On the Security of KOS

Benjamin E. Diamond*
Ulvetanna
benediamond@gmail.com

Abstract

We study the security of the random oblivious transfer extension protocol of Keller, Orsini, and Scholl (CRYPTO '15), whose security proof was recently invalidated by Roy (CRYPTO '22). We show that KOS is asymptotically secure. Our proof involves a subtle analysis of the protocol's "correlation check", and introduces several new techniques. We also study the protocol's concrete security. We establish concrete security for security parameter values on the order of 5,000. We present evidence that a stronger result than ours—if possible—is likely to require radically new ideas.

1 Introduction

The oblivious transfer extension protocol of Keller, Orsini and Scholl [KOS15, Fig. 7] (henceforth "KOS") is widely known and used. Key to that protocol is a certain "correlation check", in which a number of extension OTs are "sacrificed" in a linear combination. This check is very difficult to analyze. In recent work, Roy [Roy22, § 4.1] disproves a key lemma [KOS15, Lem. 1], upon which KOS's security analysis relies. Roy's work invalidates the security proof [KOS15, Thm. 1], as originally written.

In a recent update to their work, Keller, Orsini and Scholl propose an adjusted variant of their protocol [KOS22, Fig. 10]; essentially, they suggest a special case of Roy's construction. Though the efficiency of the updated protocol is comparable to the original, it is more complex, and uses different ideas. Indeed, we note that the analysis of [Roy22] is very theoretically involved. It is of interest to prove the security of KOS, as originally written; this open problem is noted explicitly by Roy [Roy22, § 1.1], for example.

We show that, asymptotically, KOS is secure. Our proof begins by introducing a certain numerical metric, which captures the *extent* of the corrupt receiver's compliance with the protocol. We moreover introduce a new simulation strategy, based on this metric, and show that—as this degree of compliance varies—the receiver must choose between facing negligible odds in the correlation check, on the one hand, and handing the distinguisher a negligible advantage, on the other. Our proof's key step has a coding-theoretic flavor; we show that a binary matrix with sufficiently many random columns is unlikely to reside *near* the matrix representation of any field element (in the space of matrices, where *distance* is measured in rank).

We also extract effective bounds from our proof. We show that, in order to achieve statistical security of 2^{-40} against an adversary making up to 2^{80} hash evaluations, the security parameter $\kappa = 5{,}122$ suffices (see Example 3.17). More abstractly, we show that KOS, instantiated with security parameter κ, withstands an attacker making up to $\frac{1}{2} \cdot \sqrt{\kappa} \cdot 2^{\frac{1}{2} \cdot \sqrt{\kappa}}$ hash evaluations with statistical security $2^{-\frac{1}{2} \cdot \sqrt{\kappa}}$ (see Corollary 3.19).

Obviously, this sort of κ results in a barely-practical protocol. On the other hand, we give evidence that this limitation might be intrinsic. As it turns out, our proof applies equally well to the security of Patra, Sarkar and Suresh [PSS17] (henceforth "PSS"), another protocol attacked by Roy [Roy22, § 4.1]. (Indeed, our proof invokes *only* properties of KOS which are shared by PSS; we discuss this fact further below.) Interestingly, our lower-bound tightly matches—up to the factor of $\frac{1}{2}$ present in both exponents—the upper-bound achieved by Roy [Roy22, § 4.1] on PSS. Our proof thus definitively settles the question of PSS's security, up to these constants. (As we explain in Remark 3.20 below, these constants may in fact be taken as high as $\frac{1}{\sqrt{2}} - \varepsilon$, for $\varepsilon > 0$ arbitrarily small.) It also shows that a sharper analysis of KOS—if possible at all—would have to rely on features of KOS's correlation check more delicate than those our proof considers.

*I would like to sincerely thank a handful of anonymous referees for extremely valuable feedback.

We briefly recall the details of KOS. The correlation check (see [KOS15, Fig. 7]) serves to control the row-vectors $(\mathbf{x}_i)_{i=0}^{l'-1}$ the receiver submits to the *correlated OT with errors* hybrid functionality $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$. If the receiver is honest, then each row $\mathbf{x}_i \in \mathbb{F}_2^\kappa$ is necessarily "monochromatic" (in the sense that its components are identical); if the receiver's rows $\mathbf{x}_i$ are *not* monochromatic, then the corrupt receiver may facilitate the distinguisher's learning certain bits of the sender's correlation vector $\Delta \in \mathbb{F}_2^\kappa$, by means of brute-force queries to the random oracle. The correlation check prescribes that the parties jointly sample random elements $(\chi_i)_{i=0}^{l'-1}$ from $\mathbb{F}_{2^\kappa}$, using a coin-flipping functionality, that they subject their intermediate values—i.e., those they obtained from $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$—to a linear combination using these coefficients, and finally that they exchange the results. Specifically, the sender and receiver, having received $(\mathbf{q}_i)_{i=0}^{l'-1}$ and $(\mathbf{t}_i)_{i=0}^{l'-1}$, respectively, from $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, compute $q := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$ and $t := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$, respectively; the (honest) receiver moreover computes $x := \sum_{i=0}^{l'-1} \chi_i \cdot x_i$, where $(x_i)_{i=0}^{l'-1}$ is its choice vector. Finally, the receiver sends x and t to the sender, who checks $q \stackrel{?}{=} t + x \cdot \Delta$. All multiplications here take place in the binary field $\mathbb{F}_{2^\kappa}$.

Informally, the correlation check controls whether the *individual* equalities $\mathbf{q}_i \stackrel{?}{=} \mathbf{t}_i + x_i \cdot \Delta$ hold, for each $i \in \{0, \ldots, l'-1\}$, or—equivalently—whether the vector $(\mathbf{q}_i + \mathbf{t}_i + x_i \cdot \Delta)_{i=0}^{l'-1} \in \mathbb{F}_{2^\kappa}^{l'}$ is the zero vector. In actuality, however, the correlation check merely checks whether this latter vector resides within the *random hyperplane* in $\mathbb{F}_{2^\kappa}^{l'}$ given by the coefficients $(\chi_i)_{i=0}^{l'-1}$. The difficulty is that the corrupt receiver sees these coefficients—and the resulting hyperplane—*before* sending x and t; as a result, the receiver could conceivably select these values in such a way that the vector $(\mathbf{q}_i + \mathbf{t}_i + x_i \cdot \Delta)_{i=0}^{l'-1}$—though *nonzero*—nonetheless resides within this hyperplane, and causes the check to pass. This is precisely the subtlety overlooked by [KOS15, Thm. 1]; we refer to [Roy22, § 4.1] for discussion.

We now sketch the technical details of our proof (see also Theorem 3.1). Our treatment of the corrupt sender is similar to that of [KOS15, Thm. 1] (though we supply certain details which were omitted from that proof). Our treatment of the corrupt receiver—the more difficult case—relies on a new simulation strategy for that case, as well as on a careful analysis of the adversary's and distinguisher's success conditions.

We begin by introducing a numerical characterization of the *extent* of $\mathcal{A}$'s matrix's monochromaticity. This metric—which we call the matrix's *modesty*—ranges throughout $m \in \{1, \ldots, \kappa\}$; an honest receiver necessarily has modesty κ. Informally, m measures the feasibility of assigning choice bits $(x_i)_{i=0}^{l-1}$ to $(\mathbf{x}_i)_{i=0}^{l'-1}$'s first l rows in such a way that it becomes difficult to *use* vectors of the form $\mathbf{x}_i + x_i \cdot (1, \ldots, 1)$ to *assemble* vectors of the form $\mathbf{x}_i + \overline{x_i} \cdot (1, \ldots, 1)$ (we write $\mathbf{e}_i$ and $\overline{\mathbf{e}_i}$, respectively, for these latter two vectors). Specifically, we say that $(\mathbf{x}_i)_{i=0}^{l-1}$ has modesty m if there exists an assignment $(x_i)_{i=0}^{l-1}$ with the property that, even after arbitrarily including vectors $\mathbf{e}_i$ which each successively introduce fewer than m new bit positions, we nonetheless remain at least m bit positions away from each vector $\overline{\mathbf{e}_i}$, where, here, $i \in \{0, \ldots, l-1\}$ varies arbitrarily (and if moreover $m \in \{1, \ldots, \kappa\}$ is the largest integer with this property). Our Definition 3.3 below serves the dual role of determining $m \in \{1, \ldots, \kappa\}$ *and* of producing the assignment $(x_i)_{i=0}^{l-1}$.

We show that as $m \in \{1, \ldots, \kappa\}$ varies, $\mathcal{A}$ smoothly trades off between two different bad outcomes. (A graphical depiction of our proof strategy is given in Figure 1 below.) On the one hand, if $\mathcal{A}$'s modesty is low—that is, if $\mathcal{A}$ cheats brazenly—then $\mathcal{A}$'s probability of passing the correlation check becomes low. Indeed, we note that—up to a uniform resampling of the random combination coefficients $(\chi_i)_{i=0}^{l'-1}$ used in the check—we may freely assume that the matrix $(\mathbf{e}_i)_{i=0}^{l'-1}$ is in reduced row-echelon form; we further note that, as m decreases, this reduced matrix accumulates pivots. These pivots impose independently random $\mathbb{F}_2$-linear conditions on the unknown vector Δ, and make the correlation check harder to pass. We reduce $\mathcal{A}$'s success to a coding-inspired condition on the space of binary matrices, and upper-bound its probability of passing using a union bound (see Proposition 3.9).

On the other hand, we show that as m grows—that is, as $\mathcal{A}$ becomes *more* compliant—$\mathcal{A}$ begins producing transcripts which make the presence of our simulation harder to detect. Indeed, any given distinguisher may learn bits of the hidden choice vector Δ *only* by means of brute-force queries of the form $H(i \,\|\, \mathbf{t}_i + \mathbf{e}_i * \Delta)$, where $\mathbf{e}_i$ introduces few new bit positions not already learned. On the other hand, the distinguisher may successfully distinguish the real and ideal distributions only if it manages to query $H(i \| \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$. Effectively, m controls the size of the the minimal-length "stretch" of bits which the distinguisher must brute-force, if it is to succeed (see Proposition 3.12).

By combining these two cases, we establish the result (see Theorem 3.1).

2 Background and Notation

We identify $\{0,1\} \cong \mathbb{F}_2$ as *sets*. We occasionally identify *vectors* in $\{0,1\}^\kappa \cong \mathbb{F}_2^\kappa$ with *subsets* of $\{0,\ldots,\kappa-1\}$, in the standard way; that is, for each vector $\mathbf{d} \in \{0,1\}^\kappa$, corresponding to the map $\widehat{\mathbf{d}} : \{0,\ldots,\kappa-1\} \to \{0,1\}$, say, we identify $\mathbf{d}$ with the subset $\widehat{\mathbf{d}}^{-1}(1) \subset \{0,\ldots,\kappa-1\}$ (i.e., with the set of components at which $\mathbf{d}$ is 1). We use the symbol $*$ to denote bitwise AND in $\mathbb{F}_2^\kappa$, and write w for Hamming weight. We use the symbol $\setminus$ to denote set subtraction. We fix a field structure on $\mathbb{F}_{2^\kappa}$—that is, an irreducible polynomial of degree κ in $\mathbb{F}_2[X]$—and identify $\mathbb{F}_{2^\kappa}$ with the $\mathbb{F}_2$-vectorspace $\mathbb{F}_2^\kappa$, by means of the basis $(1, X, \ldots, X^{\kappa-1})$. We write $\cdot$ for field multiplication. In what follows, we make use of linear and affine-linear algebra over $\mathbb{F}_2$, without further comment; for this, we suggest the reference Cohn [Coh82, § 5].

Following [KOS15, § 2], we write κ for a security parameter. We write λ and s for *desired* levels of computational and statistical security, respectively. We write $(\mathbf{x}_i)_{i=0}^{l'-1}$ for the rows of an $l' \times \kappa$ matrix. We write $\overline{\mathbf{x}_i}$ for the bitwise complement of a row-vector $\mathbf{x}_i \in \mathbb{F}_2^\kappa$, and $\overline{x_i}$ for the complement of a bit $x_i \in \mathbb{F}_2$.

2.1 Secure computation

Given two probability distributions $\mathcal{Y}_0$ and $\mathcal{Y}_1$ on $\{0,1\}^\kappa$, the *statistical distance* between $\mathcal{Y}_0$ and $\mathcal{Y}_1$ is defined to be $\frac{1}{2} \cdot \sum_{\mathbf{y} \in \{0,1\}^\kappa} |\Pr[\mathcal{Y}_0 = \mathbf{y}] - \Pr[\mathcal{Y}_1 = \mathbf{y}]|$. We say that two distribution ensembles $\{\mathcal{Y}_0(a,\kappa)\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ and $\{\mathcal{Y}_1(a,\kappa)\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ are *statistically indistinguishable* if there is a negligible function μ such that for each $a \in \{0,1\}^*$ and each $\kappa \in \mathbb{N}$, the statistical distance between $\mathcal{Y}_0(a,\kappa)$ and $\mathcal{Y}_1(a,\kappa)$ is at most $\mu(\kappa)$. We say that two distribution ensembles $\{\mathcal{Y}_0(a,\kappa)\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ and $\{\mathcal{Y}_1(a,\kappa)\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ are *computationally indistinguishable* if, for each probabilistic, polynomial-time distinguisher D, the distributions ensembles $\{D(\mathcal{Y}_0(a,\kappa))\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ and $\{D(\mathcal{Y}_1(a,\kappa))\}_{a \in \{0,1\}^*, \kappa \in \mathbb{N}}$ on $\{0,1\}$ are statistically indistinguishable.

We record the definition of maliciously secure two-party computation, following Lindell [Lin17, § 6.6.2].

Definition 2.1. For each functionality $\mathcal{F}$, a protocol Π, real-world adversary $\mathcal{A}$, simulator $\mathcal{S}$, and corrupt party $C \in \{0,1\}$, we have the distributions:

- $\mathsf{Real}_{\Pi,\mathcal{A},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)$: Run Π with security parameter κ, where the honest party P_{1-C} uses the input $\mathbf{x}_{1-C}$, and $\mathcal{A}$ controls the messages of the corrupt party. Return the outputs of $\mathcal{A}$ and P_{1-C}.
- $\mathsf{Ideal}_{\mathcal{F},\mathcal{S},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)$: Run $S(1^\kappa, C, \mathbf{x}_C)$ until it outputs a value $\mathbf{x}'_C$, or else outputs (`abort`) to $\mathcal{F}$, who halts. Give $\mathbf{x}_{1-C}$ and $\mathbf{x}'_C$ to $\mathcal{F}$, and obtain outputs (v_0, v_1). Give v_C to $\mathcal{S}$; if $\mathcal{S}$ outputs (`abort`), then $\mathcal{F}$ outputs (`abort`) to P_{1-C}; otherwise, $\mathcal{F}$ gives P_{1-C} v_{1-C}. Return the outputs of $\mathcal{S}$ and P_{1-C}.

We say that Π *securely computes* $\mathcal{F}$ *in the presence of one static malicious corruption with abort*, or that Π *securely computes* $\mathcal{F}$, if, for each corrupt party $C \in \{0,1\}$ and each probabilistic polynomial-time adversary $\mathcal{A}$ corrupting P_C, there is a probabilistic expected polynomial-time simulator $\mathcal{S}$ corrupting P_C in the ideal world such that the distributions $\{\mathsf{Real}_{\Pi,\mathcal{A},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)\}_{(\mathbf{x}_0,\mathbf{x}_1),\kappa}$ and $\{\mathsf{Ideal}_{\mathcal{F},\mathcal{S},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)\}_{(\mathbf{x}_0,\mathbf{x}_1),\kappa}$ are computationally indistinguishable, where $\mathbf{x}_0$ and $\mathbf{x}_1$ are required throughout to have equal lengths.

2.2 Oblivious transfer

We recall background material on oblivious transfer, following [KOS15].

FUNCTIONALITY 2.2 ($\mathcal{F}_{\mathsf{Rand}}^\kappa$—coin-flipping functionality [KOS15, Fig. 5])**.**
The security parameter κ and players S and R are fixed.

- Upon receiving $(\mathtt{random}, i)$ from both players, $\mathcal{F}_{\mathsf{Rand}}^\kappa$ samples $\chi_i \leftarrow \mathbb{F}_2^\kappa$, and outputs $(\mathtt{random}, i, \chi_i)$ to both players.

FUNCTIONALITY 2.3 ($\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$—correlated OT with errors [KOS15, Fig. 2])**.**
The security parameter κ, the number l of resulting OTs, and players S and R are fixed.

- Upon receiving $(\texttt{initialize}, \Delta)$ from S, where $\Delta \in \mathbb{F}_2^\kappa$, $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$ stores Δ.
- If both parties are honest, R submits $\left(\texttt{input}, (\mathbf{x}_i)_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$, which, for each $i \in \{0, \ldots, l-1\}$, samples $\mathbf{t}_i \leftarrow \mathbb{F}_2^\kappa$ randomly and computes $\mathbf{q}_i := \mathbf{t}_i + \mathbf{x}_i * \Delta$.
- If R is corrupt, R submits $\left(\texttt{input}, (\mathbf{x}_i)_{i=0}^{l-1}, (\mathbf{t}_i)_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$, which computes $(\mathbf{q}_i)_{i=0}^{l-1}$ identically.
- If S is corrupt, S submits $\left(\texttt{input}, (\mathbf{q}_i)_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$, which, for each $i \in \{0, \ldots, l-1\}$, sets $\mathbf{t}_i := \mathbf{q}_i + \mathbf{x}_i * \Delta$.
- In each case, $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$ outputs $\left(\texttt{output}, (\mathbf{t}_i)_{i=0}^{l-1}\right)$ to R and $\left(\texttt{output}, (\mathbf{q}_i)_{i=0}^{l-1}\right)$ to S.

We note that $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$ can be securely instantiated by the protocol of [KOS15, Fig. 3].

Remark 2.4. We slightly alter the treatment of [KOS15, Fig. 2], in that we permit the corrupt sender S to choose its values $(\mathbf{q}_i)_{i=0}^{l-1}$. This privilege appears necessary for the secure instantiation of $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$ (in the $\mathcal{F}_{\mathsf{OT}}^{\kappa}$-hybrid model) to go through; its omission appears to have been an oversight on the part of [KOS15].

We moreover recall the *random OT* functionality:

FUNCTIONALITY 2.5 ($\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$—random OT functionality [KOS15, Fig. 6])**.**
The security parameter κ, the number l of resulting OTs, and players S and R are fixed.

- If both parties are honest, R submits $\left(\texttt{input}, (x_i)_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, which, for each $i \in \{0, \ldots, l-1\}$, samples $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1}) \leftarrow \{0,1\}^\kappa \times \{0,1\}^\kappa$.
- If R is corrupt, R submits $\left(\texttt{input}, (x_i)_{i=0}^{l-1}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, which, for each $i \in \{0, \ldots, l-1\}$, samples $\mathbf{v}_{i,\overline{x_i}} \leftarrow \{0,1\}^\kappa$.
- If S is corrupt, then S submits $\left(\texttt{input}, (\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$.
- In each case, $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ outputs $\left(\texttt{output}, (\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}\right)$ to S and $\left(\texttt{output}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\right)$ to R.

Remark 2.6. We likewise give the adversary slightly more power than does [KOS15, Fig. 6], in that we let the corrupt receiver choose $(\mathbf{v}_{i,x_i})_{i=0}^{l-1}$. This concession appears necessary; indeed—aside from its other issues—the simulator [KOS15, Fig. 8] programs $H(i \parallel \mathbf{q}_i + x_i \cdot \Delta) := \mathbf{v}_{i,x_i}$ only *after* receiving $\mathbf{t}_i$ from $\mathcal{A}$. $\mathcal{A}$ can easily arrange to make this query before this programming step occurs, thereby breaking the simulation. We note that issue, as well as further discussion, appears in Masny and Rindal's *Endemic OT* [MR19, § 5.1].

For self-containedness, we finally recall the full protocol for $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, exactly as in [KOS15, Fig. 7].

PROTOCOL 2.7 ($\Pi_{\mathsf{ROT}}^{\kappa,l}$—random OT protocol [KOS15, Fig. 7])**.**
The parameters κ and l, and players S and R, are fixed. R has input bits $(x_0, \ldots, x_{l-1})$.

- The parties write $l' := l + \kappa + s$. S samples $\Delta \leftarrow \mathbb{F}_2^\kappa$, and sends $(\texttt{intialize}, \Delta)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$.
- R samples random bits $x_i \leftarrow \mathbb{F}_2$, for $i \in \{l, \ldots, l'-1\}$. For each $i \in \{0, \ldots, l'-1\}$, R constructs the monochromatic vector $\mathbf{x}_i := x_i \cdot (1, \ldots, 1)$. R sends $\left(\texttt{input}, (\mathbf{x}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$. S and R receive $\left(\texttt{output}, (\mathbf{q}_i)_{i=0}^{l'-1}\right)$ and $\left(\texttt{output}, (\mathbf{t}_i)_{i=0}^{l'-1}\right)$, respectively, from $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$.
- For each $i \in \{0, \ldots, l'-1\}$, both parties submit $(\texttt{random}, i)$ to $\mathcal{F}_{\mathsf{Rand}}^{\kappa}$, and receive $(\texttt{random}, i, \chi_i)$. R sends S $x := \sum_{i=0}^{l'-1} \chi_i \cdot x_i$ and $t := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$. S sets $q := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$, and checks $q \stackrel{?}{=} t + x \cdot \Delta$.

- For each $i \in \{0, \ldots, l-1\}$, R sets $\mathbf{v}_{i,x_i} := H(i \,\|\, \mathbf{t}_i)$, and outputs $(\mathbf{v}_{i,x_i})_{i=0}^{l-1}$. For each $i \in \{0, \ldots, l-1\}$, S sets $\mathbf{v}_{i,0} := H(i \,\|\, \mathbf{q}_i)$ and $\mathbf{v}_{i,1} := H(i \,\|\, \mathbf{q}_i + \Delta)$, and outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$.

3 Security proof

We now prove the security of Protocol 2.7.

Theorem 3.1. *In the $\mathcal{F}_{\mathsf{RO}}, \mathcal{F}_{\mathsf{Rand}}^{\kappa}, \mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ hybrid model, Protocol 2.7 securely computes Functionality 2.5.*

Proof. We define an appropriate simulator $\mathcal{S}$.

Corrupt sender. We first handle the case in which S is corrupt. Our treatment of this case is similar to that of [KOS15, Thm. 1]. Given a real-world adversary $\mathcal{A}$ corrupting S, $\mathcal{S}$ operates in the following way.

1. $\mathcal{S}$ intercepts $\mathcal{A}$'s messages $(\mathtt{initialize}, \Delta)$ and $\left(\mathtt{input}, (\mathbf{q}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$. For each $i \in \{0, \ldots, l-1\}$, $\mathcal{S}$ computes $\mathbf{v}_{i,0} := H(i \,\|\, \mathbf{q}_i)$ and $\mathbf{v}_{i,1} := H(i \,\|\, \mathbf{q}_i + \Delta)$. $\mathcal{S}$ submits $\left(\mathtt{input}, (\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$.
2. $\mathcal{S}$ receives $\left(\mathtt{output}, (\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}\right)$ from $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, and simulates $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ sending $\left(\mathtt{output}, (\mathbf{q}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{A}$.
3. For each $i \in \{0, \ldots, l'-1\}$, $\mathcal{S}$ intercepts $\mathcal{A}$'s message $(\mathtt{random}, i)$ intended for $\mathcal{F}_{\mathsf{Rand}}^{\kappa}$, samples $\chi_i \leftarrow \mathbb{F}_2^{\kappa}$ randomly, and simulates $\mathcal{F}_{\mathsf{Rand}}^{\kappa}$ sending $\mathcal{A}$ $(\mathtt{random}, i, \chi_i)$. $\mathcal{S}$ samples $x \leftarrow \mathbb{F}_2^{\kappa}$ randomly, computes $q := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$, and sets $t := q + x \cdot \Delta$. $\mathcal{S}$ simulates R sending $\mathcal{A}$ t and x.

The perfection of this simulation is self-evident, except perhaps for the distribution of x. For self-containedness, we present a full proof of the relevant lemma, whose proof is omitted from [KOS15, Lem. 2].

Lemma 3.2. *Given a random $\kappa \times (\kappa + s)$ matrix X over $\mathbb{F}_2$, where $s \geq 0$, $\Pr[\mathrm{rank}(X) = \kappa] \geq 1 - 2^{-s}$.*

Proof. For each fixed value $s \geq 0$, the probability that the random matrix X's κ rows are independent is equal to the probability that each of its successive rows resides outside of the linear subspace spanned by its previous rows. This probability is given by product expression below, which we manipulate as follows:

$$\begin{aligned}(1 - 2^{-s-1}) \cdot \cdots \cdot (1 - 2^{-s-\kappa}) &\geq 1 - \left(2^{-s-1} + \cdots + 2^{-s-\kappa}\right) \\ &= 1 - 2^{-s} \cdot \left(2^{-1} + \cdots + 2^{-\kappa}\right) \\ &\geq 1 - 2^{-s}.\end{aligned}$$

The first inequality follows from a simple union bound, which we now explain. The expression $1 - \prod_{i=0}^{\kappa-1}(1 - 2^{-s-1-i})$ gives the probability that a certain product of Bernoulli distributions resides *away* from the origin in $\{0,1\}^{\kappa}$. By the union bound, this probability is bounded from above by the sum of faces $\sum_{i=0}^{\kappa-1} 2^{-s-1-i}$. □

The second summand of the quantity $x = \sum_{i=0}^{l-1} \chi_i \cdot x_i + \sum_{i=l}^{l'-1} \chi_i \cdot x_i$ computed by the receiver can be viewed as the image of $(x_i)_{i=l}^{l'-1} \in \mathbb{F}_2^{\kappa+s}$ under the linear map $\mathbb{F}_2^{\kappa+s} \to \mathbb{F}_2^{\kappa}$ defined by the matrix:

$$\begin{bmatrix} | & & | \\ \chi_l & \cdots & \chi_{l'-1} \\ | & & | \end{bmatrix}.$$

The lemma implies that, with probability at least $1 - 2^s$ over the choice of $(\chi_i)_{i=l}^{l'-1}$, the map induced by this matrix is surjective; it follows that, in the real-world distribution, with overwhelming probability, the image of the uniformly random point $(x_i)_{i=l}^{l'-1} \in \mathbb{F}_2^{\kappa+s}$ under this matrix is itself uniform in $\mathbb{F}_2^{\kappa}$, and so perfectly hides the first term $\sum_{i=0}^{l-1} \chi_i \cdot x_i$. This completes the treatment of the corrupt sender.

Corrupt receiver. We now handle the case in which the receiver R is corrupt. We begin by formulating a numerical metric—called the *modesty*, a quantity $m \in \{1, \ldots, \kappa\}$—describing the extent to which $\mathcal{A}$'s initial matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$ is monochromatic. We identify vectors in $\mathbb{F}_2^\kappa$ with subsets of $\{0, \ldots, \kappa-1\}$ in what follows, in the obvious way. We emphasize that the definition below acts *only* on the l-row submatrix $(\mathbf{x}_i)_{i=0}^{l-1}$.

Definition 3.3. The *modesty* of $(\mathbf{x}_i)_{i=0}^{l-1}$ is the *largest* $m \in \{1, \ldots, \kappa\}$ for which $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) \stackrel{?}{=} \mathsf{false}$:

1: **function** $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$
2: set $\mathbf{d} := \varnothing$, and initialize $(x_i)_{i=0}^{l-1}$ arbitrarily, with each index $i \in \{0, \ldots, l-1\}$ marked white.
3: **for** l repetitions **do**
4: **for** $i \in \{0, \ldots, l-1\}$ **do**
5: **if** i is white **and** either $|\mathbf{x}_i \setminus \mathbf{d}| < m$ **or** $|\overline{\mathbf{x}_i} \setminus \mathbf{d}| < m$ **then**
6: overwrite $x_i \in \{0, 1\}$ so that, for $\mathbf{e}_i := \mathbf{x}_i + x_i \cdot (1, \ldots, 1)$, we have that $|\mathbf{e}_i \setminus \mathbf{d}| < m$.
7: **if** $\mathbf{e}_i \subset \mathbf{d}$ **then** mark the index $i \in \{0, \ldots, l-1\}$ grey.
8: **else** update $\mathbf{d} \cup= \mathbf{e}_i$ and mark the index $i \in \{0, \ldots, l-1\}$ black.
9: **if** $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| < m$ **then return** true.
10: **break** the inner loop 4.
11: **return** false.

Informally, Definition 3.3 captures an attempt by an adversary to iteratively "reach" some (i.e., any) off-vector $\overline{\mathbf{e}_i}$, by progressively incorporating the bit-positions indicated by various on-vectors $\mathbf{e}_i$. Indeed, $\mathbf{d} \subset \{0, \ldots, \kappa-1\}$ represents the positions which an adversary—whose "tolerance" for new stretches of bits is controlled by m—can feasibly reach. The routine returns true if the adversary succeeds in reaching an off-vector. We note that for each input m, the vector $\mathbf{d}$ and the assignment $(x_i)_{i=0}^{l-1}$ necessarily eventually stabilize, in *at most* l iterations of the outer loop 3; indeed, in each iteration, the algorithm either marks exactly one vector grey or black, or else stabilizes (possibly by returning true).

The following lemma captures the key correctness property of Definition 3.3.

Lemma 3.4. *For each* $(\mathbf{x}_i)_{i=0}^{l'-1}$*, and each* $m \in \{1, \ldots, \kappa\}$ *for which* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$*, if* $\mathbf{d}$ *is the vector assembled during the course of* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$*, then, for each* $i \in \{0, \ldots, l-1\}$*,* $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq m$*.*

Proof. We introduce notation. The vectors $(\mathbf{e}_i)_{i=0}^{l-1}$ in the lemma's hypothesis, of course, are given meaning by means of the bit assignment $(x_i)_{i=0}^{l-1}$ assembled internally throughout the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. We suppose for contradiction that, though $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$, the index $i^* \in \{0, \ldots, l-1\}$, say, is such that $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$ holds. We first note that i^* is necessarily either grey or black. Indeed, if i^* were white, then we would deduce the failure of i^* to fulfill the algorithm's condition 5 on its last iteration, and, in particular, the inequalities $|\mathbf{x}_{i^*} \setminus \mathbf{d}| \geq m$ and $|\overline{\mathbf{x}_{i^*}} \setminus \mathbf{d}| \geq m$. These would contradict our assumption that $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$. We conclude that i^* is grey or black, and that $\mathbf{e}_{i^*} \subset \mathbf{d}$.

We write $i' \in \{0, \ldots, l-1\}$ for the last index marked non-white by the algorithm. Since $\mathbf{e}_{i^*} \subset \mathbf{d}$, $\overline{\mathbf{d}} = \overline{\mathbf{e}_{i^*}} \cap \overline{\mathbf{d}}$. We conclude that $\overline{\mathbf{e}_{i'}} \cap \overline{\mathbf{d}} = \overline{\mathbf{e}_{i'}} \cap \overline{\mathbf{e}_{i^*}} \cap \overline{\mathbf{d}} \subset \overline{\mathbf{e}_{i^*}} \cap \overline{\mathbf{d}}$, so that $\overline{\mathbf{e}_{i'}} \setminus \mathbf{d} \subset \overline{\mathbf{e}}_{i*} \setminus \mathbf{d}$. Using our hypothesis whereby $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$, we see finally that $|\overline{\mathbf{e}_{i'}} \setminus \mathbf{d}| < m$. This implies that the escape condition 9 was fulfilled just after $\mathbf{e}_{i'}$ was marked non-white, contradicting our assumption that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$. □

Informally, Lemma 3.4 states that the *only* way for $\mathbf{d}$ to reach within distance m of some vector $\overline{\mathbf{e}_{i^*}}$, in Definition 3.3, is for both $|\mathbf{x}_{i^*} \setminus \mathbf{d}| < m$ and $|\overline{\mathbf{x}_{i^*}} \setminus \mathbf{d}| < m$ to become true simultaneously. In particular, it *can't* happen that the vector $\mathbf{d}$, upon being made to include the contents of some vector $\mathbf{e}_{i'}$ for which $|\mathbf{e}_{i'} \setminus \mathbf{d}| < m$ and yet $|\overline{\mathbf{e}_{i'}} \setminus \mathbf{d}| \geq m$, simultaneously comes to fulfill the inequality $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$.

Were Lemma 3.4 unproven, or even false, we could apparently compensate, at least for the purposes of our proof below, by appending to the routine of Definition 3.3 an artificial "check", which—before returning false—tested the inequalities $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \stackrel{?}{<} m$ for each $i \in \{0, \ldots, l-1\}$ (returning true upon detecting a fulfillment). This "remedy", of course, would leave unanswered whether this check was effectual (i.e., whether it was actually capable of inducing the algorithm to return true).

The following two lemmas are, as it turns out, not necessary to establish the proof of our main result. Nonetheless, they establish the "well-behavedness" of Definition 3.3.

Lemma 3.5. *For each input matrix* $(\mathbf{x}_i)_{i=0}^{l'-1}$ *and each* $m \in \{1, \dots, \kappa\}$*, the boolean return value of* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$*, and, if* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) \stackrel{?}{=} \mathsf{false}$*, the vector* $\mathbf{d} \subset \{0, \dots, \kappa-1\}$ *assembled throughout the course of the routine, depend only on the input rows* $(\mathbf{x}_i)_{i=0}^{l-1}$*, and not on the ordering of these rows.*

Proof. We first argue that the boolean return value of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ is independent of the ordering of the input rows $(\mathbf{x}_i)_{i=0}^{l-1}$. We suppose that $m \in \{1, \dots, \kappa\}$ is such that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, and that, for some permutation $(\mathbf{x}'_i)_{i=0}^{l-1}$ of $(\mathbf{x}_i)_{i=0}^{l-1}$, $\textsc{Modest}\Big((\mathbf{x}'_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$. We write $(x_i)_{i=0}^{l-1}$ and $\mathbf{d}$ for the internal values ultimately assembled during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, and $(x'_i)_{i=0}^{l-1}$ and $\mathbf{d}'$ for the corresponding values assembled during $\textsc{Modest}\Big((\mathbf{x}'_i)_{i=0}^{l-1}, m\Big)$. Throughout, we give meaning to the symbols $(\mathbf{e}_i)_{i=0}^{l-1}$ by means of the ordering $(\mathbf{x}_i)_{i=0}^{l-1}$ and the bit assignment $(x_i)_{i=0}^{l-1}$; we moreover write $(c'_i)_{i=0}^{l-1}$ for the colors respectively assigned to the vectors $(\mathbf{x}_i)_{i=0}^{l-1}$ during $\textsc{Modest}\Big((\mathbf{x}'_i)_{i=0}^{l-1}, m\Big)$. Finally, we write $(i_0, \dots, i_{r-1})$ for the *ordered* sequence of indices $i \in \{0, \dots, l-1\}$ marked black during the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, and $\varnothing = \mathbf{d}_{i_0} \subset \cdots \subset \mathbf{d}_{i_{r-1}}$ for the sequence of values taken by $\mathbf{d}$ *immediately before* the respective updates $\mathbf{d} \cup\!= \mathbf{e}_{i_j}$, for $j \in \{0, \dots, r-1\}$.

Since $\textsc{Modest}\Big((\mathbf{x}'_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$, Lemma 3.4 applies to that execution. Applying that lemma, we see that, if $x'_{i_{r-1}} \neq x_{i_{r-1}}$, then $\big|\mathbf{e}_{i_{r-1}} \setminus \mathbf{d}'\big| \geq m$; on the other hand, if $x'_{i_{r-1}} = x_{i_{r-1}}$, then $\big|\overline{\mathbf{e}_{i_{r-1}}} \setminus \mathbf{d}'\big| \geq m$. If $\mathbf{d} \subset \mathbf{d}'$ held, then these two possibilities would, respectively, contradict the inequalities $\mathbf{e}_{i_{r-1}} \subset \mathbf{d}$ and $\big|\overline{\mathbf{e}_{i_{r-1}}} \setminus \mathbf{d}\big| < m$, themselves immediate consequences of our assumption whereby $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$. We thus see that, regardless of $x'_{i_{r-1}}$, $\mathbf{d} \not\subset \mathbf{d}'$.

We select an element $\alpha_0 \in \mathbf{d} \setminus \mathbf{d}'$, and write $j_0 \in \{i_0, \dots, i_{r-1}\}$ for the index for which the update $\mathbf{d} \cup\!= \mathbf{e}_{j_0}$ first caused the inclusion $\alpha_0 \in \mathbf{d}$ to become true. Immediately before this update was performed, $|\mathbf{e}_{j_0} \setminus \mathbf{d}_{j_0}| < m$ held. On other hand, we claim that $|\mathbf{e}_{j_0} \setminus \mathbf{d}'| \geq m$. If $c'_{j_0} \stackrel{?}{=} \mathsf{white}$, then this inequality certainly holds (since, in this case, both $|\mathbf{x}_{j_0} \setminus \mathbf{d}'| \geq m$ and $|\overline{\mathbf{x}_{j_0}} \setminus \mathbf{d}'| \geq m$ hold). Assuming that $c'_{j_0} \neq \mathsf{white}$, the equality $x'_{j_0} \stackrel{?}{=} x_{j_0}$ would imply that $\mathbf{e}_{j_0} \subset \mathbf{d}'$, contradicting $\alpha_0 \notin \mathbf{d}'$. We conclude that $x'_{j_0} \neq x_{j_0}$. In this setting, Lemma 3.4 implies that $|\mathbf{e}_{j_0} \setminus \mathbf{d}'| \geq m$, as desired. Since $|\mathbf{e}_{j_0} \setminus \mathbf{d}_{j_0}| < m$ and $|\mathbf{e}_{j_0} \setminus \mathbf{d}'| \geq m$ both hold, we conclude that $\mathbf{d}_{j_0} \not\subset \mathbf{d}'$.

We iteratively repeat this process as follows. We select, as before, an element $\alpha_1 \in \mathbf{d}_{j_0} \setminus \mathbf{d}'$, and write $j_1 \in \{i_0, \dots, i_{r-1}\}$ for the index for which the update $\mathbf{d} \cup\!= \mathbf{e}_{j_1}$ first caused $\alpha_1 \in \mathbf{d}$ to become true. We note that $j_1 < j_0$, since, by definition of $\mathbf{d}_{j_0}$, $\alpha_1 \in \mathbf{d}$ held *before* the update $\mathbf{d} \cup\!= \mathbf{e}_{j_0}$ was applied. As above, we see that, if $c'_{j_1} \stackrel{?}{=} \mathsf{white}$, then $|\mathbf{e}_{j_1} \setminus \mathbf{d}'| \geq m$ necessarily holds; otherwise, since $x'_{j_1} \stackrel{?}{=} x_{j_1}$ would contradict $\alpha_1 \notin \mathbf{d}'$, we conclude that $x'_{j_1} \neq x_{j_1}$, and that, again by Lemma 3.4, $|\mathbf{e}_{j_1} \setminus \mathbf{d}'| \geq m$ holds in any case. This conclusion, in light of the inequality $|\mathbf{e}_{j_1} \setminus \mathbf{d}_{j_1}| < m$, implies that $\mathbf{d}_{j_1} \not\subset \mathbf{d}'$.

Iteratively proceeding in this way, we obtain a descending sequence of indices $j_0 > \cdots > j_{s-1}$, say, which must eventually terminate, so that $j_{s-1} = i_0$. Since $\mathbf{d}_{i_0} = \varnothing$, we conclude that $\varnothing \not\subset \mathbf{d}'$, an absurdity. It follows that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$.

In fact, the second half of the proof just given shows that the assumptions $\textsc{Modest}\Big((\mathbf{x}'_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$ and $\mathbf{d} \not\subset \mathbf{d}'$ alone suffice to derive a contradiction. Applying that argument symmetrically to $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, we conclude that $\mathbf{d} = \mathbf{d}'$. This completes the proof. □

Remark 3.6. The bit assignment $(x_i)_{i=0}^{l-1}$ ultimately constructed during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ *can actually vary* as the ordering of the rows $(\mathbf{x}_i)_{i=0}^{l-1}$ varies, even when $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$. If $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, then neither $\mathbf{d}$ nor $(x_i)_{i=0}^{l-1}$ are independent of the ordering of the rows $(\mathbf{x}_i)_{i=0}^{l-1}$.

Lemma 3.7. *For each input matrix* $(\mathbf{x}_i)_{i=0}^{l'-1}$*, the function* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ *is monotone on its domain* $\{1,\ldots,\kappa\}$*; that is, for each pair of arguments* $m \leq m'$*,* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) \implies \textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$.

Proof. Exploiting Lemma 3.5, we obtain a simple proof. Indeed, before running $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$, we may freely assume that the rows $(\mathbf{x}_i)_{i=0}^{l-1}$ are sorted in precisely the order in which they are marked non-white during the execution of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ (with white rows deferred). Since the condition 5 only becomes weaker when m is replaced by m', we see that the *same* rows marked non-white during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ will likewise be marked non-white during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$, and in the same order. Finally, the condition 9 too becomes weaker, and will be fulfilled if it was fulfilled during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. □

Remark 3.8. It is interesting that the easiest proof of Lemma 3.7 seems to be that—just given—which proceeds via the aid of the more-complicated Lemma 3.5. Though a direct proof would be interesting, and is probably possible, we have restricted ourselves, for the sake of brevity, to the approach already taken above.

We now define our simulator. Given a real-world adversary $\mathcal{A}$ corrupting the receiver R, $\mathcal{S}$ operates as follows.

1. $\mathcal{S}$ simulates the existence of $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, including S's role. $\mathcal{S}$ begins by sampling $\Delta \leftarrow \mathbb{F}_2^\kappa$, as S would.
2. Upon intercepting $\mathcal{A}$'s message $\Big(\mathtt{input}, (\mathbf{x}_i)_{i=0}^{l'-1}, (\mathbf{t}_i)_{i=0}^{l'-1}\Big)$ intended for $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, $\mathcal{S}$ sets $\mathbf{q}_i := \mathbf{t}_i + \mathbf{x}_i * \Delta$ for each $i \in \{0,\ldots,l'-1\}$. Running Definition 3.3, $\mathcal{S}$ extracts the assignment $(x_i)_{i=0}^{l-1}$ constructed during the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, where $m \in \{1,\ldots,\kappa\}$ is the modesty of $(\mathbf{x}_i)_{i=0}^{l-1}$. $\mathcal{S}$ sets $\mathbf{v}_{i,x_i} := H(i \parallel \mathbf{q}_i + x_i \cdot \Delta)$ for each $i \in \{0,\ldots,l-1\}$. $\mathcal{S}$ submits $\Big(\mathtt{input}, (x_i)_{i=0}^{l-1}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\Big)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$.
3. $\mathcal{S}$ receives $\Big(\mathtt{output}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\Big)$ from $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, and simulates $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ returning $\Big(\mathtt{output}, (\mathbf{t}_i)_{i=0}^{l'-1}\Big)$ to $\mathcal{A}$.
4. For each $i \in \{0,\ldots,l'-1\}$, $\mathcal{S}$ intercepts $\mathcal{A}$'s message $(\mathtt{random}, i)$ intended for $\mathcal{F}_{\mathsf{Rand}}^{\kappa}$, samples $\chi_i \leftarrow \mathbb{F}_2^\kappa$ randomly, and simulates $\mathcal{F}_{\mathsf{Rand}}^{\kappa}$ sending $\mathcal{A}$ $(\mathtt{rand}, i, \chi_i)$. Upon receiving x and t from $\mathcal{A}$, $\mathcal{S}$ independently computes $q := \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$, and runs the correlation check $q \stackrel{?}{=} t + x \cdot \Delta$. If the check fails, $\mathcal{S}$ submits $(\mathtt{abort})$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$; otherwise, $\mathcal{S}$ proceeds, and $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ releases the output to the ideal honest party S.

We now claim that the resulting real and ideal distributions are computationally indistinguishable. More precisely, these distributions are statistically indistinguishable to any computationally unbounded distinguisher which makes only polynomially many queries to the random oracle. We fix a distinguisher D attacking these distributions.

Following [KOS15], for each $i \in \{0,\ldots,l-1\}$, we write $\mathbf{e}_i := \mathbf{x}_i + x_i \cdot (1,\ldots,1)$, where x_i is as extracted by $\mathcal{S}$ above. We observe that the strings $\mathbf{q}_i + x_i \cdot \Delta$ and $\mathbf{q}_i + \overline{x_i} \cdot \Delta$ respectively equal $\mathbf{t}_i + \mathbf{e}_i * \Delta$ and $\mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta$; the values $\mathbf{t}_i$ and $\mathbf{e}_i$ are known to the distinguisher, while Δ is not. If the correlation check fails, then the real and ideal distributions are identical. If the correlation check succeeds, the simulation is perfect *except* for the fact that, for each $i \in \{0,\ldots,l-1\}$, in the real world, the relation $\mathbf{v}_{i,\overline{x_i}} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$ holds, whereas, in the ideal world, $\mathbf{v}_{i,\overline{x_i}}$ is independently random.

For notational purposes, given $x \in \mathbb{F}_{2^\kappa}$, we introduce the map $F_x : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\kappa$ defined by:

$$F_x : \Delta \mapsto \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{t}_i + \mathbf{x}_i * \Delta) + x \cdot \Delta + t.$$

We note, in light of the equalities $\mathbf{q}_i = \mathbf{t}_i + \mathbf{x}_i * \Delta$, that this map exactly reflects the correlation check run by the sender on its (secret) correlation vector Δ (i.e., the check passes if and only if $F_x(\Delta) \stackrel{?}{=} 0$, where x is as is sent by $\mathcal{A}$). We view all quantities above as fixed constants—known to the distinguisher—*except* for the unknown vector Δ.

Clearly, $F_x : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\kappa$ is an $\mathbb{F}_2$-affine linear map. We argue that we may assume once and for all that $\mathcal{A}$ submits an "honest" value $t = \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$. Indeed, our below arguments depend only on the dimension of the affine subspace $\{\Delta \in \mathbb{F}_2^\kappa \mid F_x(\Delta) = 0\}$, and not on its contents; $\mathcal{A}$'s use of a value $t \neq \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$ has merely the effect of replacing this subspace *either* with an affine-linear subspace of $\mathbb{F}_2^\kappa$ of identical dimension *or* with the empty affine subspace (i.e., depending on whether $t + \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$ resides within the image of $\Delta \mapsto \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{x}_i * \Delta) + x \cdot \Delta$ or not). If the subspace is empty, then the correlation check is guaranteed to fail, and the simulation becomes trivially secure. We thus introduce a simplified variant of F_x, in which the affine constants are dropped:

$$F'_x : \Delta \mapsto \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{x}_i * \Delta) + x \cdot \Delta.$$

In particular, we refer to $\mathrm{rank}(F'_x)$ and $\ker(F'_x)$ throughout. We treat only F'_x throughout the remainder of the proof.

We denote by r the *minimal* rank achieved across all maps $\{F'_x\}_{x \in \mathbb{F}_{2^\kappa}}$, so that:

$$r := \min_{x \in \mathbb{F}_{2^\kappa}} \mathrm{rank}(F'_x). \tag{1}$$

In what follows, we view r as a random variable, a function of the randomly sampled coefficients $(\chi_i)_{i=0}^{l'-1}$.

We now pause to sketch the details of our proof. We consider the protocol in steps, corresponding, respectively, to $\mathcal{A}$'s choice of $(\mathbf{x}_i)_{i=0}^{l'-1}$ (and hence of modesty), to the random sampling of $(\chi_i)_{i=0}^{l'-1}$ (which causes the minimal rank r to be defined), to whether $\mathcal{A}$ passes the correlation check, and, finally, to whether the distinguisher succeeds. The resulting structure is depicted in the figure below, which should be understood as a probability tree, in which each edge represents a conditional probability.

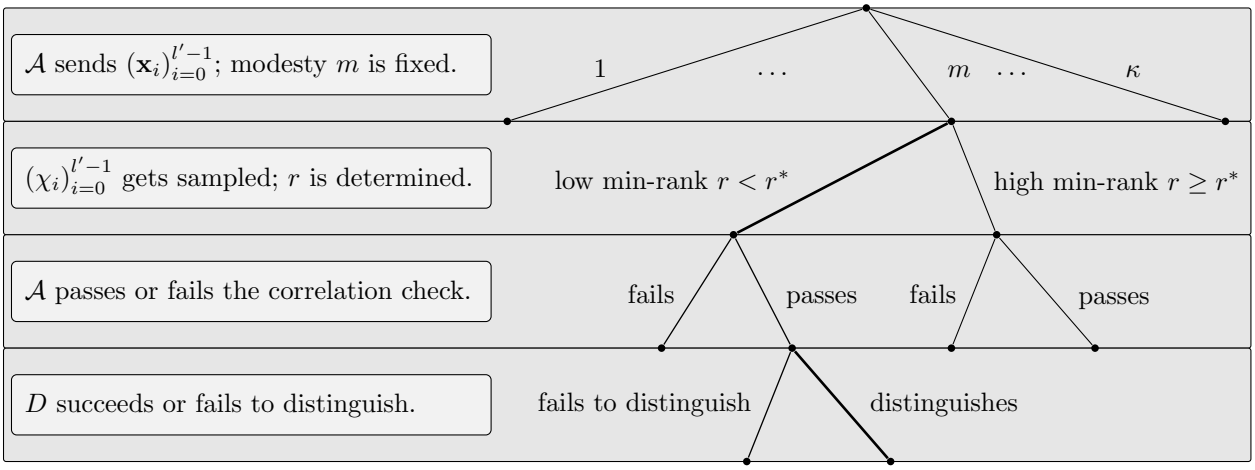

Figure 1: A depiction of the case structure considered by our proof.

We analyze an instance of the above tree for each execution (i.e., for each modesty m). We may immediately ignore those executions in which $\mathcal{A}$ fails the correlation check, since the real and ideal distributions are identical in each such execution. Beyond this, we set a *rank cutoff* r^*. If we choose for our cutoff $r^* = r^*(\kappa)$ a superlogarithmic function of κ, then we may likewise ignore the subtree in which $r \geq r^*$ *and* $\mathcal{A}$ passes the correlation check, since $\mathcal{A}$'s chance of passing the correlation check is $2^{-\mathrm{rank}(F'_x)}$, which—in light of our choice of r^*—is negligible whenever $r \geq r^*$, since $\mathrm{rank}(F'_x) \geq r$. (We discuss our specific choice of r^* below.) We are thus left with one relevant path through the tree. Our proof hinges on analyzing the two edges bolded in the diagram above. Roughly, we show that as $\mathcal{A}$'s matrix's modesty varies, *either* the lowermost bolded edge *or* the uppermost bolded edge (or both) must be negligible in κ (these cases happen when $\mathcal{A}$'s matrix *is* and *isn't* modest, respectively). This suffices to demonstrate the result.

We first study the probability that the minimal rank r of (1) is low. Instead of precisely describing the distribution of r as a random variable, we instead fix a cutoff $r^* \in \{1, \ldots, \kappa\}$, and upper-bound the probability that $r < r^*$. Our main result is as follows.

Proposition 3.9. *For each arbitrary rank cutoff $r^* \in \{1, \ldots, \kappa\}$, and each initial matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$, of modesty $m \in \{1, \ldots, \kappa\}$, say, the probability—over the choice of $(\chi_i)_{i=0}^{l'-1}$—that $r < r^*$ is at most $2^{\kappa \cdot (r^*+1) - \kappa \cdot \frac{\kappa - r^* + 1}{m}}$.*

Proof. We begin by *further* simplifying F'_x in certain ways. We first note that adding some fixed constant $x^* \in \mathbb{F}_q$ to each index x in the expression (1) above merely permutes the resulting elements $\{F'_x\}_{x \in \mathbb{F}_{2^\kappa}}$, and has no effect on r. We run the procedure $\textsc{Modest}(m+1)$ on the matrix $(\mathbf{x}_i)_{i=0}^{l-1}$, and write $(x'_i)_{i=0}^{l-1}$ for the resulting vector of assignments; we moreover assign the further components $(x'_i)_{i=l}^{l'-1}$ *arbitrarily*. We write $(\mathbf{e}'_i)_{i=0}^{l'-1}$ for the list of vectors $\mathbf{e}'_i := \mathbf{x}_i + x'_i \cdot (1, \ldots, 1)$, for $i \in \{0, \ldots, l'-1\}$. After adding $x^* := \sum_{i=0}^{l'-1} x'_i \cdot \chi_i$ to each x in (1), we may freely replace each $\mathbf{x}_i$ with $\mathbf{e}'_i$ in F'_x's definition. We thus rewrite F'_x as follows:

$$F''_x : \Delta \mapsto \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{e}'_i * \Delta) + x \cdot \Delta.$$

We finally argue that the random variable r of (1)—viewed, again, as a function of the random coefficients $(\chi_i)_{i=0}^{l'-1}$—remains identical if we replace the matrix $(\mathbf{e}'_i)_{i=0}^{l'-1}$ with its reduced row-echelon form over $\mathbb{F}_2$. Indeed, each map F''_x may be decomposed into the $\mathbb{F}_2$-linear map $\Delta \mapsto (\mathbf{e}'_i * \Delta)_{i=0}^{l'-1}$ from $\mathbb{F}_2^\kappa \to \mathbb{F}_{2^\kappa}^{l'}$, on the one hand, followed by the application of the random $\mathbb{F}_{2^\kappa}$-hyperplane given by $(\chi_i)_{i=0}^{l'-1}$, on the other (and finally by the addition of $x \cdot \Delta$). Row-reducing $(\mathbf{e}'_i)_{i=0}^{l'-1}$ amounts to interposing between these first two maps a further $l' \times l'$ invertible matrix over $\mathbb{F}_{2^\kappa}$. Up to a fresh uniform resampling of the hyperplane coefficients $(\chi_i)_{i=0}^{l'-1}$, this matrix multiplication has no effect.

Lemma 3.10. *The reduced row-echelon form of the binary matrix $(\mathbf{e}'_i)_{i=0}^{l'-1}$ has at least $\frac{\kappa}{m} - 1$ pivots.*

Proof. As each matrix's number of pivots depends only on its rank, it suffices to prove the lemma after arbitrarily permuting $(\mathbf{e}'_i)_{i=0}^{l'-1}$'s rows and columns. We thus freely sort the rows $(\mathbf{e}'_i)_{i=0}^{l'-1}$ in the order in which they are marked black by the procedure $\textsc{Modest}(m+1)$ (deferring all white and grey rows, as well as rows indexed $i \in \{l, \ldots, l'-1\}$). Moreover, we apply the following modification to the Gaussian elimination algorithm. By construction, each row marked black introduces a 1 to some column which thus far has lacked one. Upon each such row's treatment by the algorithm, after possibly transposing the column being considered for a pivot with some column strictly to its right, we may assume that this 1 resides precisely at the column being considered for a pivot, and thus becomes a pivot. This transposition preserves the invariant whereby each *further* black row introduces a 1 at some new column. (Indeed, each row which admits some cell containing the highest 1 in its column will continue to do so, even after this transposition, albeit possibly at a new column-index.) Likewise, using the new pivot row to clear the pivot column also preserves this invariant. (Indeed, for each black row strictly beneath the one being treated, which, by the invariant, necessarily features some cell containing its column's highest 1, the clearing process will have no effect on that particular column, since the row containing the newly minted pivot necessarily features a 0 at that column.) We thus conclude that there are at least as many pivots as there are black rows.

Finally, we note that $\textsc{Modest}(m+1)$ must mark at least $\frac{\kappa - m}{m} = \frac{\kappa}{m} - 1$ rows black. Indeed, by our choice of m, $\textsc{Modest}(m+1) = \mathsf{true}$, so that the vector $\mathbf{d}'$ (say) assembled during the course of $\textsc{Modest}(m+1)$ simultaneously satisfies $\mathbf{e}'_{i^*} \subset \mathbf{d}'$ and $\left|\overline{\mathbf{e}'_{i^*}} \setminus \mathbf{d}'\right| \leq m$, where $i^* \in \{0, \ldots, l-1\}$, say, is the last row marked grey or black by the algorithm (for which the condition 9 was fulfilled necessarily fulfilled). The first inclusion directly implies that $\overline{\mathbf{d}'} \subset \overline{\mathbf{e}'_{i^*}} \setminus \mathbf{d}'$; applying the second inequality, we see that $\left|\overline{\mathbf{d}'}\right| \leq m$, so that $|\mathbf{d}'| \geq \kappa - m$. Since each index $i \in \{0, \ldots, l-1\}$ marked black within $\textsc{Modest}(m+1)$ can only increase $|\mathbf{d}'|$ by at most m, we conclude the result. □

In light of the above, we freely assume throughout what follows that $(\mathbf{e}'_i)_{i=0}^{l'-1}$ is in fact row-reduced. We moreover write $\widehat{\kappa}$ for the number of pivots in this matrix.

We continue our study of the maps F''_x. We note that each such map can be written using the following matrix expression:

$$F''_x : \Delta \mapsto \left(\left[\chi_0 \cdots \right] + \cdots + \left[\chi_{l'-1} \right] + \left[x \cdots \right] \right) \cdot \left[\Delta \right],$$

where the field elements x and $(\chi_i)_{i=0}^{l'-1}$ are viewed as $\mathbb{F}_2$-linear operators on $\mathbb{F}_2^\kappa$, and hence represented as $\kappa \times \kappa$ $\mathbb{F}_2$-*matrices*, and the shaded boxes indicate that certain columns have been "struck out". Indeed, we keep or strike columns of the matrices of $(\chi_i)_{i=0}^{l'-1}$ according to the (row-reduced) data $(\mathbf{e}'_i)_{i=0}^{l'-1}$; specifically, if $\mathbf{e}'_{i,j} = 1$, we keep the j^{th} column of χ_i's matrix intact, and otherwise replace it with a column of 0s.

In light of our assumption that $(\mathbf{e}'_i)_{i=0}^{l'-1}$ is row-reduced, we see that each pivot in the matrix $(\mathbf{e}'_i)_{i=0}^{l'-1}$ adds an *independent random column* to the matrix expression above (i.e., to its left-hand sum, *excluding* x). We argue that we may, conservatively, consider the pivot columns *alone* in our study of (1). Indeed, replacing each non-pivot column with a column of 0s—in *all* matrices within the expression above, including that of x—can only *decrease* the rank of F''_x; we shall lower-bound this rank regardless.

We're thus left to consider the following modified expression for F''_x:

$$F'''_x : \Delta \mapsto \left(\left[X \cdots \right] + \left[x \cdots \right] \right) \cdot \left[\Delta \right],$$

where the first matrix, say X, contains $\widehat{\kappa}$ *independently* random columns, with its further columns identically 0, and where the second matrix is merely the field-multiplication matrix of x, with the same set of $\widehat{\kappa}$ columns kept and the rest struck out. We emphasize that, necessarily, $\text{rank}(F'''_x) \leq \text{rank}(F''_x)$.

We now consider the probability, over the uniformly random submatrix X, that $\min_{x \in \mathbb{F}_{2^\kappa}} \text{rank}(F'''_x) < r^*$. We make use of a counting argument in $\mathbb{F}_2^{\kappa \times \kappa}$; more precisely, the argument takes place in $\mathbb{F}_2^{\kappa \times \widehat{\kappa}}$. Slightly abusing notation, we identify field elements $x \in \mathbb{F}_{2^\kappa}$ with (appropriately stricken) matrices in $\mathbb{F}_2^{\kappa \times \widehat{\kappa}}$. We note that there are exactly 2^κ distinct field elements $x \in \mathbb{F}_{2^\kappa}$, and hence at most 2^κ distinct corresponding matrices. On the other hand, for each matrix $X \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ for which, for some $x \in \mathbb{F}_{2^\kappa}$, $\text{rank}(X + x) < r^*$ holds, we necessarily have that $X + x = Y$, where $Y \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ is of rank less than r^*. We undertake to count such matrices Y.

Lemma 3.11. *For each rank $r^* \in \{1, \ldots, \kappa\}$, at most $2^{(\kappa+\widehat{\kappa}) \cdot (r^*-1)}$ matrices $Y \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ satisfy $\text{rank}(Y) < r^*$.*

Proof. Each matrix $Y \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ of rank less than r^* can be written (possibly non-uniquely) as the product of a $\kappa \times (r^* - 1)$ matrix and an $(r^* - 1) \times \widehat{\kappa}$ matrix. □

The set of matrices $X \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ for which $\min_{x \in \mathbb{F}_{2^\kappa}} \text{rank}(X + x) < r^*$ is exactly the union, over all field elements $x \in \mathbb{F}_{2^\kappa}$, of the sets $\{x + Y \mid \text{rank}(Y) < r^*\} \subset \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$. In light of Lemma 3.11, we conclude that the cardinality of this union is at most $2^\kappa \cdot 2^{(\kappa+\widehat{\kappa}) \cdot (r^*-1)} = 2^{(\kappa+\widehat{\kappa}) \cdot (r^*-1)+\kappa}$. Finally, the *total* number of $\kappa \times \widehat{\kappa}$ matrices X is obviously $2^{\kappa \cdot \widehat{\kappa}}$. The probability, over the random coefficients $(\chi_i)_{i=0}^{l'-1}$, that $\min_{x \in \mathbb{F}_{2^\kappa}} \text{rank}(F'''_x) < r^*$ is thus at most $2^{(\kappa+\widehat{\kappa}) \cdot (r^*-1)+\kappa-\kappa \cdot \widehat{\kappa}}$. As this quantity, for each fixed κ and r^*, is *decreasing* in $\widehat{\kappa}$ (since $r^* \leq \kappa$), and as Lemma 3.10 implies that $\widehat{\kappa} \geq \frac{\kappa}{m} - 1$, we conclude that the probability in question is at most

$$2^{\left(\kappa + \frac{\kappa}{m} - 1\right) \cdot (r^*-1) + \kappa - \kappa \cdot \left(\frac{\kappa}{m} - 1\right)} = 2^{\kappa \cdot (r^*-1) - \kappa \cdot \frac{\kappa - r^* + 1}{m} - (r^*-1) + 2 \cdot \kappa} \leq 2^{\kappa \cdot (r^*+1) - \kappa \cdot \frac{\kappa - r^* + 1}{m}}.$$

This completes the proof of the proposition. □

We now consider the distinguisher's distinguishing probability. We recall that the real and ideal distributions are identical unless $\mathcal{A}$ passes the correlation check *and* D queries $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$, for some $i \in \{0, \ldots, l-1\}$. On the other hand—if $\mathcal{A}$ passes the correlation check—D may learn information about Δ by means of brute-force queries of the form $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \mathbf{r})$, where $i \in \{0, \ldots, l-1\}$ and $\mathbf{r} \in \{\mathbf{e}_i * \Delta \mid \Delta \in \ker(F'_x)\}$. Specifically, upon each such query, D may rule in or out (i.e., depending on whether equality holds) the candidate $\mathbf{r}$ for the value of the projection $\mathbf{e}_i * \Delta$. The proposition below argues that D, with high probability, gains no information about the bits of Δ *outside* of $\mathbf{d}$, throughout its queries.

Proposition 3.12. *For each computationally unbounded distinguisher D, which makes at most $Q(\kappa)$ queries to the random oracle, say, and each modesty $m \in \{1,\ldots,\kappa\}$ and rank cutoff $r^* \in \{1,\ldots,\kappa\}$, we have that $\left|\Pr\left[D\left(\mathsf{Real}_{\Pi^{\kappa,l}_{\mathsf{ROT}},\mathcal{A},R}(\kappa)\right)=1\right]-\Pr\left[D\left(\mathsf{Ideal}_{\mathcal{F}^{\kappa,l}_{\mathsf{ROT}},\mathcal{S},R}(\kappa)\right)=1\right]\right| \leq Q(\kappa)\cdot 2^{-m}$, where we condition both distributions on $\mathcal{A}$'s matrix having modesty m and on the rank relation $r < r^*$.*

Proof. We write $\mathbf{d}$ for the vector constructed during the course of $\textsc{Modest}\left((\mathbf{x}_i)_{i=0}^{l-1}, m\right)$, and $\mathbf{w} := \mathbf{d} * \Delta$ for the projection of the sender's secret choice vector Δ onto $\mathbf{d}$. It suffices to prove the result after giving D $\mathbf{w}$, since this information can only make D more effective.

We write $Y := \{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w} \wedge \Delta \in \ker(F'_x)\}$ for the intersection in $\mathbb{F}_2^\kappa$ between $\ker(F'_x)$ and the $\left|\overline{\mathbf{d}}\right|$-dimensional affine-linear subspace $\{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w}\}$. Clearly, D's view of Δ begins as precisely uniform over Y. We note that $\dim(Y) \geq \left|\overline{\mathbf{d}}\right| - \mathrm{rank}(F'_x)$; this fact follows from a straightforward dimension-count. For each vector $\mathbf{f} \in \mathbb{F}_2^\kappa$, we write $Y_{\mathbf{f}} := \{\mathbf{f} * \Delta \mid \Delta \in Y\}$ for the projection of Y onto $\mathbf{f}$. Slightly abusing notation, we moreover write $\mathbf{f} : Y \to Y_{\mathbf{f}}$ for the natural projection map. For each $\mathbf{f}$, as the image point $\mathbf{r} \in Y_{\mathbf{f}}$ varies, the fibers $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ partition Y into equally-sized, parallel affine subspaces.

For $\mathbf{f} \in \mathbb{F}_2^\kappa$ again arbitrary, the rank–nullity theorem entails that $\dim(Y) - \dim\left(\mathbf{f}^{-1}(\mathbf{r})\right) = \dim(Y_{\mathbf{f}})$ for each $\mathbf{r} \in Y_{\mathbf{f}}$. We note that each $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ is the intersection in $\mathbb{F}_2^\kappa$ between $\ker(F'_x)$ and affine subspace $\{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w} \wedge \mathbf{f} * \Delta = \mathbf{r}\}$, and so is of dimension at most that of this latter space, which is clearly $\left|\overline{\mathbf{d} \cup \mathbf{f}}\right| = \left|\overline{\mathbf{d}} \cap \overline{\mathbf{f}}\right|$. Using these facts, we obtain the following estimate for $\dim(Y_{\mathbf{f}})$, valid for each $\mathbf{r} \in Y_{\mathbf{f}}$:

$$\dim(Y_{\mathbf{f}}) = \dim(Y) - \dim\left(\mathbf{f}^{-1}(\mathbf{r})\right) \geq \left|\overline{\mathbf{d}}\right| - \left|\overline{\mathbf{d}} \cap \overline{\mathbf{f}}\right| - \mathrm{rank}(F'_x) = |\mathbf{f} \setminus \mathbf{d}| - \mathrm{rank}(F'_x).$$

For each $i \in \{0,\ldots,l-1\}$, there are three cases of interest: $\mathbf{f} = \mathbf{e}_i$, where $\mathbf{e}_i \subset \mathbf{d}$, $\mathbf{f} = \mathbf{e}_i$, where $\mathbf{e}_i \not\subset \mathbf{d}$, and $\mathbf{f} = \overline{\mathbf{e}_i}$. In the first case, the projection $Y_{\mathbf{f}}$ consists of the single point $\mathbf{r} = \mathbf{e}_i * \mathbf{w}$, and the fiber $\mathbf{e}_i^{-1}(\mathbf{r})$ consists of the entirety of Y. Moreover, the equality $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \,\|\, \mathbf{t}_i + \mathbf{r})$ is guaranteed to hold precisely when $\mathbf{r} = \mathbf{e}_i * \mathbf{w}$, which D already knows. We thus ignore this case. We note that each $\mathbf{e}_i \not\subset \mathbf{d}$ is necessarily white, and so satisfies $|\mathbf{e}_i \setminus \mathbf{d}| \geq m$. Similarly, Lemma 3.4 implies that each $\overline{\mathbf{e}_i}$ satisfies $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq m$. We thus conclude that, in the second and third cases, $|\mathbf{f} \setminus \mathbf{d}| \geq m$, so that $\dim(Y_{\mathbf{f}}) \geq m - \mathrm{rank}(F'_x)$.

Definition 3.13. If, for some $i \in \{0,\ldots,l-1\}$, and for a vector $\mathbf{f}$ satisfying $\mathbf{f} \not\subset \mathbf{d}$ and equal either to $\mathbf{e}_i$ or to $\overline{\mathbf{e}_i}$, D submits the query $H(i \,\|\, \mathbf{t}_i + \mathbf{r})$, where $\mathbf{r} \in Y_{\mathbf{f}}$, then we say D has *checked* the fiber $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$.

By the calculation given above, each fiber $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ as in Definition 3.13 is of *codimension* at least $m - \mathrm{rank}(F'_x)$ in Y, and so covers a proportion consisting of at most $2^{\mathrm{rank}(F'_x)-m}$ among Y's points. We thus conclude that D can check, in total, a proportion consisting of at most $Q(\kappa)\cdot 2^{\mathrm{rank}(F'_x)-m}$ among Y's points. On the other hand, if D never checks a fiber containing Δ, then D's view is identical in the real and ideal worlds. Indeed, these distributions differ only if $\Delta \in \overline{\mathbf{e}_i}^{-1}(\mathbf{r})$ holds for some fiber $\overline{\mathbf{e}_i}^{-1}(\mathbf{r})$ checked by D (equivalently, if D queries $H(i \,\|\, \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$). Moreover, D can learn information about Δ—beyond the string $\mathbf{w}$ it already received—only if $\Delta \in \mathbf{e}_i^{-1}(\mathbf{r})$ holds for some fiber $\mathbf{e}_i^{-1}(\mathbf{r})$ checked by D (i.e., if D queries $H(i \,\|\, \mathbf{t}_i + \mathbf{e}_i * \Delta)$). If neither of these events happen, then D's view is completely independent of $\overline{\mathbf{d}} * \Delta$, and of whether it's in the real or ideal world.

Finally, $\mathcal{A}$'s chance of passing the correlation check is exactly $2^{-\mathrm{rank}(F'_x)}$. The probability with which D's environment differs in the two worlds is thus at most $2^{-\mathrm{rank}(F'_x)}\cdot Q(\kappa)\cdot 2^{\mathrm{rank}(F'_x)-m} = Q(\kappa)\cdot 2^{-m}$. □

We are now in a position to prove the theorem. We set $r^* := \sqrt{\kappa}$ for the rest of the proof. Traversing the tree of Figure 1, and invoking Propositions 3.9 and 3.12, we see that for each distinguisher D as above and each modesty $m \in \{1,\ldots,\kappa\}$, the difference $\left|\Pr\left[D\left(\mathsf{Real}_{\Pi^{\kappa,l}_{\mathsf{ROT}},\mathcal{A},R}(\kappa)\right)=1\right]-\Pr\left[D\left(\mathsf{Ideal}_{\mathcal{F}^{\kappa,l}_{\mathsf{ROT}},\mathcal{S},R}(\kappa)\right)=1\right]\right|$—where, here, we condition the two distributions *only* on the modesty m—is at most:

$$\min\left(1, 2^{\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{m}}\right)\cdot\min\left(1, Q(\kappa)\cdot 2^{-m}\right) + 2^{-r^*}. \tag{2}$$

We handle two cases, corresponding to whether $m < \frac{1}{2}\cdot\sqrt{\kappa}$ or not. If $m < \frac{1}{2}\cdot\sqrt{\kappa}$, then the first factor of (2)'s exponent is at most $\kappa^{3/2}+\kappa-\frac{\kappa^2-\kappa^{3/2}+\kappa}{m} < \kappa^{3/2}+\kappa-2\cdot\kappa^{3/2}+2\cdot\kappa-2\cdot\sqrt{\kappa} \in -\Omega(\kappa^{3/2})$, so this factor is negligible, and the result holds. If $m \geq \frac{1}{2}\cdot\sqrt{\kappa}$, then the second factor's exponent is at most $-\frac{1}{2}\cdot\sqrt{\kappa} \in -\Omega(\sqrt{\kappa})$, so this factor instead is negligible (provided that $Q(\kappa)$ is polynomial). The final summand's exponent is $-\sqrt{\kappa} \in -\Omega(\sqrt{\kappa})$, so this term is negligible. This completes the proof of the theorem. □

We now extract effective bounds from our proof. Our proof can be made to yield concrete values κ at which KOS achieves prescribed security guarantees.

Theorem 3.14. *For given computational and statistical security parameters λ and s, respectively, in order for it to be the case that $\left|\Pr\left[D\left(\mathsf{Real}_{\Pi^{\kappa,l}_{\mathsf{ROT}},\mathcal{A},R}(\kappa)\right)=1\right]-\Pr\left[D\left(\mathsf{Ideal}_{\mathcal{F}^{\kappa,l}_{\mathsf{ROT}},\mathcal{S},R}(\kappa)\right)=1\right]\right|\leq 2^{-s}$ holds for each distinguisher D making at most 2^{λ} hash evaluations, it suffices that $\kappa\geq s^2+s\cdot\lambda+4\cdot s+2\cdot\lambda+2$.*

Proof. We set $r^*:=s+1$ once and for all, and assume $Q(\kappa)=2^{\lambda}$. We describe a selection procedure for κ which bounds (2) from above by 2^{-s} (i.e., for *each* $m\in\{1,\ldots,\kappa\}$).

Indeed, we set κ so that $\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{\lambda+r^*}\leq 0$ holds. As a simple algebraic manipulation demonstrates, this inequality occurs precisely when $\kappa\geq {r^*}^2+r^*\cdot\lambda+2\cdot r^*+\lambda-1$; substituting $r^*=s+1$, we obtain the expression $\kappa\geq s^2+s\cdot\lambda+4\cdot s+2\cdot\lambda+2$. For each fixed κ, λ, and r^*, we view the exponent expressions $\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{m}$ and $\lambda-m$ of (2) as functions of the *rational* variable $m\in(0,\kappa]$; we note that these functions are *increasing* and *decreasing*, respectively, over the interval $m\in(0,\kappa]$. For κ chosen as above, we write $m^*\in(0,\kappa]$ for the (generally non-integral) intersection point for which $\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{m^*}=0$. By the above, and by our choice of κ, we necessarily have that $m^*\geq\lambda+r^*$, so that $\lambda-m^*\leq -r^*$. We conclude that (2) is bounded from above by $2^{-r^*}+2^{-r^*}=2^{-s}$ at the point $m^*\in\mathbb{Q}$. It thus suffices to show that—for λ and r^* fixed, and for κ as selected above—the rational modesty m^* chosen above in fact *maximizes* (2).

Since (by choice of m^*) $\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{m}\geq 0$ whenever $m\geq m^*$, and because $\lambda-m$ is decreasing, we conclude that that (2) is decreasing over the interval $[m^*,\kappa]$. It thus suffices to show that the sum of the two exponent expressions is itself increasing over the interval $(0,m^*]$.

To this end, we show that the upward slope of $\kappa\cdot(r^*+1)-\kappa\cdot\frac{\kappa-r^*+1}{m}$ is steeper than the downward slope of $\lambda-m$ throughout the interval $(0,m^*]$. The derivative in m of the former expression is $\frac{\kappa\cdot(\kappa-r^*+1)}{m^2}$, which is at least $\frac{\kappa\cdot(\kappa-r^*+1)}{{m^*}^2}$ whenever $m\leq m^*$. Since $m^*=\frac{\kappa-r^*+1}{r^*+1}$, this derivative is thus at least $\frac{\kappa\cdot(r^*+1)^2}{\kappa-r^*+1}\geq(r^*+1)^2\geq 1$, as desired. This completes the proof. □

Remark 3.15. Theorem 3.14 can be viewed as a precise variant of the final argument of Theorem 3.1, in which s and λ are prescribed, and where we moreover select the modesty cutoff optimally (i.e., in such a way as to make (2) decay as quickly as possible). Indeed, for κ chosen as in Theorem 3.14, the optimal cutoff—and the most effective attack strategy for the adversary—appears at the modesty $\lceil m^*\rceil=\left\lceil\frac{\kappa-r^*+1}{r^*+1}\right\rceil$.

Example 3.16. For $s:=30$, and $\lambda:=60$, Theorem 3.14 guarantees security as long as $\kappa\geq 2{,}942$.

Example 3.17. For $s:=40$ and $\lambda:=80$, Theorem 3.14 guarantees security as long as $\kappa\geq 5{,}122$.

Example 3.18. For $s:=80$ and $\lambda:=128$, Theorem 3.14 guarantees security as long as $\kappa\geq 17{,}218$.

Theorem 3.14 yields parameter sizes which are barely practicable, if at all. It is, of course, possible that our proof could be strengthened (or another proof found), so as to yield stronger bounds, and security under more reasonable parameter sizes. On the other hand, an improvement to our result seems out of reach, barring strikingly new techniques. We explain this as follows.

Corollary 3.19. *For κ sufficiently large, for each distinguisher D making at most $\frac{1}{2}\cdot\sqrt{\kappa}\cdot 2^{\frac{1}{2}\cdot\sqrt{\kappa}}$ hash evaluations, the probability of success $\left|\Pr\left[D\left(\mathsf{Real}_{\Pi^{\kappa,l}_{\mathsf{ROT}},\mathcal{A},R}(\kappa)\right)=1\right]-\Pr\left[D\left(\mathsf{Ideal}_{\mathcal{F}^{\kappa,l}_{\mathsf{ROT}},\mathcal{S},R}(\kappa)\right)=1\right]\right|\leq 2^{-\frac{1}{2}\cdot\sqrt{\kappa}}$.*

Proof. For arbitrary κ, we set $s:=\frac{1}{2}\cdot\sqrt{\kappa}$ and $\lambda:=\frac{1}{2}\cdot\sqrt{\kappa}+\frac{1}{2}\cdot\log(\kappa)-1$. We observe that for s and λ chosen this way—at least if $\kappa\geq 93$—we have $\kappa\geq s^2+s\cdot\lambda+4\cdot s+2\cdot\lambda+2$. Theorem 3.14 thus implies that any attack using at most $2^{\lambda}=\frac{1}{2}\cdot\sqrt{\kappa}\cdot 2^{\frac{1}{2}\cdot\sqrt{\kappa}}$ hashes must succeed with probability at most $2^{-s}=2^{-\frac{1}{2}\cdot\sqrt{\kappa}}$. □

In other words, there does not exist an attack on KOS which uses only $\frac{1}{2}\cdot\sqrt{\kappa}\cdot 2^{\frac{1}{2}\cdot\sqrt{\kappa}}$ hash evaluations and succeeds with probability greater than $2^{-\frac{1}{2}\cdot\sqrt{\kappa}}$.

Remark 3.20. In Corollary 3.19, the constant of $\frac{1}{2}$ present in both exponents can be improved to $\frac{1}{\sqrt{2}}-\varepsilon$—for ε arbitrarily small—at the cost of increasing the implicit cutoff κ at which the corollary becomes effective.

Roy [Roy22, § 4.1] describes a "subfield attack" on KOS, which requires $2^{\frac{1}{5}\cdot\kappa}$ oracle queries and succeeds with probability $2^{-\frac{3}{5}\cdot\kappa}$. This attack is significantly more costly and unlikely to succeed than those which our proof rules out; the analysis of KOS thus still contains a gap. (Of course, the attack is nonetheless stronger than those which KOS's original proof sought to rule out.) On the other hand, Roy [Roy22, § 4.1] describes a *further* attack on a different protocol—namely, "PSS", for Patra, Sarkar and Suresh [PSS17]—which is much more devastating; that attack requires $\frac{1}{2}\cdot\sqrt{\kappa}\cdot 2^{\sqrt{\kappa}}$ hash evaluations and succeeds with probability $2^{-\sqrt{\kappa}}$. As it turns out, our proof serves equally well—without change—to describe the security of PSS. Indeed, we use *only* the property of the field elements $(\chi_i)_{i=0}^{l'-1}$ whereby, for $\chi_i \leftarrow \mathbb{F}_{2^\kappa}$ sampled randomly, each *individual* column of χ_i's matrix representation is itself uniformly random in $\{0,1\}^\kappa$. (Of course, the columns, considered jointly, are not independently random.) This property holds also for PSS, in fact, even though they construct their matrices differently (with a single random column repeated).

The lower-bound established by our Corollary 3.19, which applies to both KOS and PSS, exactly matches—up to the constant $\frac{1}{2}$ appearing in the expressions' exponents—the upper-bound achieved by Roy [Roy22, § 4.1] on PSS. Our proof thus definitively settles the question of PSS's security (up to the constant). Moreover, it demonstrates that any *better* security argument for KOS—if one exists—would have to rely in some special way on the structure of the field elements $(\chi_i)_{i=0}^{l'-1}$, and on the nature of their role in the correlation check. We emphasize that Roy's attack on PSS is *not* known to apply to KOS. Rather, the opposite is true; our defense of KOS applies to PSS. The security of KOS thus resides somewhere between the lower-bound established by our Theorem 3.14 and the upper-bound achieved by Roy's subfield attack. In any case, our result furnishes the *only* currently-known lower-bound for KOS, and its only proof of security.

We leave the task of exactly settling KOS's security to future work. In the meantime, we suggest that an alternative known to be concretely secure—like Roy's *SoftSpokenOT* [Roy22]—be used when possible.

References

[Coh82] P. M. Cohn. *Algebra*, volume 1. John Wiley & Sons, second edition, 1982.

[KOS15] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure OT extension with optimal overhead. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 724–741, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[KOS22] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure OT extension with optimal overhead. Unpublished update, `https://eprint.iacr.org/2015/546.pdf`, September 2022.

[Lin17] Yehuda Lindell. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, chapter How to Simulate It – A Tutorial on the Simulation Proof Technique, pages 277–346. Information Security and Cryptography. Springer International Publishing, 2017.

[MR19] Daniel Mansy and Peter Rindal. Endemic oblivious transfer. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 309–326, New York, NY, USA, 2019. Association for Computing Machinery.

[PSS17] Arpita Patra, Pratik Sarkar, and Ajith Suresh. Fast actively secure OT extension for short secrets. In *Network and Distributed System Security Symposium*. Internet Society, 2017.

[Roy22] Lawrence Roy. SoftSpokenOT: Quieter OT extension from small-field silent VOLE in the Minicrypt model. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, volume 13507 of *Lecture Notes in Computer Science*, pages 657–687, Cham, 2022. Springer Nature Switzerland.