $s_1,...,s_n\in\{0,1\}^\ell$.*

The functionality is publicly parameterized by one of the following message sampling methods:

Sender Chosen Message: S *sends the circuit $\mathcal{M}:[n]\to\{0,1\}^\ell$ to the functionality which defines $s_i:=\mathcal{M}(i)$.*

Receiver Chosen Message: R *sends the circuit $\mathcal{M}:[k]\to\{0,1\}^\ell$ to the functionality which defines $s_{\mathbb{S}_i}:=\mathcal{M}(i)$ for $i\in[k]$ and uniformly samples $s_i\leftarrow\{0,1\}^\ell$ for $i\in\mathbb{S}\setminus[n]$.*

Uniform Message: *The functionality uniformly samples $s_i\leftarrow\{0,1\}^\ell$ for $i\in[n]$.*

Endemic: *If S is corrupt, then S sends the circuit $\mathcal{M}:[n]\to\{0,1\}^\ell$ to the functionality which defines $s_i:=\mathcal{M}(i)$.*

If R is corrupt, R sends the circuit $\mathcal{M}:[k]\to\{0,1\}^\ell$ to the functionality which defines $s_{\mathbb{S}_i}:=\mathcal{M}(i)$ for $i\in[k]$.

All remaining s_i for $i\in[n]$ are uniformly samples $s_i\leftarrow\{0,1\}^\ell$.

The functionality is publicly parameterized by one of the following selection methods:

Receiver Selection: R *sends the circuit $\mathcal{S}:[n]\to\{0,1\}$ to the functionality where the support of $\mathcal{S}$ is of size k. The functionality defines $\mathbb{S}:=\{i\mid\mathcal{S}(i)=1\}$.*

Uniform Selection: *The functionality uniformly samples $\mathbb{S}\leftarrow\mathbb{P}([n])$ s.t. $|\mathbb{S}|=k$.*

As specified by the message sampling method, the oracle receives the circuit $\mathcal{M}$ from the appropriate party if one is called for. As specified by the selection method, the functionality receives the circuit $\mathcal{S}$ if one is called for. Thereafter, upon receiving the message (Output,i) from S, respond with s_i. Upon receiving (Output,i) from R and if $i\in\mathbb{S}$, respond with s_i.

We denote the ideal functionalities for sender chosen, receiver chosen, uniform and endemic with receiver selection as $\mathcal{F}^{\mathsf{S}}_{\mathsf{OT}},\mathcal{F}^{\mathsf{R}}_{\mathsf{OT}},\mathcal{F}^{\mathsf{U}}_{\mathsf{OT}},\mathcal{F}^{\mathsf{E}}_{\mathsf{OT}}$, respectively. The analogous oracles for Uniform Selection are denoted as $\mathcal{F}^{\mathsf{S}u}_{\mathsf{OT}}$, $\mathcal{F}^{\mathsf{R}u}_{\mathsf{OT}},\mathcal{F}^{\mathsf{U}u}_{\mathsf{OT}},\mathcal{F}^{\mathsf{E}u}_{\mathsf{OT}}$, respectively.

Remark A.14. *When n is polynomial in the security parameter κ, we simplify the above definition to Definition 2.4 to allow the parties directly input the appropriate s_i messages as opposed to specifying a circuit $\mathcal{M}$. Similarly for the set $\mathbb{S}:=\{i\mid\mathcal{S}(i)=1\}$. Lastly, instead of querying the oracle with (Output,i), the oracle sends $(s_i)_{i\in[n]}$ to S and $(\mathbb{S},(s_i)_{i\in\mathbb{S}})$ to R. This simplification can trivially be simulated when $n=\mathsf{poly}(\kappa)$.*

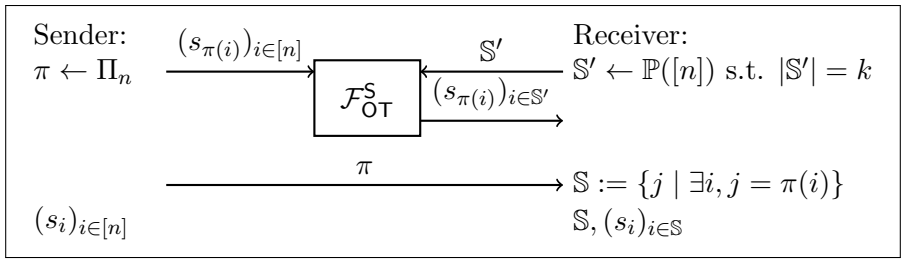

Figure 12: Uniform selection k-out-of-n OT protocol $\Pi^{\mathsf{S}u}$ in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ hybrid. Π_n is the set of permutations over $[n]$.

The following transformation allows to transform an OT where the receiver's choice bit is chosen to an OT with a random choice bit. This transformation is very useful in the context of OT extension.

Lemma A.15. *$\Pi^{\mathsf{S}u}$ of Figure 12 realizes the ideal uniform selection sender chosen message OT $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}u}$ (Definition 2.4) with unconditional security in the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ hybrid.*

Remark A.16. *The same transformation applies to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ except S does not input anything to $\mathcal{F}_{\mathsf{OT}}^{*}$.*

sketch. Consider a corrupt S. Due to $\mathbb{S}'$ being uniformly distributed, so is $\mathbb{S} = \pi(\mathbb{S}')$ since π is one to one. Consider a corrupt R. The simulator receives $\mathbb{S}'$ from R and the $\mathbb{S}, (s_i)_{i\in\mathbb{S}}$ from $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}u}$. The simulator uniformly samples π s.t. $\{s_i\}_{i\in\mathbb{S}} = \{s_{\pi(i)}\}_{i\in\mathbb{S}'}$ and completes the protocol. □

B Lower Bound on the Round Complexity of Sender and Receiver Chosen Message Security

In Lemma 3.8, we state that there cannot be an two message OT that achieves sender chosen message security where the sender sends its message first. Here we give the proof.

Proof. We show that the most general notion of OT, one out of two OT is impossible. For a two message OT where the sender sends its message first, the sender's message $\mathsf{m_S}$ is a function f_{S} on input $\mathsf{t_S}$ and some auxiliary input aux. A sender could sample s_0, s_1 during the protocol or receive them as input. We use aux to cover the second case. Further, there is a function f_{R} that takes the random tape $\mathsf{t_R}$, $\mathsf{m_S}$ and choice bit b of R. Finally, there are two functions $f_{\mathsf{OT,S}}(\mathsf{t_S}, \mathsf{m_R}, \mathsf{aux})$ that outputs (s_0, s_1) and $f_{\mathsf{OT,R}}(\mathsf{t_R}, \mathsf{m_S}, b)$ that outputs s_b.

First, we assume that S is committed to (s_0, s_1) given $\mathsf{m_S}$, i.e. there is a (s_0, s_1) such that

$$\Pr_{\mathsf{m_R},(\mathsf{t_S},\mathsf{aux})}[f_{\mathsf{OT,S}}(\mathsf{t_S}, \mathsf{m_R}, \mathsf{aux}) = (s_0, s_1) \mid \mathsf{m_S} = f_{\mathsf{S}}(\mathsf{t_S}, \mathsf{aux})] \geq \frac{3}{4}.$$

In this case, a malicious receiver can break the security as follows. It selects two random tapes $\mathsf{t}_{\mathsf{R},1}$, $\mathsf{t}_{\mathsf{R},2}$, two choice bits $b_1 = 0$, $b_2 = 1$ and computes for all $i \in [2]$, $\mathsf{m}_{\mathsf{R},i} = f_{\mathsf{R}}(\mathsf{t}_{\mathsf{R},i}, \mathsf{m_S}, b_i)$ and $s_{b_i,i} = f_{\mathsf{OT,R}}(\mathsf{t}_{\mathsf{R},i}, \mathsf{m_S}, b_i)$. It outputs $(s_{0,1}, s_{1,2})$ as a guess for s_0, s_1.

Let the scheme be δ correct, for $\delta \geq 1 - \mathsf{negl}$. Then, the probability that the first malicious receiver reconstructs (s_0, s_1) correctly is lower bounded using Jensen's inequality by

$$\Pr[(s_{0,1}, s_{1,2}) = (s_0, s_1)] \geq \left(\frac{3}{4}\delta\right)^2 > \frac{1}{2},$$

where $(s_0, s_1) = f_{\mathsf{OT},\mathsf{S}}(\mathsf{t_S}, \mathsf{m_R}, \mathsf{aux})$. A malicious receiver interacting with the ideal OT can achieve this at most with probability $\frac{1}{2}$. Hence, there is a distinguisher that breaks the sender chosen message security of the OT.

Now assume that for any (s_0, s_1),

$$\Pr_{\mathsf{m_R},\mathsf{t_S}}[f_{\mathsf{OT},\mathsf{S}}(\mathsf{t_S}, \mathsf{m_R}, \mathsf{aux}) = (s_0, s_1) \mid \mathsf{m_S} = f_{\mathsf{S}}(\mathsf{t_S}, \mathsf{aux})] < \frac{3}{4}. \tag{1}$$

In this case, we show that a malicious receiver can tweak the distribution of s_0, s_1. The malicious receiver uses a hardwired pseudorandom function (PRF) key k for a PRF PRF that outputs a single bit.[7] The malicious receiver samples two random tapes $\mathsf{t_{R,1}}$, $\mathsf{t_{R,2}}$, two uniform choice bits b_1, b_2, computes for all $i \in [2]$ $\mathsf{m}_{\mathsf{R},i} = f_{\mathsf{R}}(\mathsf{t}_{\mathsf{R},i}, \mathsf{m_S}, b_i)$ and $s_{b_i,i} = f_{\mathsf{OT},\mathsf{R}}(\mathsf{t}_{\mathsf{R},i}, \mathsf{m_S}, b_i)$. If $\mathsf{PRF}_{\mathsf{k}}(s_{b_1,1}) = 0$ it sends $\mathsf{m}_{\mathsf{R},1}$ to S and outputs $s_{b_1,1}$ otherwise it sends $\mathsf{m}_{\mathsf{R},2}$ to S and outputs $s_{b_2,2}$

We first give a bound on the probability that $\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0$. Let ϵ_{PRF} be a probability such that

$$\Pr[\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0] = \frac{1}{2} - \epsilon_{\mathsf{PRF}}.$$

Then there is a distinguisher D that simply outputs 1 if $\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0$. Hence,

$$|\Pr[\mathsf{D}(1^\kappa, s_{b_i,i}, \mathsf{PRF}_{\mathsf{k}}(s_{b_i,i})) = 1] - \Pr[\mathsf{D}(1^\kappa, s_{b_i,i}, u) = 1]| = \epsilon_{\mathsf{PRF}},$$

where $u \leftarrow \{0,1\}$. Since D breaks PRF with probability ϵ_{PRF} and PRF is secure, ϵ_{PRF} is negligible. By (1), it holds with at least probability $\frac{3}{16}$ that $s_{b_1,1} \neq s_{b_2,2}$. Therefore, the probability that for the output $s_{b_i,i}$ of the malicious holds $\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0$ is at least

$$\begin{aligned}
&\Pr[\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0] \\
&= \Pr[\mathsf{PRF}_{\mathsf{k}}(s_{b_1,1}) = 0] + \Pr[\mathsf{PRF}_{\mathsf{k}}(s_{b_1,1}) = 1 \wedge \mathsf{PRF}_{\mathsf{k}}(s_{b_2,2}) = 0] \\
&\geq \left(\frac{1}{2} - \epsilon_{\mathsf{PRF}}\right) + \left(\frac{1}{2} + \epsilon_{\mathsf{PRF}}\right) \cdot \frac{3}{16}\left(\frac{1}{2} - \epsilon_{\mathsf{PRF}}\right) \\
&= \frac{1}{2} + \frac{1}{64} - \mathsf{negl}.
\end{aligned}$$

Since the OT is $\delta = 1 - \mathsf{negl}$ correct, we get by using a union bound that the malicious receivers output $s_{b_i,i}$ is correct and $\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0$ holds at least with probability $\frac{1}{2} + \frac{1}{64} - \mathsf{negl}$. Hence, $\mathsf{PRF}_{\mathsf{k}}(s_{b_i,i}) = 0$ holds with a noticeable bias - given k- when the malicious receiver interacts with an honest sender. If there is a distribution of (s_0, s_1) from which the ideal OT samples with the same bias, then there is a distinguisher D that breaks the security of PRF. D samples from this distribution, queries PRF on the samples and outputs 1 if the query returns 0.

Since there is no such a distribution for a secure PRF, there is no adversary $\mathcal{A}'$ that creates the same output distribution when interacting with the ideal OT as when the malicious receiver interacts with the honest sender. Hence, the OT is not sender chosen message secure. □

The following proof is the proof for Lemma 3.9, which states that there cannot be an two message OT that achieves receiver chosen message security where the receiver sends its message first.

Proof. Again, we rule out the most general notion of OT, one out of two OT. We follow a similar strategy as in the previous lemma. A two message OT where the receiver sends its message first has the following structure. The receiver's message $\mathsf{m_R}$ is a function f_{R} on input $\mathsf{t_R}$, b and some

[7] OT implies one-way functions and hence also PRFs.

auxiliary input aux. Further, there is a function f_{S} that takes the random tape t_{S} and m_{R} as input. Finally, there are two functions $f_{\mathsf{OT},\mathsf{S}}(\mathsf{t}_{\mathsf{S}}, \mathsf{m}_{\mathsf{R}})$ that outputs (s_0, s_1) and $f_{\mathsf{OT},\mathsf{R}}(\mathsf{t}_{\mathsf{R}}, \mathsf{m}_{\mathsf{S}}, b, \mathsf{aux})$ that outputs s_b.

We distinguish two cases. In case one, we assume that R is committed to s_b given m_{R}, i.e. there is a s_b such that

$$\Pr_{\mathsf{m}_{\mathsf{S}},(\mathsf{t}_{\mathsf{R}},b,\mathsf{aux})}[f_{\mathsf{OT},\mathsf{R}}(\mathsf{t}_{\mathsf{R}}, \mathsf{m}_{\mathsf{S}}, b, \mathsf{aux}) = s_b \mid \mathsf{m}_{\mathsf{R}} = f_{\mathsf{R}}(\mathsf{t}_{\mathsf{R}}, b, \mathsf{aux})] = \frac{3}{4} + \alpha,$$

for $\alpha \geq 0$. Further, $s_{\bar{b}}$ should not be determined by m_{R}. Let ℓ be the length of s_0, s_1, then if there is a $s_{\bar{b}}$ s.t.

$$\Pr_{\mathsf{t}_{\mathsf{S}}}[s_{\bar{b}} \text{ is output of } f_{\mathsf{OT},\mathsf{S}}(\mathsf{t}_{\mathsf{S}}, \mathsf{m}_{\mathsf{R}}) \text{ for bit } \bar{b} \mid \mathsf{m}_{\mathsf{R}}] = \frac{1}{2^\ell} + \epsilon,$$

then a malicious receiver can sample t_{S} and compute $f_{\mathsf{OT},\mathsf{S}}(\mathsf{t}_{\mathsf{S}}, \mathsf{m}_{\mathsf{R}})$ to learn $s_{\bar{b}}$ with probability $\frac{1}{2^\ell} + \epsilon - (1 - \delta)$, where OT is $\delta = 1 - \mathsf{negl}$ correct. Since the ideal primitive samples $s_{\bar{b}}$ at uniform, the malicious receiver breaks the OT with probability $\epsilon - \mathsf{negl}$. Therefore, $\epsilon = \mathsf{negl}$.

But now, a malicious sender can sample two random tapes $\mathsf{t}_{\mathsf{S},1}$, $\mathsf{t}_{\mathsf{S},2}$ and compute for all $i \in [2]$, $(s_{0,i}, s_{1,i}) = f_{\mathsf{OT},\mathsf{S}}(\mathsf{t}_{\mathsf{S},i}, \mathsf{m}_{\mathsf{R}})$ and checks for all $i \in \{0, 1\}$ whether $s_{i,1} = s_{i,2}$. It outputs a random b' if it holds for both or no $i \in \{0, 1\}$, otherwise it outputs b' such that $s_{b',1} = s_{b',2}$. It holds that

$$\Pr[s_{b,1} = s_{b,2}] = \left(\frac{3}{4} + \alpha\right)^2 + \left(\frac{1}{4} - \alpha\right)^2 \frac{1}{2^\ell - 1} \geq \frac{9}{16}$$

and

$$\Pr[s_{\bar{b},1} = s_{\bar{b},2}] \geq \frac{1}{2^\ell} - \epsilon.$$

Therefore,

$$\begin{aligned}
&\Pr[b = b'] \\
&= \frac{1}{2}\Pr[\nexists! i : s_{i,1} = s_{i,2}] + \Pr[s_{b,1} = s_{b,2} \wedge s_{\bar{b},1} \neq s_{\bar{b},2}] - (1 - \delta) \\
&\geq \frac{2^\ell - 1}{2^{\ell+1}} - \frac{2^\ell - 2}{2^{\ell+1}}\Pr[s_{b,1} = s_{b,2}] + \frac{2^\ell - 1}{2^\ell}\Pr[s_{b,1} = s_{b,2}] - \mathsf{negl} \\
&\geq \frac{2^\ell - 1}{2^{\ell+1}} + \frac{2^\ell}{2^{\ell+1}}\frac{9}{16} - \mathsf{negl} \geq \frac{1}{2} + \frac{1}{32} - \frac{1}{2^{\ell+1}} + \frac{1}{4} - \mathsf{negl}
\end{aligned}$$

where we apply a union bound to argue that the outputs are correct and corresponds to the receivers choice. Given that $\ell \geq 1$, the malicious sender guesses b correctly with at least probability $\frac{1}{2} + \frac{1}{32} - \mathsf{negl}$. Since in the ideal model, an adversary can guess b only with probability $\frac{1}{2}$, this constitutes a break of receiver chosen message security.

In the case two, for any s_b,

$$\Pr_{\mathsf{m}_{\mathsf{S}},(\mathsf{t}_{\mathsf{R}},b,\mathsf{aux})}[f_{\mathsf{OT},\mathsf{R}}(\mathsf{t}_{\mathsf{R}}, \mathsf{m}_{\mathsf{S}}, b, \mathsf{aux}) = s_b \mid \mathsf{m}_{\mathsf{R}} = f_{\mathsf{R}}(\mathsf{t}_{\mathsf{R}}, b, \mathsf{aux})] < \frac{3}{4}.$$

Similar as in the proof of Lemma 3.8, we argue that a malicious sender can tweak the output distribution of (s_0, s_1). Due to the similarity, we only exhibit a brief version. Again we hardwire a PRF key k for a PRF with a single output bit. Given m_{R}, the malicious sender samples two random tapes $\mathsf{t}_{\mathsf{S},1}$ and $\mathsf{t}_{\mathsf{S},2}$, computes for all $i \in [2]$ $(s_{0,i}, s_{0,i}) = f_{\mathsf{OT},\mathsf{S}}(\mathsf{t}_{\mathsf{S},i}, \mathsf{m}_{\mathsf{R}})$. If $\mathsf{PRF}_{\mathsf{k}}(s_{0,1}, s_{1,1}) = 0$ output $(s_{0,1}, s_{1,1})$ and send $\mathsf{m}_{\mathsf{S},1} = f_{\mathsf{S}}(\mathsf{t}_{\mathsf{S},1}, \mathsf{m}_{\mathsf{R}})$ to R. Otherwise output $(s_{0,2}, s_{1,2})$ and send

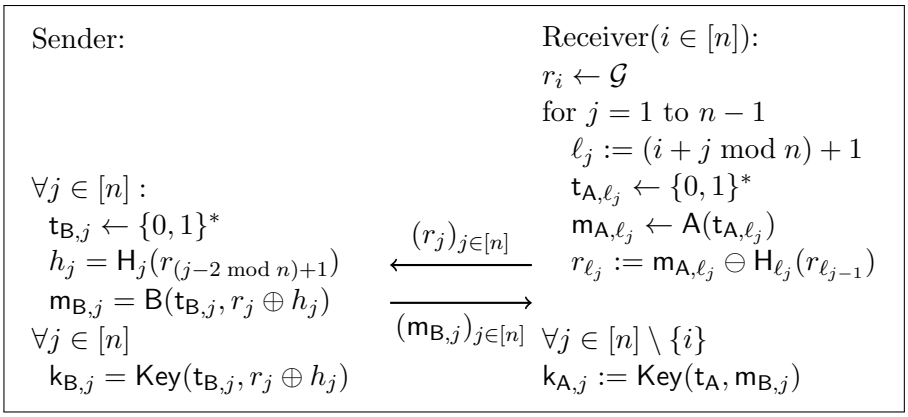

Figure 13: The figure shows a $n-1$ out of n OT using a $\mathsf{UKA} = (\mathsf{A}, \mathsf{B}, \mathsf{Key})$ and a random oracle $\mathsf{H} : \mathcal{G} \to \mathcal{G}$, where $\mathcal{G}$ is a group with operations $\oplus$, $\ominus$. By the correctness of the UKA scheme, $\mathsf{k}_{\mathsf{A},i} = \mathsf{k}_{\mathsf{B},i}$ holds. The scheme can be transformed in the same way in a one-round scheme given a one-round UKA as in the 1 out of n OT case in Section 4.

$\mathsf{m}_{\mathsf{S},2} = f_{\mathsf{S}}(\mathsf{t}_{\mathsf{S},2}, \mathsf{m}_{\mathsf{R}})$ to R. This way, $\Pr_{\mathsf{k}}(s_0, s_1) = 0$ holds for the malicious sender's output (s_0, s_1) with probability

$$\begin{aligned}
&\Pr[\mathsf{PRF}_{\mathsf{k}}(s_0, s_1) = 0] \\
&= \Pr[\mathsf{PRF}_{\mathsf{k}}(s_{0,1}, s_{1,1}) = 0] \\
&\quad + \Pr[\mathsf{PRF}_{\mathsf{k}}(s_{0,1}, s_{1,1}) = 1 \wedge \mathsf{PRF}_{\mathsf{k}}(s_{0,2}, s_{1,2}) = 0] \\
&\geq \left(\frac{1}{2} - \epsilon_{\mathsf{PRF}}\right) + \left(\frac{1}{2} + \epsilon_{\mathsf{PRF}}\right) \cdot \frac{3}{8}\left(\frac{1}{2} - \epsilon_{\mathsf{PRF}}\right) \\
&= \frac{1}{2} + \frac{1}{32} - \mathsf{negl},
\end{aligned}$$

unless one breaks the security of the PRF. As previously, this constitutes an attack against the receiver chosen message security. □

C All But One OT from Key Agreement

In this section we show how to use the techniques in Section 4 to construct an all but one, i.e. $n-1$ out of n, OT. We show the protocol in Figure 13 and give a state the achieved security in Lemma C.1 without giving a detailed proof. Security follows from the same reasoning as in Section 4.

Lemma C.1. *Given a correct and secure* UKA *scheme, then the* $n-1$ *out of* n *oblivious transfer in Figure 13 is an Endemic* $\mathsf{OT}_{n-1,n}$ *in the programmable random oracle model.*

Proof. The proof is very similar to the security proof of the 1 out of n OT. In fact, the proof is even simpler since the random oracle receives only a single r as input and for a malicious receiver, distinguishing a single string, i.e. k_i, needs to be hard. This even removes some of the complexity of the previous proof. In the following, we state the claims, which only require minor adaptations to the claims of the previous proofs. For this reason, we do not give their proofs here.

Security against a malicious sender follows by the claim below.

Claim C.2. *Given a δ correct and ϵ uniform* UKA *scheme, then it holds that in the programmable random oracle model for any ppt adversary $\mathcal{A}$, there exists a ppt adversary $\mathcal{A}$' such that for any ppt distinguisher* D *and any polynomial size auxiliary input z*

$$|\Pr[\mathsf{D}(z,(\mathcal{A},\mathsf{R})_\Pi)=1]-\Pr[\mathsf{D}(z,(\mathcal{A}',\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}))=1]|\leq\epsilon+(1-\delta),$$

where all algorithms receive input 1^κ and R *additionally receives input $\mathbb{S}$.*

By a second claim, the protocol is secure against a malicious receiver.

Claim C.3. *Given a δ correct, Q-multi-instance ϵ_u-uniform, $(Q,1)$-multi-instance ϵ_k-key-indistinguishable* UKA *scheme, where Q upper bounds the amount of random oracle queries by an adversary then it holds that in the programmable random oracle model for any ppt adversary $\mathcal{A}$, there exists a ppt adversary $\mathcal{A}$' such that for any ppt distinguisher* D *and any polynomial size auxiliary input z*

$$|\Pr[\mathsf{D}(z,(\mathsf{S},\mathcal{A})_\Pi)=1]-\Pr[\mathsf{D}(z,(\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}},\mathcal{A}'))=1]|\leq\epsilon_u+\epsilon_k+(1-\delta),$$

where all algorithms receive input 1^κ and adversary $\mathcal{A}$' rewinds $\mathcal{A}$ Q times.

□

D Instantiations

In the following, we first show how to efficiently instantiate the construction in Figure 8 using the Diffie-Hellman key exchange. In particular, we show how a tighter security reduction can be obtained using the random self-reducibility of the DDH assumption. In Figure 14, we give an optimized variant based on an interactive DDH assumption.

Afterwards, we show how to instantiate the construction in Figure 8 based on the lattice based Kyber key agreement.

We emphasize that the instantiations only achieve stand alone security. For UC security, one needs to assume that CODDH and Kyber are secure against non-uniform adversaries. For the proof of UC security against a malicious receiver, see Appendix E.

D.1 Instantiation from DDH

Definition D.1 (n-Multi-Instance DDH Assumption)**.** *For a group $\mathcal{G}$, the* decisional Diffie-Hellman *assumption is hard if for any ppt distinguisher* D*,*

$$|\Pr[\mathsf{D}([\![1]\!],[\![\vec{a}]\!],[\![b]\!],[\![\vec{a}b]\!])=1]-\Pr[D([\![1]\!],[\![\vec{a}]\!],[\![b]\!],[\![\vec{c}]\!])=1]|=\mathsf{negl},$$

where $\vec{a}\leftarrow\mathbb{Z}_p^n$, $b\leftarrow\mathbb{Z}_p$ and $\vec{c}\leftarrow\mathbb{Z}_p^n$.

By a standard hybrid argument, n-multi-instance DDH is secure under the DDH assumption with a security loss of n. In the following we show that the Diffie-Hellman key exchange is tightly multi-instance secure under multi-instance DDH.

Lemma D.2. *Let Q and n be polynomial in κ. The Diffie-Hellman key exchange over $\mathcal{G}$ is unconditionally Q-multi-instance uniform. Further, let the n-multi-instance DDH assumption hold over group $\mathcal{G}$ except with advantage ϵ, then the Diffie-Hellman key exchange is one-round (Q,n)-multi-instance key-indistinguishable except advantage $\epsilon-\mathsf{negl}$.*

Proof. The distribution of $[\![a]\!]$ over $\mathcal{G}$ is uniform, therefore

$$|\Pr[\mathsf{D}^{\mathcal{O}_\mathsf{A}}(1^\kappa)[\![]\!] = 1] - Pr[\mathsf{D}^{\mathcal{O}_u}(1^\kappa) = 1]| = 0,$$

even against an unbounded D. Hence, the Diffie-Hellman key exchange is unconditional Q-multi-instance uniform.

For proving the second part of the lemma, we construct a ppt distinguisher D that breaks n-multi-instance DDH assumption given a ppt distinguisher D_k that breaks the (Q,n)-multi-instance key-indistinguishability of the Diffie-Hellman key exchange. D receives a challenge $[\![\vec{a}]\!], [\![b]\!], [\![\vec{c}]\!]$, sets $\mathsf{m}_\mathsf{B} := [\![b]\!]$ and invokes D_k on input m_B. On the j-th query of D_k to $\mathcal{O}_\mathsf{A}$, D samples $\vec{r}_j \leftarrow \mathbb{Z}_p^{n+1}$ and responds with $\mathsf{m}_{\mathsf{A},j} := [\![\langle(\vec{a},1),\vec{r}_j\rangle]\!] = [\![a_1]\!]\cdot r_{j,1} + [\![a_2]\!]\cdot r_{j,2} \ldots [\![a_n]\!]\cdot r_{j,n} + [\![r_{j,n+1}]\!]$. When D_k queries $\mathcal{O}_\mathsf{k}$ for key k_j, D responds with $\mathsf{k}_j := [\![\langle\vec{c},\vec{r}_j\rangle]\!] = [\![c_1]\!]\cdot r_{j,1} + [\![c_2]\!]\cdot r_{j,2} \ldots [\![c_n]\!]\cdot r_{j,n} + [\![b]\!]\cdot r_{j,n+1}$. In the end, D outputs the output of D_k.

It is easy to see that $\mathcal{O}_\mathsf{A}$ has the correct output distribution. $r_{j,n+1}$ is uniform over $\mathbb{Z}_p$ and hence $\mathsf{m}_{\mathsf{A},j}$ is. Further, conditioned on $\mathsf{m}_{\mathsf{A},j}$, $r_{j,1},\ldots,r_{j,n}$ are uniform. Given that $\vec{c} = \vec{a}b$, the output

$$\mathsf{k}_j = [\![\langle\vec{c},\vec{r}_j\rangle]\!] + [\![b]\!]\cdot r_{j,n+1} = [\![\langle\vec{a}b,\vec{r}_j\rangle]\!] + [\![b]\!]\cdot r_{j,n+1} = [\![\langle(\vec{a},1),\vec{r}_j\rangle]\!]\cdot b = \mathsf{m}_{\mathsf{A},j}\cdot b$$

of $\mathcal{O}_\mathsf{k}$ is also distributed correctly. In case that $\vec{c}$ is uniform, we need to show that all the n outputs of $\mathcal{O}_\mathsf{k}$, $\vec{\mathsf{k}} = \mathsf{k}_1,\ldots,\mathsf{k}_n$ are uniform. Let m_i be the message $\mathsf{m}_{\mathsf{A},j}$ and $\vec{t}_i$ the randomness $\vec{r}_j$ that corresponds to k_i. Since $\vec{c}$ is uniform and $\vec{\mathsf{k}} = [\![\vec{c}\cdot T]\!]$, where T is the matrix with i-th column $\vec{t}_i$, $\vec{\mathsf{k}}$ is uniform if T is invertible. Since $r_{j,1},\ldots,r_{j,n}$ are uniform given $\mathsf{m}_{\mathsf{A},j}$ so is T. For a uniform T over $\mathbb{Z}_p^{n\times n}$, the probability that T is invertible is that all the rows are linear independent, i.e.

$$\Pr[T \text{ invertible}] = \frac{1}{p^{n^2}}\prod_{i=0}^{n-1}(p^n - p^i) \geq \left(1-\frac{1}{p}\right)^n \geq 1 - \mathsf{negl}.$$

Therefore, except with negligible probability, D has the same advantage in breaking n-multi-instance DDH as D_k has in breaking one-round (Q,n)-multi-instance key-indistinguishability. □

Using Theorem 4.1 and Lemma D.2, we obtain the following corollary.

Corollary D.3. *When instantiating an* 1 *out of* n *OT in Figure 8 with Diffie-Hellman key exchange over group* $\mathcal{G}$*, then in the programmable random oracle model the resulting endemic OT is statistically secure against malicious senders and secure against malicious receivers except advantage* $(n-1)\epsilon_{\mathsf{DDH}} + \mathsf{negl}$ *and a runing time loss* Q*, where the DDH assumption over group* $\mathcal{G}$ *holds except advantage* ϵ_{DDH} *and* Q *is a bound on the amount of adversarial random oracle queries.*

Remark D.4. *If we apply a hardcore predicate to* k_A *and* k_B *in the Diffie-Hellman key agreement and apply the transformation Figure 8, we receive a one-round endemically secure OT in the random oracle model based on the computational Diffie-Hellman assumption. Alternatively, one could also use the random oracle instead of a hardcore predicate to obtain longer OT strings.*

D.2 Optimized, Interactive DDH based Instantiation

In Figure 14, we show an optimized variant of our OT. It reduces the communication cost from the sender to the receiver by sending only a single group element. This is possible since it does not depend on any of the n elements sent by the receiver. A drawback of this construction is that we do not know how to prove its security under the standard DDH assumption.

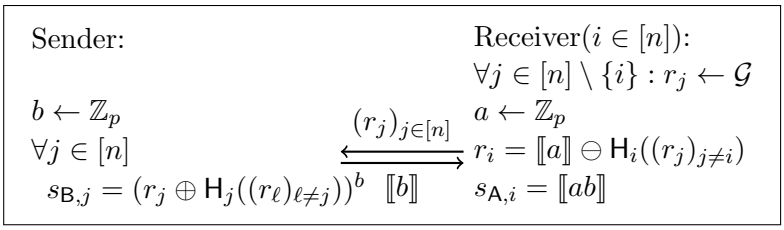

Figure 14: The figure shows an optimized variant of the protocol from Figure 8 based on an interactive DDH assumption.

The reason is simple, in our security proof during the simulation, $\mathcal{A}'$ needs compute the key $[\![ab]\!]$ while at the same time given $[\![1]\!], [\![a']\!], [\![b]\!], [\![ab']\!]$ needs to be hard to distinguish from a uniformly random group element. Since a is picked by the adversary $\mathcal{A}$ and only transmits $[\![a]\!]$, we do not know how to simulate correctly. The next definition formally states the assumption under which the protocol in Figure 14 can be proven to be secure.

Definition D.5 (Interactive Decisional Diffie-Hellman (IDDH) Assumption)**.** *For a group $\mathcal{G}$, the* interactive decisional Diffie-Hellman *assumption is hard if for any ppt distinguisher $\mathsf{D}_1, \mathsf{D}_2$,*

$$\begin{aligned}&|\Pr[\mathsf{D}_2(\mathsf{st}, [\![1]\!], [\![a]\!], [\![xb]\!], [\![ab]\!]) = 1]\\ &-\Pr[\mathsf{D}_2(\mathsf{st}, [\![1]\!], [\![a]\!], [\![xb]\!], [\![c]\!]) = 1]| = \mathsf{negl},\end{aligned}$$

where $a \leftarrow \mathbb{Z}_p$, $b \leftarrow \mathbb{Z}_p$ and $c \leftarrow \mathbb{Z}_p$ and $(\mathsf{st}, [\![x]\!]) \leftarrow \mathsf{D}_1([\![b]\!])$.

D.3 Instantiation based on Crystals-Kyber

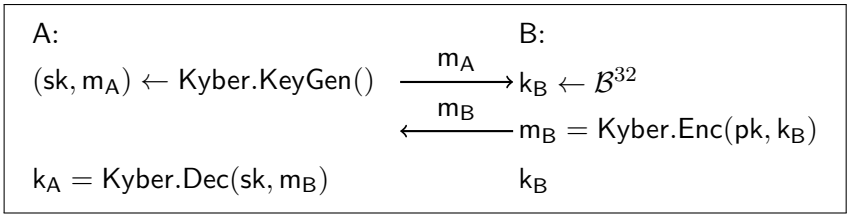

Figure 15: The figure shows a Kyber based key agreement protocol between parties A and B.

Definition D.6 (Crystals-Kyber CPAPKE)**.** Crystals-Kyber CPAPKE *is a correct and CPA secure public key encryption based on the Module LWE (*MLWE*) assumption. We follow the specifications of Kyber, in which $\mathcal{B}$ denotes the set $\{0, 1, \ldots, 255\}$. Kyber is parameterized by parameters $d_t = 11$, $n_{\mathsf{LWE}} = 256$, $k_{\mathsf{LWE}} \in \{2, 3, 4\}$, $q = 7681$ and ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^{n_{\mathsf{LWE}}} + 1)$.*

*(*Kyber.KeyGen, Kyber.Enc, Kyber.Dec*) have the following syntax.*

Kyber.KeyGen**:** *Outputs a secret and public key pair* pk, sk*, where* $\mathsf{pk} = (\tilde{\mathbf{t}}, \rho) \in \mathcal{B}^{\frac{k_{\mathsf{LWE}} n_{\mathsf{LWE}} d_t}{8}} \times \mathcal{B}^{32}$.

Kyber.Enc**:** *Takes as input a public key* pk*, a message* $\mathsf{m} \in \mathcal{B}^{32}$ *and random coins* $\mathsf{t} \in \mathcal{B}^{32}$*. It outputs a ciphertext* c.

Kyber.Dec**:** *Takes as input secret key* sk *and a ciphertext* c*. It outputs a message* m.

Further, Crystals-Kyber specifies the following algorithms.

Decode_ℓ: *Takes a an element in* $\mathcal{B}^{32\ell}$ *and maps it to* $\mathcal{R}_q$. *The inverse operation is* Encode_ℓ.

$\mathsf{Compress}_q(*, d)$: *Takes a an element in* $\mathcal{R}_q$ *and maps it to a polynomial with coefficients in* $\mathbb{Z}_{2^d}$. *The inverse is* $\mathsf{Decompress}_q(*, d)$.

Parse: *Takes a uniform byte stream in* $\mathcal{B}^*$ *and maps it to a uniform element in* $\mathcal{R}_q$.

It is important to know, that $\tilde{\mathbf{t}} := \mathsf{Encode}_{d_t}(\mathsf{Compress}_q(\mathbf{t}, d_t)$, where $\mathbf{t}$ is computationally indistinguishable from a uniform element in $\mathcal{R}_q^{k_{\mathsf{LWE}}}$ based on the MLWE assumption. In the following, we will choose $d_t = 13$ such that $q < 2^{d_t}$ and no compression takes place. This will help us to avoid complications and does not decrease efficiency besides a slightly larger public key pk. Further, we will keep component ρ of pk consistent between all pk used in a single 1 out of n OT.

In order to instantiate our framework, we need to define a group operation $\mathsf{pk} \oplus \mathsf{pk}$ and a hash functions that maps to a uniform $\tilde{\mathbf{t}}$ component of pk, i.e. a uniform element in $\mathcal{R}_q^{k_{\mathsf{LWE}}}$. For the latter, we use a hash function that produces an output bit stream, which is in $\mathcal{B}^*$, and use k_{LWE} different parts of the stream and apply Parse to the k_{LWE} streams to obtain a (pseudo) uniform element in $\mathcal{R}_q^{k_{\mathsf{LWE}}}$ whenever the bit streams are (pseudo) uniform.

We define the group operation $\mathsf{pk}_1 \oplus \mathsf{pk}_2$, by mapping $\mathsf{pk}_1 = (\tilde{\mathbf{t}}_1, \rho)$, $\mathsf{pk}_2 = (\tilde{\mathbf{t}}_2, \rho)$ to $\mathsf{pk}_3 = (\tilde{\mathbf{t}}_3, \rho)$, where $\tilde{\mathbf{t}}_3 = \mathsf{Encode}_{13}(\mathsf{Decode}_{13}(\tilde{\mathbf{t}}_1) + \mathsf{Decode}_{13}(\tilde{\mathbf{t}}_2))$ and $+$ is the addition in $\mathcal{R}_q^{k_{\mathsf{LWE}}}$. $\ominus$ is defined correspondingly.

By using Theorem 4.1, Lemma A.7 and Lemma A.8, we get the following corollary.

Corollary D.7. *When instantiating an* 1 *out of* n *OT in Figure D.3 with Crystals-Kyber, then in the programmable random oracle model the resulting endemic OT is secure against a malicious sender except advantage* $(n-1)\epsilon_{\mathsf{MLWE}} + \mathsf{negl}$ *and secure against malicious receivers except advantage* $Q\epsilon_{\mathsf{MLWE}} + \mathsf{negl}$ *and a runing time loss* Q, *where the* MLWE *assumption holds except advantage* ϵ_{MLWE} *and* Q *is a bound on the amount of adversarial random oracle queries.*

In this work, we will instantiate Kyber with $k = 3$ which is claimed to have a qbit security level of 161 bit. This security level does not immediately carry over to our Kyber based OT, there is an additional security loss of Q^2. Though we are unaware of an attack that is significantly more efficient on our Kyber based OT than the attacks on Kyber.

E UC Security against Malicious Receivers

In Theorem 4.1 in Section 4, we only show stand-alone security. While UC security for a malicious sender is already covered by Claim C.2, Claim C.3 uses an adversary $\mathcal{A}$' that rewinds $\mathcal{A}$ and hence does not accomplish UC security. Here, we proof UC security against a malicious receiver for two settings. First, in case of a two-round OT from any uniform two-round key agreement. Second, in case of a one-round OT based on the Diffie Hellman key agreement under a variant of DDH.

E.1 UC Security of the Two-Round OT

Claim E.1. *Given a* δ *correct,* Q*-multi-instance* ϵ_u*-uniform,* $(Q, n-1)$*-multi-instance* ϵ_k*-key indistinguishable two-round* UKA *scheme, where* Q *upper bounds the amount of random oracle queries by an adversary. Then the proposed two-round OT is UC-secure against malicious receivers, i.e. in the programmable random oracle model for any ppt adversary* $\mathcal{A}$, *there exists a ppt adversary* $\mathcal{A}'$ *such that for any ppt distinguisher* D *and any polynomial size auxiliary input* z,

$$|\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_\Pi) = 1] - \Pr[\mathsf{D}(z, (\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{A}')) = 1]| \leq 2Q\epsilon_u + Q\epsilon_k + (1 - \delta),$$

where all algorithms receive input 1^κ.

Proof. We follow the same line of arguement as in Claim C.3 with the exception that we now use a hybrid argument rather than guessing the correct random oracle query.

A malicious receiver will make random oracle queries and each query will correspond to a potential choice for $(r_i)_{i\in[n]}$. For each potential choice of $(r_i)_{i\in[n]}$ there will be corresponding OT strings, i.e. keys computed by the key agreement.

During the first hybrid, we replace the corresponding keys of the first random oracle query with uniform. Through a sequence of hybrids, we will do this for every query until all the keys are replaced with uniform. Notice that in each hybrid, D will only get to see the key that corresponds to the malicious receivers choice of $(r_i)_{i\in[n]}$ and not for all potential choices of $(r_i)_{i\in[n]}$.

We start by giving a description of $\mathcal{A}$'. For each random oracle query q to H_i for an $i \in [n]$, $\mathcal{A}$' responds with a random group element $\mathsf{H}_i(q) \leftarrow \mathcal{G}$. When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, $\mathcal{A}$' looks up the first oracle query of the form $q = (r_1, \ldots, r_{i^*-1}, r_{i^*+1}, \ldots, r_n)$ for an $i^* \in [n]$. $\mathcal{A}$' sends i^* to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. $\mathcal{A}$' computes for all $i \in [n]$ $\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i})$, $\mathsf{t}_{\mathsf{B},i} \leftarrow \{0,1\}^*$ and $\mathsf{m}_{\mathsf{B},i} \leftarrow \mathsf{B}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. It also computes $s_{\mathsf{B},i^*} := \mathsf{Key}(\mathsf{t}_{\mathsf{B},i^*}, \mathsf{m}_{\mathsf{A},i^*})$. $\mathcal{A}$' sends s_{B,i^*} to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$, $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ and outputs the output of $\mathcal{A}$. This concludes the description of $\mathcal{A}$'. We emphasize that here, the other OT strings $(s_{\mathsf{B},i})_{i\neq i^*}$ will not be the output of $\mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$ but sampled uniformly by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$.

We now define a sequence of hybrids. The first hybrid is hyb_1 and corresponds the interaction of $\mathcal{A}$ with the sender of the protocol description. The last hybrid, hyb_{3Q+1}, corresponds to simulator $\mathcal{A}'$. Let us define the *critical query* with index $j^* \in [Q]$ as the first query of the form $H_d(r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ where $\mathcal{A}$ sends $(r_i)_{i\in[n]}$. For $k \in [Q+1]$, we define:

hyb_{3k-2}: In this hybrid the simulator $\mathcal{A}'$ does not program the random oracle and outputs uniform OT messages as the ideal functionality would if $j^* < k$. In more detail, $\mathcal{A}'$ does the following:

When $\mathcal{A}$ makes an oracle query q_j respond normally with a random group element $\mathsf{H}_i(q_j) \leftarrow \mathcal{G}$. When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, look up the the critical query of the form $q_{j^*} = (r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ to H_d for a $d \in [n]$. Let j^* be the query index. Compute for all $i \in [n]$, $\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i})$, $\mathsf{t}_{\mathsf{B},i} \leftarrow \{0,1\}^*$ and $\mathsf{m}_{\mathsf{B},i} \leftarrow \mathsf{B}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. Further, compute $s_{\mathsf{B},i} := \mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$.

If $j^* < k$, sample for all $i \neq d$, s_i uniformly. Otherwise, for all $i \neq d$, $s_i := s_{\mathsf{B},i}$. Define $\mathbb{S}_{\mathsf{B}} := (s_1, \ldots, s_{d-1}, s_{\mathsf{B},d}, s_{d+1}, \ldots, s_n)$. Send $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ and output $\mathbb{S}_{\mathsf{B}}$ together with the output of $\mathcal{A}$.

hyb_{3k-1}: In this hybrid $\mathcal{A}'$ programs the oracle to prepare a switch to uniform keys when $j^* = k$. In particular, the hybrid is:

When $\mathcal{A}$ makes an oracle query q_j respond normally with a random group element $\mathsf{H}_i(q_j) \leftarrow \mathcal{G}$ except for the following queries. Let us define $i^*, (g_1^*, \ldots, g_{i^*-1}^*, g_{i^*+1}^*, \ldots, g_n^*) := q_k$ s.t. the k'th oracle query $\mathcal{A}$ makes is $\mathsf{H}_{i^*}(q_k)$. For all following random oracle queries $H_i(q_j)$ and $i \neq i^*$ s.t. $q_j \in \{(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^* \mid g \in G\}$, sample random tape $\mathsf{t}_j \leftarrow \{0,1\}^*$ and compute $\mathsf{m}_j \leftarrow \mathsf{A}(\mathsf{t}_j)$. Respond abnormally with $\mathsf{H}_i(q_j) := \mathsf{m}_j \ominus g_i^*$. Here we define $(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^*$ as the ordered sequence with the element g_i^* removed.

When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, look up the the critical query of the form $q_{j^*} = (r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ to H_d for a $d \in [n]$. Let j^* be the query index. Compute for all $i \in [n]$, $\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i})$, $\mathsf{t}_{\mathsf{B},i} \leftarrow \{0,1\}^*$ and $\mathsf{m}_{\mathsf{B},i} \leftarrow \mathsf{B}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. Further, compute $s_{\mathsf{B},i} := \mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$.

If $j^* < k$, sample for all $i \neq d$, s_i uniformly. Otherwise, for all $i \neq d$, $s_i := s_{\mathsf{B},i}$. Define $\mathbb{S}_{\mathsf{B}} := (s_1, \ldots, s_{d-1}, s_{\mathsf{B},d}, s_{d+1}, \ldots, s_n)$. Send $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ and output $\mathbb{S}_{\mathsf{B}}$ together with the output of $\mathcal{A}$.

hyb_{3k}**:** In this hybrid $\mathcal{A}'$ replaces the true key exchange keys for query k with the uniform challenges. This change is only observable if $j^* = k$. In particular, the hybrid is:

When $\mathcal{A}$ makes an oracle query q_j respond normally with a random group element $\mathsf{H}_i(q_j) \leftarrow \mathcal{G}$ except for the following queries. Let us define $i^*, (g_1^*, \ldots, g_{i^*-1}^*, g_{i^*+1}^*, \ldots, g_n^*) := q_k$ s.t. the k'th oracle query $\mathcal{A}$ makes is $\mathsf{H}_{i^*}(q_k)$. For all following random oracle queries $H_i(q_j)$ and $i \neq i^*$ s.t. $q_j \in \{(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^* \mid g \in G\}$, sample random tape $\mathsf{t}_j \leftarrow \{0,1\}^*$ and compute $\mathsf{m}_j \leftarrow \mathsf{A}(\mathsf{t}_j)$. Respond abnormally with $\mathsf{H}_i(q_j) := \mathsf{m}_j \ominus g_i^*$. Here we define $(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^*$ as the ordered sequence with the element g_i^* removed.

When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, look up the the critical query of the form $q_{j^*} = (r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ to H_d for a $d \in [n]$. Let j^* be the query index. Compute for all $i \in [n]$, $\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i})$, $\mathsf{t}_{\mathsf{B},i} \leftarrow \{0,1\}^*$ and $\mathsf{m}_{\mathsf{B},i} \leftarrow \mathsf{B}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. Further, compute $s_{\mathsf{B},i} := \mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$.

If $j^* \leq k$, sample for all $i \neq d$, s_i uniformly. Otherwise, for all $i \neq d$, $s_i := s_{\mathsf{B},i}$. Define $\mathbb{S}_\mathsf{B} := (s_1, \ldots, s_{d-1}, s_{\mathsf{B},d}, s_{d+1}, \ldots, s_n)$. Send $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ and output $\mathbb{S}_\mathsf{B}$ together with the output of $\mathcal{A}$.

Claim E.2. *For any $k \in [Q+1]$, let there be a distinguisher D and a polynomial size auxiliary input z with*

$$\epsilon_\mathsf{D} := |\Pr[\mathsf{D}(z, \mathsf{hyb}_{3k-2}) = 1] - \Pr[\mathsf{D}(z, \mathsf{hyb}_{3k-1}) = 1]|.$$

Then, there is a distinguisher D_u breaking the Q-multi-instance uniformity of the UKA protocol.

Proof. D_u gets access to an oracle $\mathcal{O}$ which either outputs uniform messages, i.e. $\mathcal{O}_u$ or messages of the form $\mathsf{m}_\mathsf{A} \leftarrow \mathsf{A}(\mathsf{t}_\mathsf{A})$ for $\mathsf{t}_\mathsf{A} \leftarrow \{0,1\}^*$. D_u invokes D and creates its input as follows. It invokes $\mathcal{A}$ and interacts with him as hyb_{3k-1} does with the difference that m_j are requested from $\mathcal{O}$ rather than computing them. After receiving the output, D_u uses it as input for D together with $(s_{\mathsf{B},i})_{i\in[n]}$, where $s_{\mathsf{B},i} \leftarrow \mathsf{Key}(\mathsf{t}_{\mathsf{B},i}, \mathsf{m}_{\mathsf{A},i})$. D_u outputs the output of D.

If $\mathcal{O}$ is oracle $\mathcal{O}_u$, all m_j are uniform and hence all random oracle queries q are answered with a uniformly random $\mathsf{H}_i(q) \in \mathcal{G}$. Otherwise, $\mathcal{A}$' is identical with S as well as $(s_{\mathsf{B},i})_{i\in[n]}$ are identical with the output of S. Hence

$$\begin{aligned}
\epsilon_u &= |\Pr[\mathsf{D}_u^{\mathcal{O}_\mathsf{A}}(z)] = 1] - Pr[\mathsf{D}_u^{\mathcal{O}_u}(z) = 1]| \\
&= |\Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_u^{\mathcal{O}_\mathsf{A}}}) = 1] \\
&\quad - \Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_u^{\mathcal{O}_u}}) = 1]| \\
&\geq \epsilon_\mathsf{D}.
\end{aligned}$$

□

Claim E.3. *For any $k \in [Q+1]$, let there be a distinguisher D and a polynomial size auxiliary input z with*

$$\epsilon_\mathsf{D} := |\Pr[\mathsf{D}(z, \mathsf{hyb}_{3k-1}) = 1] - \Pr[\mathsf{D}(z, \mathsf{hyb}_{3k}) = 1]|.$$

Then there is a distinguisher D_k that breaks the $(Q, n-1)$-multi-instance key-indistinguishability of the UKA protocol.

Proof. D_k has access to oracles $\mathcal{O}_{\langle\mathsf{A},\mathsf{B}\rangle}$ and $\mathcal{O}$ which is either $\mathcal{O}_u$ or $\mathcal{O}_k$. D_k invokes D and creates its input as follows. D_k invokes $\mathcal{A}$ and interacts with it as hyb_{3k+2} does with the difference, that D_k generates m_j by querying a transcript $\langle\mathsf{A},\mathsf{B}\rangle = (\mathsf{m}'_{\mathsf{A},j}, \mathsf{m}'_{\mathsf{B},j})$ from $\mathcal{O}_{\langle\mathsf{A},\mathsf{B}\rangle}$ and setting $\mathsf{m}_j = \mathsf{m}'_{\mathsf{A},j}$.

If $(r_i)_{i\in[n]}$ corresponds to a query $j \neq k$, then hyb_{3k-1} and hyb_{3k} are equivalent. Follow the description of hyb_{3k-1} and ignore oracle $\mathcal{O}$, since the keys for the challenge transcripts are not needed.

If $(r_i)_{i\in[n]}$ corresponds to query k, compute for all $i \in [n] \setminus \{i^*\}$

$$\mathsf{m}_{\mathsf{A},i} := r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i}) = \mathsf{m}'_{\mathsf{A},j}$$

where there exists a $j \in [Q]$ such that the last equality holds. It also uses oracle $\mathcal{O}$ to query for all $i \in [n] \setminus \{i^*\}$ the $n-1$ corresponding keys k_i that match with the transcripts containing $\mathsf{m}_{\mathsf{A},i}$. D_k sets $\mathsf{m}_{\mathsf{B},i} := \mathsf{m}'_{\mathsf{B},j}$ and $s_{\mathsf{B},i} := \mathsf{k}_i$. It creates $\mathsf{m}_{\mathsf{B},i^*}$ and s_{B,i^*} as usual. It sends $(\mathsf{m}_{\mathsf{B},i})_{i\in[n]}$ to $\mathcal{A}$ to receive its output which it uses together with $(s_{\mathsf{B},i})_{i\in[n]}$ as input for D. D_k outputs D's output.

$$\begin{aligned}
\epsilon_k &= |\Pr[\mathsf{D}_k^{\mathcal{O}_k}(z)] = 1] - Pr[\mathsf{D}_k^{\mathcal{O}_u}(z) = 1]| \\
&= |\Pr[\mathsf{D}(z, ((s_{\mathsf{B},i})_{i\in[n]}, \mathcal{A})_{\mathsf{D}_k^{\mathcal{O}_k}}) = 1] \\
&\quad -Pr[\mathsf{D}(z, (\mathbb{S}_{\mathsf{B},i^*}, \mathcal{A})_{\mathsf{D}_k^{\mathcal{O}_u}}) = 1]| \\
&\geq \epsilon_{\mathsf{D}}.
\end{aligned}$$

□

Claim E.4. *For any $k \in [Q]$, let there be a distinguisher D and a polynomial size auxiliary input z with*

$$\epsilon_{\mathsf{D}} := |\Pr[\mathsf{D}(z, \mathsf{hyb}_{3k}) = 1] - \Pr[\mathsf{D}(z, \mathsf{hyb}_{3k+1}) = 1]|.$$

Then, there is a distinguisher D_u breaking the Q-multi-instance uniformity of the UKA protocol.

Proof. The proof is almost identical to the proof of Claim E.4 and therefore omitted. □

For the last step, we need to replace s_{B,i^*} with s_{A,i^*}. We use the same argument as in Claim C.2 using the correctness of the scheme. Hence we obtain

$$\begin{aligned}
\epsilon_{\mathsf{OT}} &= |\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_\Pi) = 1] - \Pr[\mathsf{D}(z, (\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{A})) = 1]| \\
&\leq 2Q\epsilon_u + Q\epsilon_k + (1-\delta).
\end{aligned}$$

□

Remark E.5. *For stand alone security, security of UKA against uniform adversaries is sufficient, i.e. auxiliary input z is the empty string. Further, for UC security in the global random oracle model, it is sufficient for the sender to send a salt at the start of each session that is used as an additional input to the random oracle within the session.*

E.2 Diffie-Hellman based One-Round Endemic OT with UC Security

Definition E.6 (Choose-and-Open Decisional Diffie-Hellman (CODDH) Assumption)**.** *For a group $\mathcal{G}$, the* choose-and-open decisional Diffie-Hellman *assumption for parameters k, m is hard if for any ppt distinguisher $\mathsf{D}_1, \mathsf{D}_2$ and any polynomial size auxiliary input z,*

$$\begin{aligned}
&|\Pr[\mathsf{D}_2(\mathsf{st}, (a_j)_{j\in K}, ([\![a_j b_j]\!])_{j\notin K}) = 1] \\
&-\Pr[\mathsf{D}_2(\mathsf{st}, (a_j)_{j\in K}, ([\![c_j]\!])_{j\notin K}) = 1]| = \mathsf{negl},
\end{aligned}$$

where for $j \in [m]$, $a_j, b_j, c_j \leftarrow \mathbb{Z}_p$ and $(\mathsf{st}, K) \leftarrow \mathsf{D}_1(z, [\![1]\!], ([\![a_j]\!], [\![b_j]\!])_{j\in[m]})$ with $K \subset [m]$, $|K| = k$.

Lemma E.7. *Let the Choose-and-Open DDH assumption for parameters $k = n$, $m = 2n$ (Definition E.6) hold over group $\mathcal{G}$. Then the one-round Diffie-Hellman based protocol on Figure 8 satisfies malicious receiver security (Definition 2.6) in the UC model with respect to the 1-out-of-n $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ functionality.*

Proof. The difference to the previous regimes is that now the simulator $\mathcal{A}'$ and sender will send their message before seeing the adversaries first message. The simulator for the malicious receiver is still straight forward. It sends the first message according to protocol. As previously, he will extract the receiver's input after seeing the malicious receveirs response $(r_i)_{i\in[n]}$. The input will be the index i^* of random oracle H_{i^*} for which the malicious receiver makes the first query of the form $(g_1, \ldots, g_{i^*-1}, g_{i^*+1}, \ldots, g_n)$. The simulator sends the choice bit and all keys to the ideal functionality, where only the i^*th key will be computed according to the protocol, all other keys are uniformly random.

As in the previous regime, we define a sequence of hybrids. We now define a sequence of hybrids. The first hybrid is hyb_1 and corresponds the interaction of $\mathcal{A}$ with the sender of the protocol description. The last hybrid, hyb_{Q+1}, corresponds to simulator $\mathcal{A}'$. Since the messages in the Diffie-Hellman key agreement are statistically close to uniform, we need less hybrids. Let us define the *critical query* with index $j^* \in [Q]$ as the first query of the form $H_d(r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ where $\mathcal{A}$ sends $(r_i)_{i\in[n]}$. For $k \in [Q+1]$, we define:

hyb_k: In this hybrid the simulator $\mathcal{A}'$ outputs uniform OT messages as the ideal functionality would if $j^* < k$. In more detail, $\mathcal{A}'$ does the following:

$\mathcal{A}'$ sends $(\llbracket a_j \rrbracket)_{j\in[n]}$ to the malicious receiver as its OT message. When $\mathcal{A}$ makes an oracle query q_j, respond with a random group element $\mathsf{H}_i(q_j) \leftarrow \mathcal{G}$. When $\mathcal{A}$ sends $(r_i)_{i\in[n]}$, look up the the critical query of the form $q_{j^*} = (r_1, \ldots, r_{d-1}, r_{d+1}, \ldots, r_n)$ to H_d for a $d \in [n]$. Let j^* be the query index. Compute for all $i \in [n]$, $s_{\mathsf{B},i} := a_i \cdot (r_i \oplus \mathsf{H}_i((r_\ell)_{\ell\neq i}))$.

If $j^* < k$, sample for all $i \neq d$, s_i uniformly. Otherwise, for all $i \neq d$, $s_i := s_{\mathsf{B},i}$. Define $\mathbb{S}_{\mathsf{B}} := (s_1, \ldots, s_{d-1}, s_{\mathsf{B},d}, s_{d+1}, \ldots, s_n)$. Output $\mathbb{S}_{\mathsf{B}}$ together with the output of $\mathcal{A}$.

Claim E.8. *For any $k \in [Q+1]$, let there be a distinguisher D and a polynomial size auxiliary input z with*

$$\epsilon_{\mathsf{D}} := |\Pr[\mathsf{D}(z, \mathsf{hyb}_k) = 1] - \Pr[\mathsf{D}(z, \mathsf{hyb}_{k+1}) = 1]|.$$

Then there is a distinguisher D' that breaks CODDH for parameter $k = n$, $m = 2n$.

Proof. First, D' receives challenge $\llbracket 1 \rrbracket, (\llbracket a_i \rrbracket, \llbracket b_i \rrbracket)_{i\in[m]}$. He sends $(\llbracket a_i \rrbracket)_{i\in[n]}$ to the malicious receiver as its OT message ($(\llbracket a_i \rrbracket, \llbracket b_i \rrbracket)_{i\in[m]\setminus[n]}$ are ignored). He programs the random oracle similar as in the proof of Claim E.1. I.e. when $\mathcal{A}$ makes an oracle query q_j respond normally with a random group element $\mathsf{H}_i(q_j) \leftarrow \mathcal{G}$ except for the following queries. Let us define $i^*, (g_1^*, \ldots, g_{i^*-1}^*, g_{i^*+1}^*, \ldots, g_n^*) := q_k$ s.t. the k'th oracle query $\mathcal{A}$ makes is $\mathsf{H}_{i^*}(q_k)$. For all following random oracle queries $H_i(q_j)$ and $i \neq i^*$ s.t. $q_j \in \{(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^* \mid g \in G\}$, sample $\beta_j \leftarrow \mathbb{Z}_p$ and respond with $\mathsf{H}_i(q_j) := \llbracket b_i \rrbracket \cdot \beta_j - g_i^*$. Here we define $(g_1^*, \ldots, g_{i^*-1}^*, g, g_{i^*+1}^*, \ldots, g_n^*) \setminus g_i^*$ as the ordered sequence with the element g_i^* removed.

After A sends $(r_i)_{i\in[n]}$, D' checks whether it corresponds to query k. If not, D' requests $(a_i)_{i\in[n]}$ and continues as the honest server. If $(r_i)_{i\in[n]}$ corresponds to oracle query k, D' requests a_{i^*}, challenges $\llbracket c_i \rrbracket_{i\in[n]\setminus\{i^*\}}$ and computes s_{B,i^*} according to protocol. For all $i \neq i^*$, $s_i := \llbracket c_i \rrbracket \cdot \delta_j$, where δ_j was sampled when query $(r_\ell)_{\ell\neq i}$ was made to the random oracle, i.e. the jth query for some $j \in [Q]$. D' outputs $\mathbb{S}_{\mathsf{B}} := (s_1, \ldots, s_{d-1}, s_{\mathsf{B},d}, s_{d+1}, \ldots, s_n)$ and the output of A to D.

Clearly, since for all $j \in [Q]$, δ_j is uniform, $H_i(q_j)$ for the corresponding $i \in [n]$ is uniform as well. When c_i is uniform, so will be s_i and thus it is distributed as the ideal functionalities output. When $c_i = a_i b_i$,

$$s_i = [\![c_i \delta_j]\!] = a_i \cdot [\![b_i \delta_j]\!] = a_i \cdot (r_i + \mathsf{H}((r_j)_{j \neq i}))$$

and thus distributed as the OT strings computed by the honest sender. □

For the last step, we need to replace s_{B,i^*} with s_{A,i^*}. We use the same argument as in Claim C.2 using the correctness of the scheme. Hence we obtain

$$\begin{aligned} \epsilon_{\mathsf{OT}} &= |\Pr[\mathsf{D}(z, (\mathsf{S}, \mathcal{A})_{\Pi}) = 1] - \Pr[\mathsf{D}(z, (\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}, \mathcal{A})) = 1]| \\ &\leq Q\epsilon + (1 - \delta), \end{aligned}$$

where ϵ_k is the advantage for breaking CODDH for parameters k, m. □

F OT Extension

F.1 Protocol Diagrams

F.2 Proof of Lemma F.1 (Attack of Π^{OOS})

Lemma F.1. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{OOS}}) = 1] - \Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1]| = 1 - 2^{-\kappa}$$

where Π^{OOS} is the protocol in Definition 5.1. All algorithms also receive input 1^κ.

Proof. For simplicity let $N = 2$ and $m = 1$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of R and replaces the input to base OTs, the sender input, with strings $\boldsymbol{t}_0^j, \boldsymbol{t}_1^j \in \{0\}^{m'}$ and then completes the protocol as normal.

We define D as follows. D executes S and $\mathcal{A}$ with input $x_1 = 1$. S outputs $(\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2})$ and D outputs 1 if $\boldsymbol{v}_{1,1} = \mathsf{H}(1, \{0\}^{n_C})$ and 0 otherwise. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{\mathsf{OOS}}}) = 1] = 1$. In the ideal interaction the honest S will output a uniformly distributed $\boldsymbol{v}_{1,1} \in \{0,1\}^\kappa$ which was sampled by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ and therefore $\Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1] = 2^{-\kappa}$. □

F.3 Proof of Lemma F.2 (Attack of $\Pi^{\mathsf{OOS+}}$)

Lemma F.2. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{OOS+}}) = 1] - \Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1]| = 1 - 2^{-\kappa}$$

where Π^{OOS+} is the protocol in Definition 5.4 and all algorithms additionally receive input 1^κ.

Proof. For simplicity let $N = 2$ and $m = \kappa$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of R and receives the strings $\boldsymbol{t}_0^j, \boldsymbol{t}_1^j \in \{0\}^{m'}$ from $\mathcal{F}_{\mathsf{OT}}$. $\mathcal{A}$ redefines the selection values $x_1, ..., x_m \in [2]$ of R such that $x_i := \mathrm{LSB}(\mathsf{H}(i, \boldsymbol{t}_i)) + 1$. That is, x_i equals the least significant bit of $\boldsymbol{v}_{i,x_i} = \mathsf{H}(i, \boldsymbol{t}_i)$ plus 1. $\mathcal{A}$ executes the rest of the protocol as R would and outputs $(x_i)_{i \in [m]}$.

We define D as follows. D executes S and $\mathcal{A}$. S outputs $(\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2})_{i \in [m]}$ and D outputs 1 if $\forall i \in [m], \mathrm{LSB}(\boldsymbol{v}_{i,x_i}) + 1 = x_i$ and 0 otherwise. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathsf{S}, \mathcal{A})_{\Pi^{\mathsf{OOS+}}}) = 1] = 1$. In the ideal interaction the honest S will output a uniformly distributed $\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2} \in \{0,1\}^\kappa$ which are independent of x_i and therefore $\Pr[\mathsf{D}((\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}, \mathcal{A}')) = 1] = 2^{-\kappa}$. □

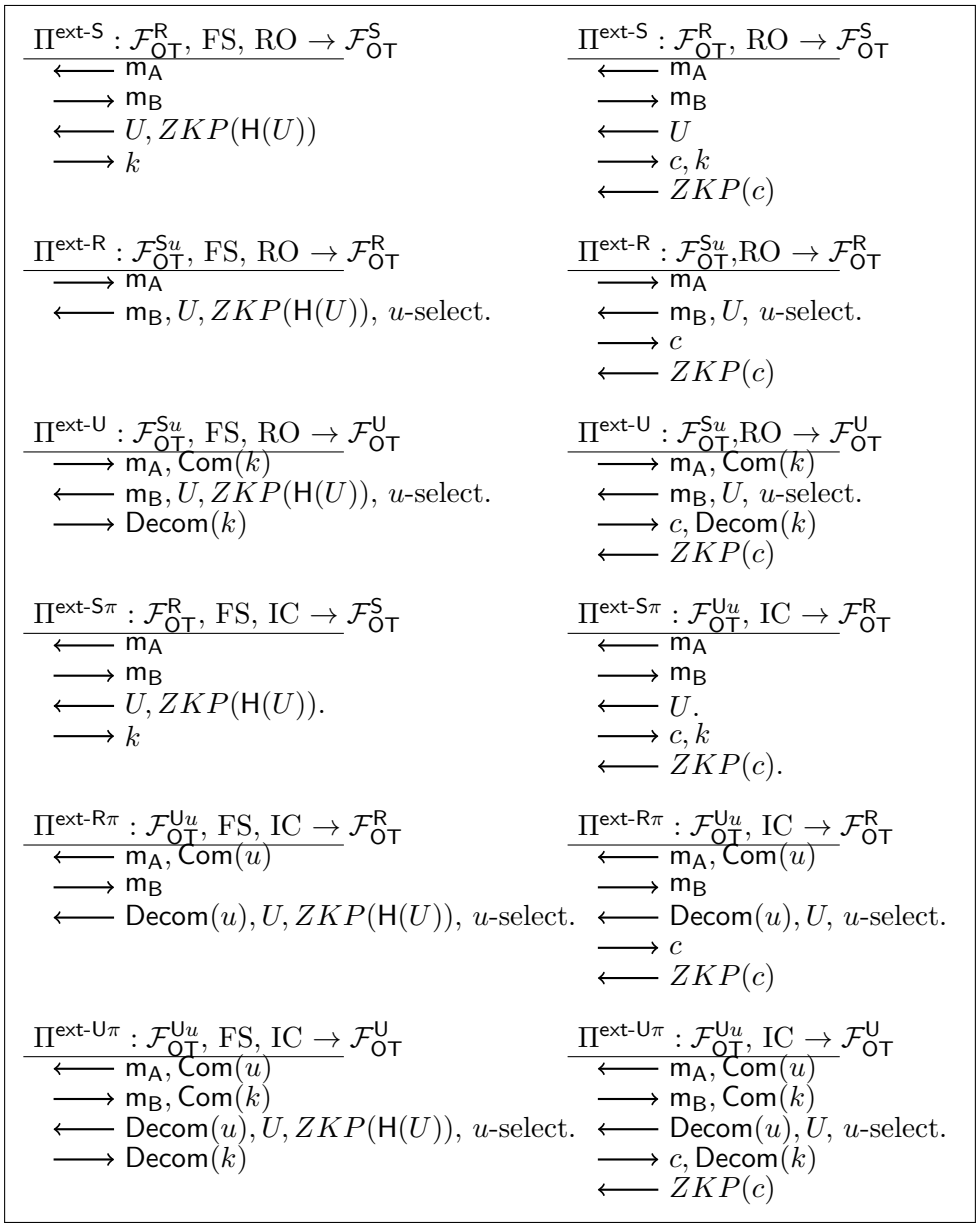

Figure 16: Messages flow for our various OT extension protocols. The protocols on the left have the Fiat-Shamir transformation applied, where the challenge value c is replaced with $\mathsf{H}(U)$. $\mathsf{m}_{\mathsf{A}}, \mathsf{m}_{\mathsf{B}}$ are the first and second messages of the base OTs. U is the OT extension matrix. $ZKP(x)$ is the proof that U is correct given a challenge value of $x \in \{\mathsf{H}(U), c\}$. u is a seed used to randomized the sender or receiver chosen message OTs into uniform message OTs. u must be committed to in the first round. u-select similarly transforms the receiver's selection into a uniform selection. k is the key used to generated the output messages as described in Section 5.

F.4 Proof of Lemma F.3 (Attack of $\Pi^{\mathsf{OOS+}}$)

Lemma F.3. *There exists a ppt adversary $\mathcal{A}$ and distinguisher D s.t. $\forall \mathcal{A}'$*

$$|\Pr[\mathsf{D}((\mathcal{A},\mathsf{R})_{\Pi^{OOS+}})=1]-\Pr[\mathsf{D}((\mathcal{A}',\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}))=1]|=1-\mathsf{negl}$$

where Π^{OOS+} is the protocol in Definition 5.4 and all algorithms additionally receive input 1^κ.

Proof. For simplicity let $N=2$ and $m=\kappa$. We define $\mathcal{A}$ as follows. $\mathcal{A}$ plays the role of S and replaces the input to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$, the receiver input, with the string $\boldsymbol{b}:=\{0\}^{nc}$. $\mathcal{A}$ outputs the matrix Q.

We define D as follows. D samples the selection bits $x_1,...,x_m \leftarrow [2]$ and sends them to R. D executes $\mathcal{A}$ who outputs Q and R outputs $\boldsymbol{v}_{1,x_1},...,\boldsymbol{v}_{m,x_m}$. If $\boldsymbol{v}_{i,x_i}=\mathsf{H}(i,\boldsymbol{q}_i)$ for all $i\in[m]$, output 1, otherwise 0. In the real interaction it clearly holds that $\Pr[\mathsf{D}((\mathcal{A},\mathsf{R})_{\Pi^{\mathsf{OOS+}}})=1]=1$ since $\boldsymbol{q}_i=\boldsymbol{t}_i$.

By definition the input of $\mathcal{A}'$ is independent of x_i and receives no output from $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ (apart from their input $(\boldsymbol{v}_{0,i},\boldsymbol{v}_{1,i})_{i\in[m]}$). Therefore, it must hold that $\Pr[\mathsf{D}((\mathcal{A}',\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}))=1]=2^\kappa$. □

F.5 Proof of Lemma F.7: $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ Extension with a Random Oracle

Proof.

Claim F.4 (Malicious Sender Security)**.** $\Pi^{\mathsf{ext\text{-}Su+}}$ *satisfies security against a malicious sender (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ oracle.*

Proof. The simulation follows the same strategy as Lemma 5.15 except now $\mathcal{A}$ is allowed to sample k and have the parties output messages of the form $\boldsymbol{v}_{i,x}:=\mathsf{H}(i,k+\boldsymbol{t}_i+\boldsymbol{b}\odot(\boldsymbol{c}_i+\mathcal{C}(map(x))))$. The simulator $\mathcal{A}'$ samples $\boldsymbol{t}_i$ uniformly at random after $\mathcal{A}$ is bound to their choice of k and therefore its easy to verify that $\mathcal{A}$ has negligible probability of querying H on such an input before receiving k. □

Claim F.5 (Malicious Receiver Security)**.** $\Pi^{\mathsf{ext\text{-}Su+}}$ *satisfies security against a malicious receiver (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ oracle.*

Proof. The simulation also follows the same strategy as Lemma 5.15 with a few key differences.

1. $\mathcal{A}'$ sends a dummy commitment in place of the commitment to k, i.e. a uniform string from the same distribution.

2. Then $\mathcal{A}'$ runs the normal simulation described by Lemma 5.15 up to the point that S would decommit to k except that $\mathcal{A}'$ does not program H as described.

3. At this point $\mathcal{A}'$ has received U in step Step 2 and $\mathcal{A}$ send a valid proof for Step 4 (by assumption or $\mathcal{A}'$ would have aborted). $\mathcal{A}'$ now uniformly samples $k\leftarrow\mathbb{F}_2^{nc}$ and programs the commitment random oracle to decommit to k. $\mathcal{A}'$ then programs H' to output the ideal output $\boldsymbol{v}_{i,x_i}$ of R for the query $\mathsf{H}'(i,k+\boldsymbol{t}_i)$. Since $k\in\mathbb{F}_2^{nc}$ is uniformly distributed in the view of $\mathcal{A}$, it follows that $\mathcal{A}$ has probability at most $q2^{-nc}\leq q2^{-\kappa}=\mathsf{negl}$ probability of querying the oracle at this point, where q is the number of queries that $\mathcal{A}$ has made.

4. $\mathcal{A}'$ then sends the decommits of k to $\mathcal{A}$ and completes the simulation as Lemma 5.15 does.

□

□

F.6 Proof of Lemma F.7: $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ Extension with an Ideal Cipher

Definition F.6. *Let $\Pi^{\mathsf{ext}\text{-}S\pi}$ be the protocol of Figure 9 where $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ and the random oracle $\mathsf{H}(i,x)$ required by $\Pi^{\mathsf{ext}\text{-}E}$ is replaced as follows: after Step 4, S samples $k \leftarrow \{0,1\}^\kappa$ and sends it to R. Both parties define $\mathsf{H}(x) = \pi_k(x) + x$ where $\pi : \{0,1\}^\kappa \times \mathbb{F}_2^{n_\mathcal{C}} \to \{0,1\}^\kappa$ is an ideal cipher. Note: the i parameter of H is removed.*

Lemma F.7. *The $\Pi^{\mathsf{ext}\text{-}S\pi}$ protocol realizes 1-out-of-N $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$-security, for $N = \mathsf{poly}(\kappa)$. Against a malicious R, $\Pi^{\mathsf{ext}\text{-}R\pi}$ realizes $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-security. That is, the input messages of an honest S are sampled uniformly from $\{0,1\}^\kappa$ by the protocol.*

Proof.

Claim F.8 (Malicious Sender Security)**.** *$\Pi^{\mathsf{ext}\text{-}R}$ satisfies security against a malicious sender (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ functionality.*

Proof. The simulation follows essentially the same strategy as Lemma 5.8. Consider the following hybrids which will define the simulator $\mathcal{A}'$.

Hybrid 1. $\mathcal{A}'$ internally runs $\mathcal{A}$ while plays the role of R and base OT oracle $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$. For $j \in [n_\mathcal{C}]$, $\mathcal{A}'$ receives $(\boldsymbol{b}'_j, \boldsymbol{t}^j_{\boldsymbol{b}_j}) \in [2] \times \mathbb{F}_2^{m'}$ from $\mathcal{A}$ in Step 1 where $\boldsymbol{b}_j := \boldsymbol{b}'_j - 1$. $\mathcal{A}'$ uniformly samples $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ as $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ would. $\mathcal{A}'$ sends $(\boldsymbol{b}', \{\boldsymbol{t}^j_{\boldsymbol{b}}\})$ to $\mathcal{A}$ on behalf of $\mathcal{F}_{\mathsf{OT}}$. $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. The view of $\mathcal{A}$ is unmodified.

Hybrid 2. For Step 2 $\mathcal{A}'$ does not sample $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ and instead uniformly samples $U \leftarrow \mathbb{F}_2^{m' \times n_\mathcal{C}}$. $\mathcal{A}'$ sends U to $\mathcal{A}$ and then computes Q as S would. The view of $\mathcal{A}$ is identically distributed. This follows from the fact that $\boldsymbol{t}^j_{1-\boldsymbol{b}_j}$ is uniformly distributed in the view of $\mathcal{A}$ and masks the j-th column of U in the previous hybrid.

Hybrid 3. For Step 4 $\mathcal{A}'$ simulates the consistency proof. This change is indistinguishable.

Hybrid 4. $\mathcal{A}'$ receives k from $\mathcal{A}$ as specified in Definition F.6. For each row $\boldsymbol{q}_i$, $\mathcal{A}'$ defines the circuit $\mathcal{M}_i : [N] \to \{0,1\}^\kappa$ such that on input $j \in [N]$ it outputs $\mathsf{H}(\boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(j)))$. $\mathcal{A}'$ sends $\mathcal{M}_i$ to the ideal oracle $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$ as the sender's input to the i-th OT instance. This change allows the ideal oracle to output the same distribution as the real protocol. The view of $\mathcal{A}$ is unmodified.

Let $y_j = \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(map(j)) = \boldsymbol{t}_i + \boldsymbol{b} \odot (c_i + \mathcal{C}(map(j))$ and note that $\mathcal{A}$ can influence $\mathcal{M}_i(j) = \mathsf{H}(y_j) = \pi_k(y_j) + y_j$ by choosing $k, \boldsymbol{b}$ and the bits $\{\boldsymbol{t}_i[j] \mid \boldsymbol{b}_j = 0\}$.

Hybrid 5. $\mathcal{A}'$ does not take the input of R. R only interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{E}}$. This change is identically distributed since $\mathcal{A}'$ was not using the input of R.

□

Claim F.9 (Malicious Receiver $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-Security)**.** *$\Pi^{\mathsf{ext}\text{-}R}$ satisfies security against a malicious receiver (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ functionality.*

Proof. The simulation also follows a similar strategy as Lemma 5.8. Consider the following hybrids which will define the simulator $\mathcal{A}'$.

Hybrid 1. $\mathcal{A}'$ internally runs $\mathcal{A}$ while plays the role of S and base OT oracle $\mathcal{F}_{\mathsf{OT}} = \mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$. $\mathcal{A}'$ uniformly samples $\{\boldsymbol{t}^j_0, \boldsymbol{t}^j_1\}_{i \in n_\mathcal{C}}$ and sends them to $\mathcal{A}$ in Step 1. $\mathcal{A}'$ samples $\boldsymbol{b}$ as S would. $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. The view of $\mathcal{A}$ is unmodified.

Hybrid 2. In Step 2 $\mathcal{A}'$ receives U from $\mathcal{A}$. $\mathcal{A}'$ computes C and Q using $\boldsymbol{t}_0^j, \boldsymbol{t}_1^j, \boldsymbol{b}$. $\mathcal{A}'$ performs the proof of Step 4 as S would. If the proof fails, $\mathcal{A}'$ aborts as S would. Otherwise, by the correctness of the proof, $\boldsymbol{c}_i$ decodes to $\boldsymbol{w}_i$ and computes x_i s.t. $\boldsymbol{w}_i = map(x_i)$.

For all $i \in [m]$, $\mathcal{A}'$ defines the circuit $\mathcal{S}_i : [N] \to \{0,1\}$ which outputs 1 at x_i and 0 otherwise. $\mathcal{A}'$ sends $\mathcal{S}_i$ and then (Output, x_i) to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ as the receiver's input to the i-th $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ instance which responds with v_{i,x_i}. The view of $\mathcal{A}$ is unmodified.

Hybrid 3. $\mathcal{A}'$ then uniformly samples $k \leftarrow \{0,1\}^\kappa$ as S would and defines the ideal permutation π_k. If π_k has been queries by $\mathcal{A}$, then $\mathcal{A}'$ aborts. The probability of this event is negligible due to k being uniformly sampled from $\{0,1\}^\kappa$. Otherwise, before sending k to S, $\mathcal{A}'$ programs π_k s.t. $\pi_k(\boldsymbol{t}_i) = \boldsymbol{v}_{i,x_i} + \boldsymbol{t}_i$. Conditioned on these input/outputs not colliding for $i \in [m]$, which happens with overwhelming probability, this modification is identically distributed due to $\boldsymbol{v}_{i,x_i} \leftarrow \{0,1\}^\kappa$ being sampled uniformly by $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$.

Hybrid 4. Assuming $\mathcal{A}'$ did not abort in Step 4, let $E = \{j \mid \exists i \in [m], (\boldsymbol{c}_i \oplus \mathcal{C}(\boldsymbol{w}_i))_j = 1\}$ index the columns of C where $\mathcal{A}$ added an error to any codeword $\boldsymbol{c}_i$ (w.r.t $\boldsymbol{w}_i$). By the correctness of Step 4, it holds that $E \subseteq B_0$, otherwise the consistency proof would have failed. By passing the consistency proof, $\mathcal{A}$ learns what $\boldsymbol{b}_j = 0$ for all $j \in E$. Similarly, the probability of passing the check and $\Pr[|E| = d] = \Pr[\boldsymbol{b}_j = 0 \mid \forall j \in E] = 2^{-d}$ due to the proof being independent of $\boldsymbol{b}$. We will see that this is equivalent to $\mathcal{A}$ simply guessing E (which is correct with the same probability) and then being honest.

For all $w \neq \boldsymbol{w}_i$, $\mathcal{A}$ has negl probability of computing $g = \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w)$. If this was not the case, then $\mathcal{A}$ could compute

$$\begin{aligned} g + \boldsymbol{t}_i &= \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w) + \boldsymbol{t}_i \\ &= \boldsymbol{c}_i \odot \boldsymbol{b} + \boldsymbol{t}_i + \boldsymbol{b} \odot \mathcal{C}(w) + \boldsymbol{t}_i \\ &= (\boldsymbol{c}_i + \mathcal{C}(w)) \odot \boldsymbol{b} \\ &= (\mathcal{C}(\boldsymbol{w}_i) + \mathcal{C}(w)) \odot \boldsymbol{b} \end{aligned}$$

This last equality holds due to $\mathcal{A}'$ aborting if $(\boldsymbol{c}_i + \mathcal{C}(\boldsymbol{w}_i)) \odot \boldsymbol{b} \neq 0$. Recall that $\mathcal{C}$ has minimum distance $d_{\mathcal{C}} \geq \kappa$ and therefore computing g is equivalent $\mathcal{A}$ guessing $d_{\mathcal{C}} \geq \kappa$ bits of $\boldsymbol{b}$ which happens with probability $2^{-d_{\mathcal{C}}} \leq 2^{-\kappa}$. As such, the probability that $\mathcal{A}$ has made a query of the form $\pi_k(\boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w))$ for $w \neq \boldsymbol{w}_i$ is also negligible. If such as query does happen $\mathcal{A}'$ aborts. This hybrid is indistinguishably distributed from the previous.

Hybrid 5. When S makes an π_k query of the form $\pi_k(h)$ which they have not previously been queried, $\mathcal{A}'$ must determine if there is a unique $w \in \mathbb{F}_2^{n_{\mathcal{C}}}, i \in [m]$ such that $h = \boldsymbol{q}_i + \boldsymbol{b} \odot \mathcal{C}(w)$. For the sake of contradiction, let us assume there exists any two $i, i' \in [m]$ or $w, w' \in \mathbb{F}_2^{k_{\mathcal{C}}}$ which result in the same input to π_k. If $i = i'$ and $w = w'$, then a unique (i, w) exist. Otherwise,

$$\begin{aligned} \boldsymbol{t}_i + \boldsymbol{b} \odot (c_i + \mathcal{C}(w)) &= \boldsymbol{t}_{i'} + \boldsymbol{b} \odot (c_{i'} + \mathcal{C}(w')) \\ \boldsymbol{b} \odot (\mathcal{C}(w) + \mathcal{C}(w') + \boldsymbol{c}_i + \boldsymbol{c}_{i'}) &= \boldsymbol{t}_i + \boldsymbol{t}_{i'} \\ \boldsymbol{b} \odot \delta &= \boldsymbol{t}_i + \boldsymbol{t}_{i'} \end{aligned}$$

where $\delta := \mathcal{C}(w) + \mathcal{C}(w') + \boldsymbol{c}_i + \boldsymbol{c}_{i'}$. If $i = i'$, then it must hold $\boldsymbol{b} \odot (\mathcal{C}(w) + \mathcal{C}(w')) = 0$ for $w \neq w'$. Recall that $\mathcal{C}$ by construction has minimum distance $d_{\mathcal{C}} \geq \kappa$ and that $\boldsymbol{b}$ is

uniformly distributed. Let $E = \{i \mid \delta_i = 1\}$, then $|E| \geq d_{\mathcal{C}} \geq \kappa$ and for the above to hold we require $\boldsymbol{b}_i = 0 \mid \forall i \in E$ which occurs with probability $\Pr[\boldsymbol{b}_i = 0 \mid \forall i \in E] = 2^{-|E|} \leq 2^{-d_{\mathcal{C}}} \leq 2^{-\kappa}$. Therefore with overwhelming probability a unique (i, w) exist if $i = i'$.

Otherwise, let $B_j := \{i \mid \boldsymbol{b}_i = j\}$ and due to Step 4 it holds that for all $i \in [m]$, $\boldsymbol{c}_i \odot \boldsymbol{b}$ erasure decodes to $\boldsymbol{w}_i$ with B_0 indexing the erasures. Therefore, by the linearity of $\mathcal{C}$, δ erasure decodes to some w^* with B_0 indexing the erasures s.t. $\boldsymbol{b} \odot c^* = \boldsymbol{b} \odot \delta$ where $c^* := \mathcal{C}(w^*)$.

Fixing some i, i', the probability $\boldsymbol{b} \odot c^* = \boldsymbol{t}_i + \boldsymbol{t}_{i'}$ is $p_0 = \Pr[(\boldsymbol{t}_i + \boldsymbol{t}_{i'})_\ell = 0 \mid \forall \ell \in B_0] \leq 2^{-|B_0|}$ times $p_1 = \Pr_{c^*}[(\boldsymbol{t}_i + \boldsymbol{t}_{i'} + c^*)_\ell = 0 \mid \forall \ell \in B_1] \leq N 2^{-|B_1|}$. Therefore, the probability that $i \neq i'$ and $w \neq w'$ is at most the union bound over all $i, i' \in [m]$,

$$\Pr_{i,i',c^*}[\boldsymbol{b} \odot c^* \leq \boldsymbol{t}_i + \boldsymbol{t}_{i'}] \leq m^2 p_0 p_1 = m^2 N 2^{-n_{\mathcal{C}}} \tag{2}$$

which is negligible[8]. Therefore we conclude that (i, w) is unique if such a pair exists.

If so then $\mathcal{A}'$ can use Gaussian elimination to identify it. In particular, $\mathcal{A}'$ computes $h + \boldsymbol{q}_i$ for all $i \in [m]$ and checks that $(h + \boldsymbol{q}_i)_\ell = 0$ for all $\ell \in B_1$ and if so tries erasure decodes $h + \boldsymbol{q}_i$ to w where the erasures are index by B_0. For $h + \boldsymbol{q}_i$ this will happen and $\mathcal{A}'$ computes x s.t. $map(x) = w$ and sends (OUTPUT, x) to the i-th instance of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ and receives $\boldsymbol{v}_{i,x} \leftarrow \{0,1\}^\ell$ in response. Let $y_{i,x} := h = \boldsymbol{t}_i + \boldsymbol{b}(\boldsymbol{c}_i + \mathcal{C}(map(x)))$.

$\mathcal{A}'$ programs $\pi_k(y_{i,x}) = \boldsymbol{v}_{i,x} + y_{i,x}$. Programming π_k requires the input/output pair $(y_{i,x}, \boldsymbol{v}_{i,x} + y_{i,x})$ to have not previously been queried on π_k, π_k^{-1}. It is easy to verify that with overwhelming probability $\pi_k^{-1}(\boldsymbol{v}_{i,x} + y_{i,x})$ has not been queried since $\boldsymbol{v}_{i,x}$ is uniformly distributed.

In the other direction, $y_{i,x}$ could have been queried in two ways. 1) D or $\mathcal{A}$ guessed it which is negligible as discussed in Hybrid F.6. 2) D inverted $\boldsymbol{v}_{i',x'} := \mathsf{H}(y_{i',x'}) = \pi_k(y_{i',x'}) + y_{i',x'}$ and then recovered $\boldsymbol{b}$. However, $v = \pi_k(y) + y$ is preimage resistant[BRS02, Win84] which informally follows from the difficulty of finding an input to the random permutation π_k which differs from v by itself.

Hybrid 6. $\mathcal{A}'$ does not take the input of S and does not program π in Hybrid F.6. S only interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$. This change is identically distributed.

□

Claim F.10 (Malicious Receiver $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$-Security)**.** $\Pi^{\mathsf{ext\text{-}R}}$ *satisfies Security Against a Malicious Receiver (Definition 2.6) with respect to the* $\mathcal{F}_{\mathsf{OT}}^{\mathsf{S}}$ *functionality.*

Proof. Follows from Lemma 3.1 and the previous claim. □

□

Definition F.11. *Let* $\Pi^{\mathsf{ext\text{-}U}\pi}$ *be the protocol of Figure 9 where* $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}_u}$ *and the random oracle* $\mathsf{H}(i, x)$ *required by Figure 9 is replaced as follows:*

1. *In round one,* S *samples* $k \leftarrow \{0,1\}^\kappa$ *and sends a commitment of* k *to* R.
2. *After Step 4,* S *decommits* k *to* R *who aborts if it fails.*
3. *Both parties define* $\mathsf{H}(x) = \pi_k(x) + x$ *where* $\pi : \{0,1\}^\kappa \times \mathbb{F}_2^{n_{\mathcal{C}}} \to \{0,1\}^\kappa$ *is an ideal cipher. Note: the* i *parameter of* H *is removed.*

[8] Note, N is assumed to be polynomial. This is true in the target use case where $N = 2$ and $n_{\mathcal{C}} = \kappa$.

F.7 Proof of Lemma F.11: $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ Extension with an Ideal Cipher

Lemma F.12. *The $\Pi^{\mathsf{ext\text{-}U}\pi}$ protocol realizes 1-out-of-N $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-security, for $N = \mathsf{poly}(\kappa)$.*

Proof.

Claim F.13 (Malicious Sender Security)**.** *$\Pi^{\mathsf{ext\text{-}U}}$ satisfies security against a malicious sender (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ functionality.*

Proof. The simulation follows essentially the same strategy as Lemma F.7. The differences to the hybrids are as follows.

Hybrid 1. $\mathcal{A}'$ extracts k from the commitment. Then $\mathcal{A}'$ samples T_0, T_1 and the selections $\boldsymbol{b}$ uniformly at random and simulates the base OTs using them.

Hybrid 4. $\mathcal{A}'$ no longer sends the messages specified by S to $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$. Instead, when $\mathcal{A}$ makes a query to $\pi_k(h)$, $\mathcal{A}'$ checks if $h = y_{i,x} = \boldsymbol{t}_i + \boldsymbol{b} \odot (\boldsymbol{c}_i + \mathcal{C}(map(x)))$ for some pair (i, x). If so, then (i, x) are unique as described by Lemma F.7. $\mathcal{A}'$ queries the i-th instance of $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ with $(\textsc{Output}, x)$ and receives $\boldsymbol{v}_{i,x}$ in response. $\mathcal{A}'$ programs $\pi_k(y_{i,x}) = \boldsymbol{v}_{i,x} + y_{i,x}$. The probability of the input/output being previously queries is negligible due to $\mathcal{A}$ extracting k before $\boldsymbol{t}_i$ was sampled and $\boldsymbol{v}_{i,x}$ being uniformly distributed.

Hybrid 5. $\mathcal{A}'$ does not take the input of R. R only interacts with $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$. This change is identically distributed since $\mathcal{A}'$ was not using the input of R.

□

Claim F.14 (Malicious Receiver $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$-Security)**.** *$\Pi^{\mathsf{ext\text{-}R}}$ satisfies security against a malicious receiver (Definition 2.6) with respect to the $\mathcal{F}_{\mathsf{OT}}^{\mathsf{U}}$ oracle.*

Proof. Follows directly from Lemma F.7 claim 2 and the hiding property of the commitment. □

□

F.8 $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$ Extension with an Ideal Cipher

Definition F.15. *Let $\Pi^{\mathsf{ext\text{-}R}\pi}$ be the protocol of Figure 9 where $\mathcal{F}_{\mathsf{OT}} := \mathcal{F}_{\mathsf{OT}}^{\mathsf{U}u}$ and the random oracle $\mathsf{H}(i, x)$ required by $\Pi^{\mathsf{ext\text{-}E}}$ is replaced as follows: $\mathsf{H}(x) = \pi(x) + x$ where $\pi : \mathbb{F}_2^{n_C} \to \{0,1\}^\kappa$ is an ideal permutation. Note: the i parameter of H is removed.*

Lemma F.16. *The $\Pi^{\mathsf{ext\text{-}R}\pi}$ protocol realizes 1-out-of-N $\mathcal{F}_{\mathsf{OT}}^{\mathsf{R}}$-security, for $N = \mathsf{poly}(\kappa)$.*

sketch. The proof follows the same strategy as Lemma F.7 except π is not keyed. As such, R can compute $\mathsf{H}(\boldsymbol{t}_i)$ *before* making their selection x_i. This can be simulator by having the simulator extract $\mathsf{H}(\boldsymbol{t}_i)$ as their chosen message. □