On the Security of KOS

Benjamin E. Diamond*

Irreducible

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. 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 certain 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; 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 3.27). 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 3.29).

This sort of κ results in a barely-practical protocol. On the other hand, we demonstrate that this limitation is intrinsic to our proof strategy. 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 3.30 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—will 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.

1.1 Summary of KOS

We recall the details of KOS. We fix a security parameter κ and a problem size l. In the *random oblivious transfer* paradigm [KOS15, § 2.2], the *receiver* inputs a vector of choice bits $(x_i)_{i=0}^{l-1}$ and the *sender* inputs nothing. The functionality $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ (see also our Functionality 2.5 below) samples, for each $i \in \{0,\ldots,l-1\}$, random κ-bit pads $\mathbf{v}_{i,0}$ and $\mathbf{v}_{i,1}$. It outputs $(\mathbf{v}_{i,0},\mathbf{v}_{i,1})_{i=0}^{l-1}$ to the sender and $(\mathbf{v}_{i,x_i})_{i=0}^{l-1}$ to the receiver. By the aid of a standard *derandomization* procedure, random oblivious transfer itself suffices to yield traditional oblivious transfer; this reduction is given in [KOS15, § 3.4].

KOS introduces a key protocol [KOS15, Fig. 7] for random OT (we reproduce KOS as our Protocol 2.7 below). That protocol, internally, invokes a further ideal, hybrid functionality, which KOS calls *correlated OT with errors*. The correlated OT with errors functionality serves precisely to abstract away the role played the seminal subprotocol of Ishai, Kilian, Nissim and Petrank [IKNP03, Fig. 1] (henceforth, "IKNP"); that protocol itself appears, in a slightly adjusted guise, as [KOS15, Fig. 3] (we refer also to Nielsen [Nie07, § 2], which proposes a similar formalism). We reproduce—and slightly amend—KOS's correlated OT with errors functionality as our Functionality 2.3 below (see also Remark 2.4). That functionality, which we call $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, formalizes the scope for cheating which IKNP admits. In other words, though IKNP is not maliciously secure as a random OT protocol in its own right, it nonetheless does securely instantiate the functionality $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ (essentially by design).

The internal functionality $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ allows the receiver to input an arbitrary $l' \times \kappa$ binary matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$; the sender moreover submits a single, secret choice vector $\Delta \in \{0,1\}^\kappa$. By definition, for each row-index $i \in \{0,\ldots,l'-1\}$, that functionality samples a random κ-bit pad $\mathbf{t}_i \leftarrow \{0,1\}^\kappa$ and sets $\mathbf{q}_i \coloneqq \mathbf{t}_i + \mathbf{x}_i * \Delta$; here, we write $*$ for the *bitwise AND* operation. It finally returns $(\mathbf{t}_i)_{i=0}^{l'-1}$ to the receiver and $(\mathbf{q}_i)_{i=0}^{l'-1}$ to the sender. In short, that functionality outputs to the parties random matrices $(\mathbf{t}_i)_{i=0}^{l'-1}$ and $(\mathbf{q}_i)_{i=0}^{l'-1}$ which differ by the secret matrix $(\mathbf{x}_i * \Delta)_{i=0}^{l'-1}$; here, $(\mathbf{x}_i)_{i=0}^{l'-1}$ is chosen by the receiver and $\Delta \in \{0,1\}^\kappa$ by the sender.

If the receiver is honest, with input choice vector $(x_i)_{i=0}^{l-1}$ say, then $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$—here, we temporarily specialize $l' \coloneqq l$—suffices to effect an l-fold random OT of κ-bit strings, as we now explain. (Indeed, this fact underlies IKNP's suitability as a random OT protocol secure against semi-honest adversaries.) Indeed, R may simply, for each $i \in \{0,\ldots,l-1\}$, set its vector $\mathbf{x}_i \coloneqq x_i \cdot (1,\ldots,1)$ *monochromatically*; that is, for each $i \in \{0,\ldots,l-1\}$, R sets as $\mathbf{x}_i$ the κ-fold repetition of its choice bit x_i. This behavior on the part of R induces $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l}$ to send S the pad $\mathbf{q}_i \coloneqq \mathbf{t}_i + \mathbf{x}_i * \Delta = \mathbf{t}_i + x_i \cdot \Delta$, for each $i \in \{0,\ldots,l-1\}$. Having begun in this way, R may output $\mathbf{v}_{i,x_i} \coloneqq H(i \,\|\, \mathbf{t}_i)$ for each $i \in \{0,\ldots,l-1\}$; S, for its part, may output $\mathbf{v}_{i,0} \coloneqq H(i \,\|\, \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \,\|\, \mathbf{q}_i + \Delta)$ for each $i \in \{0,\ldots,l-1\}$. In Remark 2.8 below, we demonstrate that this procedure yields a correct random OT (again, assuming honest behavior by R).

If R is dishonest, on the other hand, it may submit an arbitrary $l' \times \kappa$ binary matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$ (its rows not necessarily monochromatic). In this way, R may introduce a subtle sort of leakage into the protocol transcript, perceptible to the distinguisher D (we discuss this fact further in Subsection 1.2 below, as well as in Subsection 3.2). KOS seeks to establish malicious security against R, in the above setting, by adding to IKNP a *correlation check*. That is, the parties begin by fixing a field structure on the set of κ-bit strings; that is, they introduce the field $\mathbb{F}_{2^\kappa}$. Since this object is isomorphic as an $\mathbb{F}_2$-vector space to $\mathbb{F}_2^\kappa$, the parties may freely interpret their respective elements $\mathbf{t}_i$ and $\mathbf{q}_i$ of $\mathbb{F}_2^\kappa$, for each $i \in \{0,\ldots,l'-1\}$, as $\mathbb{F}_{2^\kappa}$-elements. Using a coin-tossing functionality, the parties jointly sample l' further random field elements $\chi_i \leftarrow \mathbb{F}_{2^\kappa}$, for $i \in \{0,\ldots,l'-1\}$. Finally, the sender and receiver jointly $\mathbb{F}_{2^\kappa}$-linearly combine their respective equalities $\mathbf{q}_i = \mathbf{t}_i + x_i \cdot \Delta$, for $i \in \{0,\ldots,l'-1\}$, using the scalars $(\chi_i)_{i=0}^{l'-1}$. That is, the sender computes $q \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$; the receiver computes both $t \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$ and $x \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot x_i$ (here, we use $\cdot$ to denote field multiplication). The receiver sends t and x to the sender, who finally checks $q \stackrel{?}{=} t + x \cdot \Delta$.

On its face, this check may leak information about R's secret vector $(x_i)_{i=0}^{l-1}$ to S. Fortunately, KOS [KOS15, Thm. 1] shows that, by setting $l' \geq l$ sufficiently large—and prescribing that the receiver sample its further choice bits $(x_i)_{i=l}^{l'-1}$ randomly—the parties may guarantee that, with high probability over $(\chi_i)_{i=0}^{l'-1}$, no leakage occurs (specifically, it suffices to set $l' \coloneqq l + \kappa + s$, where s is a statistical security parameter). The correlated OTs indexed $i \in \{l,\ldots,l'-1\}$, then, serve rather to protect the receiver, and not the sender; in this sense, these OTs are "sacrificed".

We turn to the impact of the correlation check on the corrupt receiver R.

1.2 Our Proof

In Subsection 3.1 below, we thoroughly treat, for the sake of completeness, the case in which the sender S is corrupt; in this endeavor, we largely follow [KOS15, § 3.2]. In Subsection 3.2, we turn to the difficult case in which the receiver is corrupt. In that subsection, we present a much-more-rigorous rendition of this subsection's content. For now, we sketch our approach informally.

As we note below, it is not difficult for $\mathcal{S}$ to simulate a perfectly indistinguishable view to the adversary $\mathcal{A}$. On the other hand, provided that $\mathcal{A}$ passes the correlation check, our distinguisher encounters real and ideal worlds which subtly differ; specifically, these worlds' respective random oracles remain in slightly different states. If $\mathcal{A}$ fails the correlation check, on the other hand, then the sender (in the real world) and simulator (in the ideal world) both abort, thereby excising from both the real and ideal transcripts the honest party's outputs. Deprived of this critical resource, D becomes powerless; in those executions for which $\mathcal{A}$ fails the check, our simulation thus becomes perfectly secure. Our proof, then, proceeds in the inevitable way; that is, it considers simultaneously D's probability of distinguishing (conditioned on $\mathcal{A}$ passing) and $\mathcal{A}$'s chance of passing in the first place. Ultimately, we are able to show that at least one of these quantities—i.e., depending on $\mathcal{A}$'s behavior—is necessarily bounded from above by a negligible function in κ.

We begin by characterizing in detail the discrepancies—in D's view—between the real and ideal worlds, conditioned on $\mathcal{A}$ passing the check. This characterization itself seems largely new, though it contains echoes of Nielsen [Nie07, § 4]. Below, we express D's object in terms of a certain "game", which we presently sketch (see also Definition 3.4, as well as the following lemmas, for a rigorous treatment). D's game takes place on a "board" consisting of an $l \times \kappa$ binary matrix. The adversary $\mathcal{A}$ begins by submitting $(\mathbf{x}_i)_{i=0}^{l-1}$ (we ignore this matrix's lower part $(\mathbf{x}_i)_{i=l}^{l'-1}$, since it can't help D). $\mathcal{S}$, for its part, is granted the privilege of flipping (i.e., inverting all the bits of) any among this matrix's rows. We write $(\mathbf{e}_i)_{i=0}^{l-1}$ for the resulting matrix (i.e., after $\mathcal{S}$'s intervention). D finally plays on the board $(\mathbf{e}_i)_{i=0}^{l-1}$. D's goal is to "control" as many of the matrix's columns $j \in \{0, \ldots, \kappa - 1\}$ as possible. More precisely, D's goal is to control—for some (i.e., any) particular row index $i \in \{0, \ldots, l-1\}$—all of the columns $j \in \{0, \ldots, \kappa - 1\}$ at which $\mathbf{e}_{i,j} = 0$. Each time D "plays" a row $i \in \{0, \ldots, l-1\}$, it thereby "gains control" of those column indices $j \in \{0, \ldots, \kappa - 1\}$ at which $\mathbf{e}_{i,j} = 1$. Critically, however, D seeks to win in such a way that each particular row it plays seizes as few new columns as possible. We say that the game is *easy* for D if it can win even by seizing but few new columns at each successive step. If D may win only by seizing—at some point during its execution—many new columns in a single step, then we say that the game is *hard* for D. $\mathcal{S}$'s goal is to make the game as hard as possible for D to win; $\mathcal{A}$'s goal is to make it easy for D (even modulo $\mathcal{S}$'s best efforts to the contrary).

Fascinatingly, this game captures both D's distinguishing probability and $\mathcal{A}$'s passing probability, as we now explain. On the one hand, as the game gets "harder" for D to win, D's probability of distinguishing the two worlds decreases (this fact is made precise in Proposition 3.18). On the other hand, as $\mathcal{A}$ submits board states which—notwithstanding $\mathcal{S}$'s best efforts to the contrary—make D's task easier, $\mathcal{A}$'s chance of passing the correlation check in the first place becomes smaller (this fact is proved in Proposition 3.15). In Definition 3.4 below, we rigorously define this game. That definition moreover serves to "score" the difficulty of each given matrix submitted by $\mathcal{A}$; that is, it assigns to each matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ a numerical score $m \in \{1, \ldots, \kappa\}$, which we call $(\mathbf{x}_i)_{i=0}^{l-1}$'s *modesty*. That definition also produces $\mathcal{S}$'s bit assignment $(x_i)_{i=0}^{l-1}$ (i.e., it tells $\mathcal{S}$ which rows to "flip"). We note that each monochromatic matrix $(\mathbf{x}_i)_{i=0}^{l-1}$—i.e., submitted by an honest receiver—has the maximal possible modesty, namely κ (we explain this in Example 3.6 below).

In the most technically difficult part of our proof, we show that an adversary $\mathcal{A}$ who submits an immodest matrix—i.e., one which is easy for D—is unlikely to pass the correlation check. To do this, we analyze in detail KOS's correlation check (and for that matter, PSS's). Crucially, $\mathcal{A}$ gets to choose its final messages x and t adaptively, after seeing the combination coefficients $(\chi_i)_{i=0}^{l'-1}$. (Indeed, a failure to account for this fact underlies the flaw in KOS's original proof [KOS15, Lem. 1]; we refer to Roy [Roy22, § 4.1.3] for further details.) Roughly, we show that—again, for $(\mathbf{x}_i)_{i=0}^{l-1}$ immodest—$\mathcal{A}$'s task amounts to finding, given a binary, $\kappa \times \kappa$ matrix X many of whose columns are uniformly random, a field element $x \in \mathbb{F}_{2^\kappa}$ for which $X + \left[x\right]$ is of low rank; here, we denote by $\left[x\right]$ the binary, $\kappa \times \kappa$ matrix which captures x's action on $\mathbb{F}_2^\kappa$ by field multiplication. As we argue in Proposition 3.15 below, since the space $\mathbb{F}_2^{\kappa \times \kappa}$ of binary matrices is enormous, while the number of field elements $x \in \mathbb{F}_2^\kappa$ is relatively "small", a random matrix $X \in \mathbb{F}_2^{\kappa \times \kappa}$ is unlikely even to fall "near" a field-element matrix $\left[x\right]$—i.e., in the sense that $\mathrm{rank}(X + \left[x\right])$ is low—let alone to equal one.

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 $\mathbb{F}_2^\kappa$. 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 $\mathbb{F}_2$-vector space $\mathbb{F}_2^\kappa$, by means of the $\mathbb{F}_2$-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 understand each expression of the form $(\mathbf{x}_i)_{i=0}^{l'-1}$ as an $l' \times \kappa$ matrix, with entries in $\mathbb{F}_2$; i.e., for each $i \in \{0,\ldots,l'-1\}$, the element $\mathbf{x}_i \in \mathbb{F}_2^\kappa$ is a row-vector. We write $\overline{\mathbf{x}_i}$ for the bitwise complement of the row-vector $\mathbf{x}_i \in \mathbb{F}_2^\kappa$, and $\overline{x_i}$ for the complement of the bit $x_i \in \mathbb{F}_2$. We write $w(\mathbf{x}_i)$ for the *Hamming weight* of the vector $\mathbf{x}_i$.

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 define maliciously secure two-party computation, following Lindell [Lin17, § 6.6.2]. Our use of the random oracle induces various subtleties, which we pause to explain. We operate in the *nonprogrammable random oracle model*, due apparently to Nielsen [Nie02], and discussed explicitly in both Ishai, Kilian, Nissim and Petrank [IKNP03, § 2.1] and Lindell [Lin17, § 6.10.2]. In the nonprogrammable random oracle model, the random oracle exists externally in both in the real and ideal worlds; in particular, the simulator $\mathcal{S}$ can't "program" it. Finally, the distinguisher D—upon being fed a transcript, drawn from either of the two worlds—inherits the state of the oracle. Critically, D may attempt to distinguish the two worlds by correlating its oracle's behavior with its on-hand execution transcript.

We finally allow real-world protocols which make use of hybrid sub-functionalities; we follow Lindell [Lin17, § 6.6.3]'s treatment of this matter. That is, we fix ideal sub-functionalities $\mathcal{F}_0, \ldots, \mathcal{F}_{p-1}$. We require, as Lindell does, that each protocol Π call these ideal functionalities strictly sequentially, as well as that no protocol messages intervene between the respective stages of any multi-stage (or *reactive*) functionality.

Definition 2.1. For each functionality $\mathcal{F}$, sub-functionalities $\mathcal{F}_1, \ldots, \mathcal{F}_p$, 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 v_{1-C} to P_{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 in the nonprogrammable random oracle and $\mathcal{F}_1, \ldots, \mathcal{F}_p$-hybrid model*, or that Π *securely computes $\mathcal{F}$ in the $\mathcal{F}_1, \ldots, \mathcal{F}_p$-hybrid model*, if, for each corrupt party $C \in \{0,1\}$ and each probabilistic polynomial-time adversary $\mathcal{A}$ corrupting P_C, there is a probabilistic polynomial-time simulator $\mathcal{S}$ corrupting P_C in the ideal world such that the distributions $\left\{\mathsf{Real}_{\Pi,\mathcal{A},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)\right\}_{(\mathbf{x}_0,\mathbf{x}_1),\kappa}$ and $\left\{\mathsf{Ideal}_{\mathcal{F},\mathcal{S},C}((\mathbf{x}_0,\mathbf{x}_1),\kappa)\right\}_{(\mathbf{x}_0,\mathbf{x}_1),\kappa}$ are computationally indistinguishable, where we require $\mathbf{x}_0$ and $\mathbf{x}_1$ throughout to have equal lengths.

Comparison with the fully programmable model. In the *fully programmable random oracle model* (see [Nie02, § 3]), $\mathcal{S}$ simulates internally the random oracle to $\mathcal{A}$, just as it does the various other hybrid functionalities Π makes use of. This model differs from the nonprogrammable model, firstly, in that it allows $\mathcal{S}$ to program its oracle arbitrarily (i.e., subject to the usual requirement whereby it simulate an indistinguishable view to $\mathcal{A}$). More importantly, the random oracle is "wiped" or "destroyed" before D begins its execution. This latter model is much weaker than that which we consider. That is, it more readily deems protocols "secure" than ours does [Nie02]; it also fails to guarantee sequential composition [Lin17, § 6.10.2].

Interestingly, a clear separation between these models is exhibited by KOS itself, or rather by its predecessor work Ishai, Kilian, Nissim, and Petrank [IKNP03] (IKNP). IKNP differs from KOS (reproduced as our Protocol 2.7 below) in that lacks the correlation check entirely; it moreover makes possible the choice $l' \coloneqq l$. IKNP certainly fails to be maliciously secure in the nonprogrammable model (we explain this fact in Example 3.22 below and the remarks that follow it). In the fully programmable random oracle model, on the other hand, IKNP becomes maliciously "secure", with the concrete security parameter just $\kappa \coloneqq 2 \cdot \lambda + 2 \cdot s$, no less, and under a straightforward simulation strategy which opts simply to extract $x_i \coloneqq \textsc{Maj}(\mathbf{x}_i)$ for each $i \in \{0, \ldots, l-1\}$ (here, $\textsc{Maj} : \{0,1\}^\kappa \to \{0,1\}$ is the standard *majority* function on κ bits). We discuss this separation result—which we believe may be of independent interest—in Remark 3.23 below.

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.

- S sends $(\mathtt{initialize}, \Delta)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, where $\Delta \in \mathbb{F}_2^\kappa$.
- R sends $\left(\mathtt{input}, (\mathbf{x}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$. If both parties are honest, then, for each $i \in \{0, \ldots, l'-1\}$, $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ samples $\mathbf{t}_i \leftarrow \mathbb{F}_2^\kappa$ randomly and sets $\mathbf{q}_i \coloneqq \mathbf{t}_i + \mathbf{x}_i * \Delta$.
- If R is corrupt, then R sends the further input $\left(\mathtt{pads}, (\mathbf{t}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, which computes $(\mathbf{q}_i)_{i=0}^{l'-1}$ in the usual way.
- If S is corrupt, then S submits $\left(\mathtt{pads}, (\mathbf{q}_i)_{i=0}^{l'-1}\right)$ to $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, which, for each $i \in \{0, \ldots, l'-1\}$, instead sets $\mathbf{t}_i \coloneqq \mathbf{q}_i + \mathbf{x}_i * \Delta$.
- In each case, $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ outputs $\left(\mathtt{output}, (\mathbf{t}_i)_{i=0}^{l'-1}\right)$ to R and $\left(\mathtt{output}, (\mathbf{q}_i)_{i=0}^{l'-1}\right)$ 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 technical amendment is in fact necessary; the protocol [KOS15, Fig. 3] does not securely instantiate the functionality [KOS15, Fig. 2], but does—exactly as written—securely instantiate our Functionality 2.3 (i.e., in the $\mathcal{F}_{\mathsf{OT}}^{\kappa}$-hybrid model). This error apparently goes back to Nielsen [Nie07, § 2], to whose treatment [KOS15, § 2.2] defers. As it happens, [Nie07, § 2.1] itself offers very little in the way of proof, and in fact neglects entirely to treat the case of corrupt S.

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 $i \in \{0,\dots,l-1\}$, samples $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1}) \leftarrow \{0,1\}^\kappa \times \{0,1\}^\kappa$.
- If R is corrupt, R submits both $\left(\texttt{input}, (x_i)_{i=0}^{l-1}\right)$ and $\left(\texttt{pads}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\right)$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, which, for each $i \in \{0,\dots,l-1\}$, samples $\mathbf{v}_{i,\overline{x_i}} \leftarrow \{0,1\}^\kappa$.
- If S is corrupt, then S submits $\left(\texttt{pads}, (\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}\right)$ 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 $H(i \parallel \mathbf{q}_i + x_i \cdot \Delta) \coloneqq \mathbf{v}_{i,x_i}$ 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 $(x_0,\dots,x_{l-1})$.

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

Remark 2.8. We informally explain the correctness of Protocol 2.7 as follows. If both parties are honest, then, for each $i \in \{0,\dots,l'-1\}$, by definition of $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, $\mathbf{q}_i = \mathbf{t}_i + \mathbf{x}_i * \Delta = \mathbf{t}_i + x_i \cdot \Delta$ holds (here, we use the fact that $\mathbf{x}_i = x_i \cdot (1,\dots,1)$ is monochromatic). By linearly combining the l' instances of this equation using the coefficients $(\chi_i)_{i=0}^{l'-1}$, we obtain $q = \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i = \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{t}_i + x_i \cdot \Delta) = \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i + \left(\sum_{i=0}^{l'-1} \chi_i \cdot x_i\right) \cdot \Delta = t + x \cdot \Delta$, which is exactly the consistency check S runs. Finally, we note that S's outputs $\mathbf{v}_{i,0} \coloneqq H(i \parallel \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \parallel \mathbf{q}_i + \Delta)$ themselves equal $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + x_i \cdot \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + x_i \cdot \Delta + \Delta)$. We see that for each $b \in \{0,1\}$, S's output satisfies $\mathbf{v}_{i,b} = H(i \parallel \mathbf{t}_i + x_i \cdot \Delta + b \cdot \Delta)$; in particular S's output $\mathbf{v}_{i,x_i} = H(i \parallel \mathbf{t}_i + x_i \cdot \Delta + x_i \cdot \Delta) = H(i \parallel \mathbf{t}_i)$, which is exactly what R outputs. We discuss the randomness of S's other outputs $(\mathbf{v}_{i,\overline{x}_i})_{i=0}^{l-1}$—i.e., in the case of honest R—in Example 3.21 below.

3 Security proof

We now prove the security of Protocol 2.7.

3.1 Corrupt Sender

We first treat the case in which S is corrupt. Our treatment of this case is similar to that of [KOS15, Thm. 1].

The simulator's mandate. We begin by informally explaining $\mathcal{S}$'s mandate. Upon intercepting $\mathcal{A}$'s messages Δ and $(\mathbf{q}_i)_{i=0}^{l'-1}$ intended for $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, $\mathcal{S}$ must first extract from $\mathcal{A}$ ideal inputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$ for $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, which, in particular, induce $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ to send R ideal outputs $(\mathbf{v}_{i,x_i})_{i=0}^{l-1}$ which exactly match its real-world counterparts. In the real world, for each $i \in \{0, \ldots, l'-1\}$, R sends $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ the monochromatic vector $\mathbf{x}_i \coloneqq x_i \cdot (1, \ldots, 1)$, and receives in response $\mathbf{t}_i \coloneqq \mathbf{q}_i + \mathbf{x}_i * \Delta = \mathbf{q}_i + x_i \cdot \Delta$. We conclude that R's real-world outputs satisfy $\mathbf{v}_{i,x_i} \coloneqq H(i \parallel \mathbf{t}_i) = H(i \parallel \mathbf{q}_i + x_i \cdot \Delta)$, for each $i \in \{0, \ldots, l-1\}$. $\mathcal{S}$, therefore, may simply submit to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ the ideal-world inputs $\mathbf{v}_{i,0} \coloneqq H(i \parallel \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \parallel \mathbf{q}_i + \Delta)$. Though $\mathcal{S}$ doesn't know R's secret bits $(x_i)_{i=0}^{l-1}$, this construction guarantees that $\mathbf{v}_{i,x_i} = H(i \parallel \mathbf{q}_i + b \cdot \Delta)$ holds for both possible bits $b \in \{0,1\}$, so that $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$'s ideal output $\mathbf{v}_{i,x_i}$ too equals $H(i \parallel \mathbf{q}_i + x_i \cdot \Delta)$ (i.e., regardless of $x_i \in \{0,1\}$).

It remains for $\mathcal{S}$ to simulate R's messages x and t to $\mathcal{A}$. $\mathcal{S}$'s essential obstacle is that it doesn't know R's choice bits $(x_i)_{i=0}^{l-1}$; the distinguisher D does get access to these bits. On the other hand, $\mathcal{S}$ is critically aided by the further random bits $(x_i)_{i=l}^{l'-1}$ sampled by R during the protocol. Since these bits are internal values, independent of R's inputs, neither $\mathcal{S}$ nor D knows them. We argue below that these bits randomize x in the real world, so that $\mathcal{S}$—knowing neither $(x_i)_{i=0}^{l-1}$ nor $(x_i)_{i=l}^{l'-1}$—can nonetheless simulate x convincingly to D, who knows $(x_i)_{i=0}^{l-1}$ but not $(x_i)_{i=l}^{l'-1}$. Finally, though $\mathcal{S}$ doesn't know $(\mathbf{t}_i)_{i=0}^{l'-1}$ either, it does know $(\mathbf{q}_i)_{i=0}^{l'-1}$, x, Δ and $(\chi_i)_{i=0}^{l'-1}$; these values exhaustively determine t, from the respective perspectives both of $\mathcal{A}$ and of D.

Why sacrificing is needed. As a warmup, we note that the hypothetical variant of $\Pi_{\mathsf{ROT}}^{\kappa,l}$ which opted to set $l' \coloneqq l$ would be—though correct—insecure. To show this, we write $\chi : \mathbb{F}_2^l \to \mathbb{F}_2^\kappa$ for the $\mathbb{F}_2$-linear map defined by the $\kappa \times l$ matrix:

$$\begin{bmatrix} | & & | \\ \chi_0 & \cdots & \chi_{l-1} \\ | & & | \end{bmatrix};$$

here, we interpret each $\chi_i \in \mathbb{F}_{2^\kappa}$ as a column vector by identifying $\mathbb{F}_{2^\kappa} \cong \mathbb{F}_2^\kappa$. In this hypothetical protocol variant, the quantity $x \coloneqq \sum_{i=0}^{l-1} \chi_i \cdot x_i$ sent by R to S would equal the image of $(x_i)_{i=0}^{l-1}$ under χ. In particular, R's message $x = \chi \cdot (x_i)_{i=0}^{l-1}$ would leak information to S about $(x_i)_{i=0}^{l-1}$; that is, it would leak the membership of R's secret choice vector $(x_i)_{i=0}^{l-1}$ in the fiber $\chi^{-1}(x) \subset \mathbb{F}_2^l$ (a proper affine subspace of $\mathbb{F}_2^l$, except in the vanishingly improbable case in which $\chi_i = 0$ for each $i \in \{0, \ldots, l-1\}$). Equivalently, the simulator $\mathcal{S}$—who, we repeat, doesn't know $(x_i)_{i=0}^{l-1}$—would become unable to simulate x to $\mathcal{A}$, since it wouldn't know $(x_i)_{i=0}^{l-1}$'s image under χ. (If $\mathcal{S}$ simply randomly sampled $x \leftarrow \mathbb{F}_2^\kappa$, or better yet $x \leftarrow \mathrm{im}(\chi)$, then the distinguisher D—who does know $(x_i)_{i=0}^{l-1}$—could reliably identify the real-world distribution as that within which $x \stackrel{?}{=} \sum_{i=0}^{l-1} \chi_i \cdot x_i$ held.)

When $l' > l$ instead holds, R's quantity x rather satisfies $x = \chi \cdot (x_i)_{i=0}^{l-1} + \chi' \cdot (x_i)_{i=l}^{l'-1}$; here, we write $\chi' : \mathbb{F}_2^{l'-l} \to \mathbb{F}_2^\kappa$ for the further $\kappa \times l' - l$ matrix:

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

Since the real-world honest party R's high bits $(x_i)_{i=l}^{l'-1}$ are random and independent of its choice vector $(x_i)_{i=0}^{l-1}$, x's right-hand summand $\chi' \cdot (x_i)_{i=l}^{l'-1}$ masks its sensitive left-hand summand $\chi \cdot (x_i)_{i=0}^{l-1}$. More precisely, R's message x reveals to S merely that $(x_i)_{i=0}^{l-1}$ resides in $\chi^{-1}(x') \subset \mathbb{F}_2^\kappa$, for some unknown element $x' \in \mathrm{im}(\chi)$ which differs from x by a secret—and random—element of $\mathrm{im}(\chi')$. In particular, as soon as $\mathrm{im}(\chi) \subset \mathrm{im}(\chi')$ holds, x comes to reveal, information-theoretically, no information whatsoever about $(x_i)_{i=0}^{l-1}$; in this latter case, in the real world, x is uniform in $\mathrm{im}(\chi')$. In the further special case in which $\mathrm{im}(\chi') = \mathbb{F}_2^\kappa$ in fact holds, we conclude that x is uniform in $\mathbb{F}_2^\kappa$ in the real world. In our proof below, we argue that, if $l' \geq l$ is chosen high enough—in fact, $l' \coloneqq l + \kappa + s$ suffices—then, with high probability over $(\chi_i)_{i=l}^{l'-1}$, χ' is in fact surjective.

The proof. We proceed with our proof of security in the presence of a corrupt sender.

Theorem 3.1. *In the $\mathcal{F}_{\mathsf{Rand}}^{\kappa}, \mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$-hybrid model, Protocol 2.7 securely computes Functionality 2.5 in the presence of a corrupt sender.*

Proof. We define an appropriate simulator $\mathcal{S}$. Given a real-world adversary $\mathcal{A}$ corrupting S, $\mathcal{S}$ operates as follows.

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

We assert the suitability of $\mathcal{S}$ in the following way. For each $i \in \{0, \ldots, l-1\}$, R's real-world output equals $\mathbf{v}_{i,x_i} \coloneqq H(i \,\|\, \mathbf{t}_i) = H(i \,\|\, \mathbf{q}_i + \mathbf{x}_i \cdot \Delta) = H(i \,\|\, \mathbf{q}_i + x_i \cdot \Delta)$. On the other hand, by construction, $\mathcal{S}$'s ideal-world inputs respectively equal $\mathbf{v}_{i,0} \coloneqq H(i \,\|\, \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \,\|\, \mathbf{q}_i + \Delta)$. We conclude that R's ideal-world output too equals $\mathbf{v}_{i,x_i} = H(i \,\|\, \mathbf{q}_i + x_i \cdot \Delta)$, as required.

We turn to the view simulated by $\mathcal{S}$ to $\mathcal{A}$. The following lemma appears as [KOS15, Lem. 2]; for self-containedness, we include a proof.

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

Proof. The probability that χ''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 $\sum_{i=0}^{\kappa-1} 2^{-s-1-i}$. □

Lemma 3.2 shows that, in both the real and ideal worlds, χ' fails to be surjective with probability at most $\frac{1}{2^s}$. On the other hand, if χ' is surjective, then $\mathcal{S}$ sampling strategy—whereby it picks $x \leftarrow \mathbb{F}_2^{\kappa}$ uniformly—perfectly matches x's real-world distribution.

We turn finally to R's quantity t. Though $\mathcal{S}$ doesn't know $(x_i)_{i=0}^{l'-1}$ or R's $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$-responses $(\mathbf{t}_i)_{i=0}^{l'-1}$, it does know $\mathcal{A}$'s local quantities Δ and $(\mathbf{q}_i)_{i=0}^{l'-1}$. Since each honest receiver R's quantities x and t necessarily cause S's correlation check $q \stackrel{?}{=} t + x \cdot \Delta$ to pass, $\mathcal{S}$ assignment strategy $t \coloneqq q + x \cdot \Delta$ exactly characterizes t, conditioned on $\mathcal{A}$'s available data Δ,$(\mathbf{q}_i)_{i=0}^{l'-1}$, and x. This completes our treatment of the corrupt sender. □

Remark 3.3. As the above proof makes evident, a slightly subtler simulation strategy would stipulate that $\mathcal{S}$ instead sample $x \leftarrow \mathrm{im}(\chi')$. This strategy would yield a perfectly correct simulation even in those rare executions in which, though $\mathrm{im}(\chi') \subsetneq \mathbb{F}_2^{\kappa}$, $\mathrm{im}(\chi) \subset \mathrm{im}(\chi')$ nonetheless holds. Since the strict inclusion $\mathrm{im}(\chi') \subsetneq \mathbb{F}_2^{\kappa}$ itself occurs only with negligible chance, this scenario isn't worth accounting for. Of course, when $\mathrm{im}(\chi) \not\subset \mathrm{im}(\chi')$ instead holds, $\mathcal{S}$ can proceed at best only by "guessing" $(x_i)_{i=0}^{l-1}$ and setting $x \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot x_i$, for $(x_i)_{i=l}^{l'-1}$ random. The distinguisher D—given a transcript created in this way—may check whether $x - \sum_{i=0}^{l-1} \chi_i \cdot x_i \stackrel{?}{\in} \mathrm{im}(\chi')$. This inclusion will hold for each real-world execution, but will hold in the ideal world only when $\mathcal{S}$ manages to guess a string $(x_i)_{i=0}^{l-1}$ for which $\sum_{i=0}^{l-1} \chi_i \cdot x_i$ differs from its true value by an element of $\mathrm{im}(\chi) \cap \mathrm{im}(\chi')$, an $\mathbb{F}_2$-linear subspace of $\mathrm{im}(\chi)$ which is proper whenever $\mathrm{im}(\chi) \not\subset \mathrm{im}(\chi')$.

3.2 Corrupt Receiver

We now handle the case in which the receiver R is corrupt.

The simulator's mandate. We informally explain $\mathcal{S}$'s mandate in the following way. By generating its own secret choice vector $\Delta \leftarrow \mathbb{F}_2^\kappa$ and mimicking the role of S and of $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ internally to $\mathcal{A}$, $\mathcal{S}$ may simulate a view to $\mathcal{A}$ which perfectly matches $\mathcal{A}$'s real-world view. $\mathcal{S}$ must then, further, extract from $\mathcal{A}$ input choice bits $(x_i)_{i=0}^{l-1}$, as well as pads $(\mathbf{v}_{i,x_i})_{i=0}^{l-1}$, for which S's resulting ideal outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$ become indistinguishable from their real-world counterparts. In the real world, S, for each $i \in \{0, \ldots, l-1\}$, simply computes $\mathbf{v}_{i,0} \coloneqq H(i \parallel \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \parallel \mathbf{q}_i + \Delta)$; these quantities themselves respectively equal $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$ (this is an immediate consequence of the definition of $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, and of the fact that $\mathbf{q}_i + \Delta = \mathbf{t}_i + \mathbf{x}_i * \Delta + \Delta = \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta$). Since $\mathcal{S}$ has $\mathcal{A}$'s vectors $(\mathbf{x}_i)_{i=0}^{l'-1}$ and $(\mathbf{t}_i)_{i=0}^{l'-1}$, $\mathcal{S}$ may easily run $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ in its own head, thereby anticipating S's hypothetical real-world outputs $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$. $\mathcal{S}$'s critical obstacle is that, for each $i \in \{0, \ldots, l-1\}$, it may send only one choice bit x_i and one output pad $\mathbf{v}_{i,x_i}$ to the functionality $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$, which, for its part, demands the right to sample the other pad $\mathbf{v}_{i,\overline{x_i}} \leftarrow \{0,1\}^\kappa$ randomly. (If $\mathcal{S}$ could supply both, then it could generate a perfect simulation.) $\mathcal{S}$ thus must opt instead to compute either $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ or $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$ exactly as S would, and to relinquish control of the other quantity to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$.

The distinguisher D, finally, sees $\mathcal{A}$'s full internal view, and in particular knows $(\mathbf{t}_i)_{i=0}^{l'-1}$, $(\mathbf{x}_i)_{i=0}^{l'-1}$, and $(x_i)_{i=0}^{l-1}$. Moreover, if the correlation check passes, then D also obtains the honest sender's outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$. Critically, D doesn't know the sender's (or simulator's) choice vector $\Delta \in \mathbb{F}_2^\kappa$, a secret and internal value. For each $i \in \{0, \ldots, l-1\}$, D may attempt to compute either or both of the quantities $H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$. In general, however, one of these will be vastly easier for D to compute than the other will. That is, among the vectors $\mathbf{x}_i$ and $\overline{\mathbf{x}_i}$, one will generally have lower Hamming weight. The Hamming weights of $\mathbf{x}_i$ and of $\overline{\mathbf{x}_i}$ respectively control the computational costs to D of computing $H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$. In these respective cases, D—which, we repeat, knows all of the quantities at hand except Δ—must submit $2^{w(\mathbf{x}_i)}$ and $2^{w(\overline{\mathbf{x}_i})}$ queries to the random oracle, in the worst case, in order to be assured of "hitting" the relevant query. (The unknown quantities $\mathbf{x}_i * \Delta$ and $\overline{\mathbf{x}_i} * \Delta$ capture the "restrictions" or "projections" of the unknown vector Δ onto the bit-positions which are respectively on in $\mathbf{x}_i$ and in $\overline{\mathbf{x}_i}$; there are $2^{w(\mathbf{x}_i)}$ and $2^{w(\overline{\mathbf{x}_i})}$ possible such projections.)

The simulator $\mathcal{S}$, then—which gets to ensure only one of the equalities $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$—should choose $x_i \in \{0,1\}$ in such a way that $\mathbf{v}_{i,x_i}$ is easy for D to query and $\mathbf{v}_{i,\overline{x_i}}$ is hard for D to query. To express this phenomenon in a more notationally convenient form, we abbreviate $\mathbf{e}_i \coloneqq x_i \cdot (1, \ldots, 1) + \mathbf{x}_i$ and $\overline{\mathbf{e}_i} \coloneqq \overline{x_i} \cdot (1, \ldots, 1) + \mathbf{x}_i$, for each $i \in \{0, \ldots, l-1\}$; here, we write $(x_i)_{i=0}^{l-1}$ for the bits extracted by $\mathcal{S}$. We moreover call these vectors *on-vectors* and *off-vectors*, respectively. In the real world, both $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$ hold. On the other hand, in the ideal world—and provided $\mathcal{S}$ submits the choice vector $(x_i)_{i=0}^{l-1}$ to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$—then only $\mathbf{v}_{i,x_i} = H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta)$ holds, while $\mathbf{v}_{i,\overline{x_i}}$ is random. We conclude that D can win if and only if it manages to query $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$. $\mathcal{S}$'s mandate is to choose the bits $(x_i)_{i=0}^{l-1}$ in such a way as to make this task as hard as possible for D.

The distinguisher's attack strategy. To build intuition, we begin with the simplest possible case. If R is honest, then each of its vectors $\mathbf{x}_i = x_i \cdot (1, \ldots, 1)$ is monochromatic. It seems obvious that $\mathcal{S}$ should extract from each monochromatic vector $x_i \cdot (1, \ldots, 1)$ the bit x_i. The discussion above explains this phenomenon. If $\mathcal{S}$ were to submit the "wrong" bit, then we would have $\mathbf{e}_i = (1, \ldots, 1)$ and $\overline{\mathbf{e}_i} = (0, \ldots, 0)$. We'd thus obtain the equality $\mathbf{v}_{i,x_i} = H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta) = H(i \parallel \mathbf{t}_i + \Delta)$ in both worlds; on the other hand, $\mathbf{v}_{i,\overline{x_i}} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta) = H(i \parallel \mathbf{t}_i)$ would hold only in the real world. By making just a single query, D could thus check $\mathbf{v}_{i,\overline{x_i}} \stackrel{?}{=} H(i \parallel \mathbf{t}_i)$, thereby distinguishing the two worlds.

A similar phenomenon applies to those vectors $\mathbf{x}_i$ which are "mostly monochromatic", in that most of their bits equal $x_i \in \{0,1\}$, say. In this case, $\mathcal{S}$ should again extract the bit $x_i \in \{0,1\}$ which agrees with most of $\mathbf{x}_i$'s components. If it didn't, then $\overline{\mathbf{e}_i}$ would become of low Hamming weight. The distinguisher D could, then, brute-force all $2^{w(\overline{\mathbf{e}_i})}$ queries $\mathbf{v}_{i,\overline{x_i}} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$. Upon finding a match—or else failing to—D could once again detect whether the transcript represented a real-world or an ideal-world execution.

We thus see, as a first conclusion, that $\mathcal{S}$ should heuristically choose each x_i so that $\overline{\mathbf{e}_i}$ has high Hamming weight, for example by setting $x_i := \textsc{Maj}(\mathbf{x}_i)$; this measure would guarantee that each $i \in \{0, \ldots, l-1\}$ satisfied $w(\overline{\mathbf{e}_i}) \geq \frac{\kappa}{2}$. (This phenomenon reappears in Remark 3.23 below.)

However, D has a further, subtler attack strategy, as we now explain. D may also test on-vectors, by brute-forcing all $2^{w(\mathbf{e}_i)}$ queries of the form $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta)$, for some $i \in \{0, \ldots, l-1\}$. Indeed, in both worlds, this equality holds precisely when D queries the "right" projection $\mathbf{e}_i * \Delta$. If $\mathbf{e}_i$ is of sufficiently low Hamming weight, then D may in this way learn the projection $\mathbf{e}_i * \Delta$ (i.e., the restriction of Δ to the bit-positions on in $\mathbf{e}_i$). Of course, this feat alone doesn't solve D's problem. On the other hand, D may attempt to repeat this process across various $i \in \{0, \ldots, l-1\}$, accumulating knowledge all the while. Importantly, once D has learned the projection $\mathbf{e}_i * \Delta$ for some particular $i \in \{0, \ldots, l-1\}$, it may "fix" in place the corresponding bits of Δ during each of its subsequent executions of the same brute-force procedure. Thus, the complexity of learning some further projection $\mathbf{e}_{i'} * \Delta$ is not $2^{w(\mathbf{e}_{i'})}$, but rather $2^{w(\mathbf{e}_{i'} \wedge \neg \mathbf{e}_i)}$ (i.e., it depends only on the number of new on-bits of $\mathbf{e}_{i'}$). By proceeding in this way, D may attempt to learn so many bits of Δ that an off-vector $\overline{\mathbf{e}_i}$ comes to fall within reach (though its Hamming weight be large). At this point, D may try as before to brute-force the off-vector $\mathbf{v}_{i,\overline{x_i}} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$.

D thus has a complicated—and path-dependent—space of attack strategies. Roughly, D may approach in arbitrary order the various on-vectors $\mathbf{e}_i$, for $i \in \{0, \ldots, l-1\}$—seeking, in each case, to minimize the number of new on-bits—and performing repeatedly the brute-force search $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta)$. D, in this way, might attempt to learn so many bits of Δ that it becomes capable of brute-forcing an off-vector. $\mathcal{S}$'s best hope, then, is more subtle than simply maximizing the Hamming weight of the vectors $\overline{\mathbf{e}_i}$. Rather, $\mathcal{S}$ must seek to lengthen the "chain" of segments which D must brute-force, if it is to reach an off-vector.

The correlation check. As the above explanation makes clear, the adversary $\mathcal{A}$—by submitting pathologically formed matrices $(\mathbf{x}_i)_{i=0}^{l'-1}$—may attempt to aid the distinguisher. Specifically, $\mathcal{A}$ may elect to submit matrices which contain multiple short segments of on-bits, hoping, in this way, to build a traversable "bridge"—culminating in an off-vector—for future use by the distinguisher.

On the other hand, $\mathcal{A}$ must also pass its correlation check. In each execution in which $\mathcal{A}$'s quantities x and t fail the sender's correlation check $q \stackrel{?}{=} t + x \cdot \Delta$, either the honest party (in the real world) or the simulator (in the ideal world), as the case may be, aborts. $\mathcal{A}$'s failure to pass this check, then, has the effect of depriving the distinguisher D of its most critical resource, the honest party's outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$. Indeed, if $\mathcal{A}$ fails the correlation check, then the simulation becomes perfectly secure (regardless of which bits $(x_i)_{i=0}^{l-1}$ $\mathcal{S}$ extracts and of which pads it submits).

$\mathcal{A}$, therefore, is caught between conflicting aims. On the one hand, it must submit a rather pathological matrix, in order to aid the distinguisher. On the other hand, it must pass the correlation check, if is to vouchsafe the distinguisher any chance at all. As we argue below, these aims are at odds. Indeed, as $\mathcal{A}$'s matrix becomes more pathological, $\mathcal{A}$'s correlation check simultaneously becomes harder to pass.

Modesty. We begin the proof with one of this paper's main contributions. As it turns out, we are able to write down a measure of "compliance"—which assigns to each matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$ submitted by $\mathcal{A}$ a numerical score $m \in \{1, \ldots, \kappa\}$—which simultaneously captures both of $\mathcal{A}$'s conflicting aims. We define this metric, which we call $\mathcal{A}$'s matrix's *modesty*, in Definition 3.4 below. Indeed, as $(\mathbf{x}_i)_{i=0}^{l'-1}$'s modesty decreases, $\mathcal{A}$'s correlation check becomes steadily harder to pass, a fact we formalize in Proposition 3.15 below. On the other hand, as $(\mathbf{x}_i)_{i=0}^{l'-1}$'s modesty increases, D must work progressively harder to distinguish the two worlds, as we show in Proposition 3.18. In fact, these aims work so strongly against each other that—as it turns out—D's chance of succeeding grows at best negligibly in κ, regardless of how modestly $\mathcal{A}$ constructs its matrix. This fact underlies our main security result, which appears as Theorem 3.14 below.

For now, we rigorously define the modesty of $(\mathbf{x}_i)_{i=0}^{l'-1}$, or rather of its upper part $(\mathbf{x}_i)_{i=0}^{l-1}$ (by submitting a non-monochromatic lower submatrix $(\mathbf{x}_i)_{i=l}^{l'-1}$, $\mathcal{A}$ can only hurt itself, and can't help D). In Definition 3.4—and throughout our entire proof below—we identify vectors $\mathbf{d} \in \mathbb{F}_2^\kappa$ with subsets $\mathbf{d} \subset \{0, \ldots, \kappa-1\}$, by associating to each vector its set of on-positions in $\{0, \ldots, \kappa-1\}$ (see also Section 2). The *cardinality* $|\mathbf{d}|$ of a vector is nothing other than its Hamming weight $w(\mathbf{d})$. The *union* of vectors $\mathbf{d} \cup \mathbf{e}_i$ is nothing other than the bitwise OR $\mathbf{d} \vee \mathbf{e}_i$. Finally, the *set difference* $\mathbf{x}_i \setminus \mathbf{d}$ is $\mathbf{x}_i \wedge \neg \mathbf{d}$; its cardinality $|\mathbf{x}_i \setminus \mathbf{d}|$ is $w(\mathbf{x}_i \wedge \neg \mathbf{d})$.

Definition 3.4. The *modesty* of $(\mathbf{x}_i)_{i=0}^{l-1}$ is the largest $m \in \{1, \dots, \kappa\}$ 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 $\mathbf{d} \coloneqq \varnothing$, initialize $(x_i)_{i=0}^{l-1}$ arbitrarily, and mark each row-index $i \in \{0, \dots, l-1\}$ white.
3: **for** l repetitions **do**
4: **for** $i \in \{0, \dots, l-1\}$ **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, writing $\mathbf{e}_i \coloneqq x_i \cdot (1, \dots, 1) + \mathbf{x}_i$ we get $|\mathbf{e}_i \setminus \mathbf{d}| < m$.
7: **if** $\mathbf{e}_i \subset \mathbf{d}$ **then** mark the index $i \in \{0, \dots, l-1\}$ grey.
8: **else** update $\mathbf{d} \cup\!= \mathbf{e}_i$ and mark the index $i \in \{0, \dots, l-1\}$ black.
9: **if** $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| < m$ **then return** true.
10: **break** the inner loop 4.
11: **return** false.

We note that, though only the white–non-white distinction figures explicitly within Definition 3.4, the grey–black distinction itself becomes important throughout our proof below.

Example 3.5. We note first that, for $(\mathbf{x}_i)_{i=0}^{l-1}$ arbitrary, $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, 1\Big) = \mathsf{false}$ necessarily holds. Indeed, we claim by induction that, during the course of that algorithm, $\mathbf{d} = \varnothing$ will hold throughout. That algorithm's condition 5 will hold only when at least one of the cardinalities $|\mathbf{x}_i \setminus \mathbf{d}|$ or $|\overline{\mathbf{x}_i} \setminus \mathbf{d}|$ equals $0 < 1$. Under our hypothesis $\mathbf{d} = \varnothing$, this latter condition entails precisely that $\mathbf{x}_i$ is monochromatic. We conclude that the algorithm will reach line 6 only on monochromatic $\mathbf{x}_i$, at which point it will set $x_i \in \{0, 1\}$ so that $\mathbf{e}_i = (0, \dots, 0)$ and $\overline{\mathbf{e}_i} = (1, \dots, 1)$ both hold. On each execution of line 7, we will thus have $\mathbf{e}_i = \varnothing$, so that i will be marked grey, and our inductive hypothesis $\mathbf{d} = \varnothing$ will be preserved. Finally, in each execution of line 9, we will have $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| = \kappa \geq 1$, so that the algorithm won't exit. We conclude that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, 1\Big)$ will mark each monochromatic row grey and leave the rest of the rows white, and finally will return false.

In particular, Example 3.5 shows that the modesty $m \in \{1, \dots, \kappa\}$ of each matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ is well-defined.

Example 3.6. We examine the operation of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ in the monochromatic case, which represents the behavior of an honest receiver. Indeed, we claim that the modesty of each monochromatic matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ is κ. It suffices to show that, for $(\mathbf{x}_i)_{i=0}^{l-1}$ monochromatic and $m \in \{1, \dots, \kappa\}$ arbitrary, $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$. We proceed essentially as in Example 3.5. For each execution of the test 5, $|\mathbf{x}_i \setminus \mathbf{d}|$ and $|\overline{\mathbf{x}_i} \setminus \mathbf{d}|$ will equal 0 and κ in some order; in particular, the condition 5 will be fulfilled. In line 6, the algorithm will choose x_i so that $\mathbf{e}_i = (0, \dots, 0)$ and $\overline{\mathbf{e}_i} = (1, \dots, 1)$ both hold. On line 7, we will have $\mathbf{e}_i = \varnothing$, so that $\mathbf{e}_i \subset \mathbf{d}$ will hold and i will be marked grey. By induction, we see again that $\mathbf{d} = \varnothing$ will hold throughout. Finally, in line 9, we will have $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| = \kappa \geq m$, so that the condition 9 will fail to be fulfilled. After marking each vector $\mathbf{e}_i$ grey in this way, $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ will finally return false on line 11.

Example 3.7. We examine the execution of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ on the matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ consisting of a $\kappa \times \kappa$ identity submatrix $(\mathbf{x}_i)_{i=0}^{\kappa-1}$, followed by a number of identically zero rows $(\mathbf{x}_i)_{i=\kappa}^{l-1}$ (we assume here that $l \geq \kappa$). In a sense, this matrix is as "pathological" as possible. We claim that its modesty is 1. By Example 3.5, we know already that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, 1\Big) = \mathsf{false}$. We argue that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$ for each $m > 1$. To show this, we fix a candidate modesty $m \in \{2, \dots, \kappa\}$. We note first that the algorithm, upon considering each successive row indexed $i \in \{0, \dots, \kappa - m\}$, will find that $|\mathbf{x}_i \setminus \mathbf{d}| = 1 < m$, and so will set $x_i \coloneqq 0$, mark i black, and update $\mathbf{d} \cup\!= \mathbf{e}_i$. By induction, we conclude that, for each $i \in \{0, \dots, \kappa - m\}$, in the i^{th} iteration of the main outer loop, immediately following the update step 8, we will have $\mathbf{d} = \{0, \dots, i\}$. As for line 9, in the i^{th} iteration of the main outer loop, we will thus have in turn that $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| = |\{0, \dots, i-1, i+1, \dots, \kappa-1\} \setminus \{0, \dots, i\}| = |\{i+1, \dots, \kappa-1\}| = \kappa - i - 1$, which is at least m so long as $i \in \{0, \dots, \kappa - m - 1\}$. During the $\kappa - m^{\text{th}}$ iteration, on the other hand, $|\overline{\mathbf{e}_{\kappa-m}} \setminus \mathbf{d}| = |\{\kappa - m + 1, \dots, \kappa - 1\}| = m - 1 < m$ will hold, so that the algorithm will finally return true.

As we will see in Example 3.22 below, if $\mathcal{A}$ manages to pass the correlation check despite having submitted the matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ of Example 3.7, then D will obtain an immediate $O(\kappa)$-time attack (even under our carefully chosen extraction strategy). The point, of course, is that, upon submitting this matrix, $\mathcal{A}$'s chance of passing will become vanishingly small.

We now examine more formally the behavior of Definition 3.4. In the following two lemmas, we fix arbitrary inputs $(\mathbf{x}_i)_{i=0}^{l-1}$ and $m \in \{1, \ldots, \kappa\}$. We moreover write $\mathbf{d} \subset \{0, \ldots, \kappa - 1\}$ and $(x_i)_{i=0}^{l-1}$ for the respective states ultimately taken by these values as of the end of the execution of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. We give meaning to the vectors $(\mathbf{e}_i)_{i=0}^{l-1}$ using the bit assignment $(x_i)_{i=0}^{l-1}$.

Lemma 3.8. *If* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$*, then, for each* white *index* $i \in \{0, \ldots, l-1\}$*, we have both* $|\mathbf{x}_i \setminus \mathbf{d}| \geq m$ *and* $|\overline{\mathbf{x}_i} \setminus \mathbf{d}| \geq m$*.*

Proof. Each iteration of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$'s outer loop 3 either marks exactly one row non-white, or else does nothing (in which case each further iteration also does nothing). On the other hand, our assumption whereby $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$ implies that the algorithm actually ran for all l iterations. Combining these facts, we see that the mere existence of white rows as of the end of the algorithm's execution implies that the outer loop 3's last iteration did nothing. Since each iteration of the outer loop which does nothing subjects to each row $i \in \{0, \ldots, l-1\}$ the test 5, we conclude that each row $i \in \{0, \ldots, l-1\}$ which remains white as of the algorithm's end was subjected to that condition during the algorithm's last iteration (and of course failed to fulfill it, or else it would have been marked non-white). We conclude that—again for each $i \in \{0, \ldots, l-1\}$ which remains white to the end—both $|\mathbf{x}_i \setminus \mathbf{d}| \geq m$ and $|\overline{\mathbf{x}_i} \setminus \mathbf{d}| \geq m$ hold, as required. □

Lemma 3.9. *If* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$*, then, for each index* $i \in \{0, \ldots, l-1\}$*, we have* $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq m$*.*

Proof. For each index $i \in \{0, \ldots, l-1\}$ which remains white as of the algorithm's termination, Lemma 3.8 immediately implies the lemma's conclusion (and, in fact, that $|\mathbf{e}_i \setminus \mathbf{d}| \geq m$ and $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq m$ both hold). We thus fix an arbitrary non-white row $i \in \{0, \ldots, l-1\}$; we moreover 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 $\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d} \subset \overline{\mathbf{e}_i} \setminus \mathbf{d}$ in fact holds. If $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| < m$ held, then we'd thus conclude in turn that $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$ also held, and hence that the escape condition 9 was fulfilled just after $\mathbf{e}_{i^*}$ was marked non-white. This would imply that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, contradicting the lemma's hypothesis. □

Remark 3.10. Lemma 3.9 suggests the following further interpretation of the algorithm $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. Indeed, that lemma shows that it is an algorithmic invariant of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ that either every non-white row $i \in \{0, \ldots, l-1\}$ simultaneously satisfies $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| < m$ or else no non-white row does. To show this, we fix a non-white index $i^* \in \{0, \ldots, l-1\}$ and consider the state of $\mathbf{d}$ immediately after the update step $\mathbf{d} \cup= \mathbf{e}_{i^*}$. Certainly, $\mathbf{e}_{i^*} \subset \mathbf{d}$ holds. If $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < m$ moreover holds—that is, if the condition 9 is fulfilled—then, since $\mathbf{e}_{i^*}$ and $\overline{\mathbf{e}_{i^*}}$ partition $\{0, \ldots, \kappa - 1\}$, we see that $|\{0, \ldots, \kappa - 1\} \setminus \mathbf{d}| = |\mathbf{e}_{i^*} \setminus \mathbf{d}| + |\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| < 0 + m$, so that each $i \in \{0, \ldots, l-1\}$ (in fact regardless of color) satisfies $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \leq |\{0, \ldots, \kappa - 1\} \setminus \mathbf{d}| < m$. Conversely, if $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| \geq m$ instead holds, then the proof of Lemma 3.9 shows that each row $i \in \{0, \ldots, l-1\}$ marked non-white before i^* in fact satisfies $\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d} \subset \overline{\mathbf{e}_i} \setminus \mathbf{d}$, so that $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq |\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| \geq m$ in turn necessarily holds. This fact is surprising; it seems *a priori* conceivable that the set $\mathbf{d}$, upon being made to include the contents of some set $\mathbf{e}_{i^*}$, could—though $|\overline{\mathbf{e}_{i^*}} \setminus \mathbf{d}| \geq m$—nonetheless cause the inequality $|\overline{\mathbf{e}_i} \setminus \mathbf{d}| < m$ to hold, for some other $i \in \{0, \ldots, l-1\}$ previously marked non-white by the algorithm. The proof of Lemma 3.9 shows that this can't happen.

Were Lemma 3.9 unproven, or even false, we could apparently compensate, at least for the purposes of our proof below, by appending to the routine of Definition 3.4 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). More generally, it would leave unaccounted for an important aspect of the operation of the algorithm, and hence of our proof below.

The following results are, as it turns out, not necessary to establish the proof of our main result. Nonetheless, they collectively establish the "well-behavedness" of Definition 3.4. We defer their proofs to Appendix A.

Lemma 3.11. *For each* $(\mathbf{x}_i)_{i=0}^{l-1}$ *and each* $m \in \{1, \ldots, \kappa\}$*, the boolean return value of* $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ *is independent of the order in which the indices* $i \in \{0, \ldots, l-1\}$ *are tested by that algorithm's inner loop 4.*

Proof. Deferred to Appendix A. □

Exploiting Lemma 3.11, we obtain a simple proof of the following result. It is interesting that the easiest proof of this result seems to be that—given below—which proceeds via the aid of the more-complicated Lemma 3.11. Though a direct proof would be interesting, and is probably possible, we have restricted ourselves, for the sake of brevity, to the approach presented below.

Corollary 3.12. *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* $\{1, \ldots, \kappa\}$*; i.e., 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. Deferred to Appendix A. □

Remark 3.13. We record without proof the following further consequences of Lemma 3.11, all valid only when $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$, which we presently assume. That result, for one, implies that the vector $\mathbf{d} \subset \{0, \ldots, \kappa-1\}$ ultimately constructed during the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ is likewise independent of the order in which that algorithm treats its rows. Finally, Lemma 3.11 implies that the subset consisting of those rows $i \in \{0, \ldots, l-1\}$ marked non-white during the algorithm is likewise order-independent, as is, for each non-white row $i \in \{0, \ldots, l-1\}$, the bit assignment x_i. If $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, then none of these quantities are, in general, independent of the algorithm's row-treatment order.

The proof. We now proceed with our main security theorem.

Theorem 3.14. *In the* $\mathcal{F}_{\mathsf{Rand}}^{\kappa}, \mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$*-hybrid model, Protocol 2.7 securely computes Functionality 2.5 in the presence of a corrupt receiver.*

Proof. We define our simulator $\mathcal{S}$. 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 messages $\Big(\mathtt{input}, (\mathbf{x}_i)_{i=0}^{l'-1}\Big)$ and $\Big(\mathtt{pads}, (\mathbf{t}_i)_{i=0}^{l'-1}\Big)$ intended for $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$, $\mathcal{S}$ 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 $(\mathbf{x}_i)_{i=0}^{l-1}$. For each $i \in \{0, \ldots, l-1\}$, $\mathcal{S}$ moreover writes $\mathbf{q}_i \coloneqq \mathbf{t}_i + \mathbf{x}_i * \Delta$, and sets $\mathbf{v}_{i,x_i} \coloneqq H(i \parallel \mathbf{q}_i + x_i \cdot \Delta)$. $\mathcal{S}$ submits $\Big(\mathtt{input}, (x_i)_{i=0}^{l-1}\Big)$ and $\Big(\mathtt{pads}, (\mathbf{v}_{i,x_i})_{i=0}^{l-1}\Big)$ 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 $q \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{q}_i$, 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 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.

The view simulated by $\mathcal{S}$ to $\mathcal{A}$ identically matches $\mathcal{A}$'s real-world view. In particular, $\mathcal{S}$ simulates the machines S and $\mathcal{F}_{\mathsf{COTe}}^{\kappa,l'}$ perfectly, and runs the same correlation check $q \stackrel{?}{=} t + x \cdot \Delta$ that S does.

We turn to the output received by the honest party. If the correlation check fails, then the sender (in the real world) and the simulator (in the ideal world) both abort. If this happens, then the honest party (real or ideal) receives no output at all, and the simulation is perfect. Otherwise, for each $i \in \{0, \ldots, l-1\}$, S in the real world outputs $\mathbf{v}_{i,0} \coloneqq H(i \parallel \mathbf{q}_i)$ and $\mathbf{v}_{i,1} \coloneqq H(i \parallel \mathbf{q}_i + \Delta)$; these quantities respectively equal $\mathbf{v}_{i,0} = H(i \parallel \mathbf{t}_i + \mathbf{x}_i * \Delta)$ and $\mathbf{v}_{i,1} = H(i \parallel \mathbf{t}_i + \overline{\mathbf{x}_i} * \Delta)$ (as was already explained above). The simulator $\mathcal{S}$ may itself calculate both of these quantities; on the other hand, it must choose just one bit $x_i \in \{0,1\}$ and one pad $\mathbf{v}_{i,x_i}$ to submit to $\mathcal{F}_{\mathsf{ROT}}^{\kappa,l}$ (which insists on sampling $\mathbf{v}_{i,\overline{x_i}} \leftarrow \{0,1\}^\kappa$ randomly). In the ideal world, therefore, for each $i \in \{0, \ldots, l-1\}$, S outputs $\mathbf{v}_{i,x_i} = H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta)$ exactly as in the real world, while its output $\mathbf{v}_{i,\overline{x_i}} \leftarrow \{0,1\}^\kappa$ is independently random. If D never queries $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$, then the real and ideal distributions are identical. Here, we again write $\mathbf{e}_i \coloneqq x_i \cdot (1, \ldots, 1) + \mathbf{x}_i$ and $\overline{\mathbf{e}_i} \coloneqq \overline{x_i} \cdot (1, \ldots, 1) + \mathbf{x}_i$ for each $i \in \{0, \ldots, l-1\}$, where the bits $(x_i)_{i=0}^{l-1}$ are as extracted by $\mathcal{S}$.

We introduce a convenient notational device, which captures the adequacy of $\mathcal{A}$'s messages x and t to S. We fix $\mathcal{A}$'s local quantities $(\mathbf{x}_i)_{i=0}^{l'-1}$ and $(\mathbf{t}_i)_{i=0}^{l'-1}$; we moreover fix the elements $(\chi_i)_{i=0}^{l'-1}$ of $\mathbb{F}_{2^\kappa}$, which we view for now as fixed constants. For each pair of further elements x and t of $\mathbb{F}_{2^\kappa}$, we define the map $F_{x,t} : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\kappa$ as follows:

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

Since $\mathbf{q}_i = \mathbf{t}_i + \mathbf{x}_i * \Delta$ holds for each $i \in \{0, \ldots, l'-1\}$, this map exactly reflects the correlation check which the sender applies to its secret correlation vector $\Delta \in \mathbb{F}_2^\kappa$. In other words, its check passes if and only if $\mathcal{A}$ sends elements x and t which cause $F_{x,t}(\Delta) \stackrel{?}{=} 0$ to hold.

Clearly, the map $F_{x,t} : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\kappa$ is affine $\mathbb{F}_2$-linear, for each fixed $(\chi_i)_{i=0}^{l'-1}$ and for each x and t. 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, for each fixed candidate $x \in \mathbb{F}_{2^\kappa}$ and for $t \in \mathbb{F}_{2^\kappa}$ varying, the resulting maps $(F_{x,t})_{t \in \mathbb{F}_{2^\kappa}}$ are all affine translates of each other. For those $t \notin \mathrm{im}(F_{x,0})$, we have that $0 \notin \mathrm{im}(F_{x,t})$, so that the null set $\{\Delta \in \mathbb{F}_2^\kappa \mid F_{x,t}(\Delta) = 0\}$ is empty and the correlation check is guaranteed to fail (i.e., regardless of $\Delta \in \mathbb{F}_2^\kappa$). We may thus ignore these candidates t. On the other hand, for $t \in \mathrm{im}(F_{x,0})$ varying, the resulting null sets $\{\Delta \in \mathbb{F}_2^\kappa \mid F_{x,t}(\Delta) = 0\}$ yield a family of parallel affine subspaces in $\mathbb{F}_2^\kappa$ of identical dimension. Since our below arguments depend only on the dimension of the affine subspace $\{\Delta \in \mathbb{F}_2^\kappa \mid F_{x,t}(\Delta) = 0\}$ and not on its contents, we may freely fix $t \in \mathrm{im}(F_{x,0})$ arbitrarily throughout the course of our treatment below (i.e., for each candidate $x \in \mathbb{F}_{2^\kappa}$). We thus set $t \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot \mathbf{t}_i$, the image of 0 itself under $F_{x,0}$. (This choice $t \in \mathbb{F}_{2^\kappa}$ is in fact *a posteriori* independent of $x \in \mathbb{F}_{2^\kappa}$.) For each $x \in \mathbb{F}_{2^\kappa}$, the resulting map $F_{x,t} : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\kappa$ is in fact linear. We write below the map which results from this simplifying assumption, in which all affine constants are omitted:

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

In particular, we refer to $\mathrm{rank}(F'_x)$ and $\ker(F'_x)$ throughout.

We note that those $x \in \mathbb{F}_{2^\kappa}$ for which $\mathrm{rank}(F'_x)$ is small yield larger kernels $\ker(F'_x)$, which in turn make $\mathcal{A}$'s correlation check easier to pass (over the secret and random vector Δ). In particular, $\mathcal{A}$'s goal is to submit an element $x \in \mathbb{F}_{2^\kappa}$ which makes $\mathrm{rank}(F'_x)$ is as small as possible. We denote by r the minimal rank achieved across all maps $(F'_x)_{x \in \mathbb{F}_{2^\kappa}}$; in other words:

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

The minimal rank r depends exhaustively on $(\mathbf{x}_i)_{i=0}^{l'-1}$ and on the random coefficients $(\chi_i)_{i=0}^{l'-1}$. In our proof below, we understand the minimal rank r as a function of the random mixing scalars $(\chi_i)_{i=0}^{l'-1}$ (i.e., for $(\mathbf{x}_i)_{i=0}^{l'-1}$ implicitly fixed). In other words, for each list $(\chi_i)_{i=0}^{l'-1}$ sampled by $\mathcal{F}_{\mathsf{Rand}}^\kappa$ (in the real world) or $\mathcal{S}$ (in the ideal world), we obtain a resulting minimum rank r; in this way, we view r as a random variable (i.e., depending on the random quantities $(\chi_i)_{i=0}^{l'-1}$).

We 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 1 below, which should be understood as a probability tree, in which each edge represents a conditional probability. The distinguishing advantage of any distinguisher D is given by a recursive traversal of this tree, in which the probability of each node is calculated as the weighted average of those of its children (where each child's probability is multiplied by its edge weight).

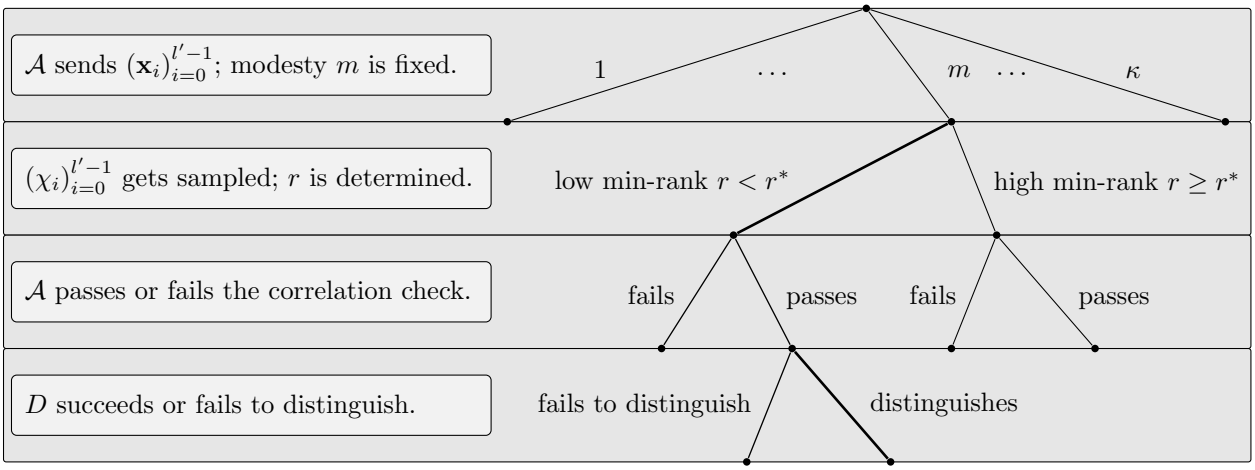

Figure 1: A depiction of the case structure considered by our proof.

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. Indeed, $\mathcal{A}$'s chance of passing the correlation check is exactly $2^{-\mathrm{rank}(F'_x)}$. If the minimal rank $r \geq r^*$, then $\mathrm{rank}(F'_x) \geq r^*$ certainly holds (i.e., regardless of which $x \in \mathbb{F}_{2^\kappa}$ $\mathcal{A}$ chooses). We see in this case that $\mathcal{A}$'s chance of passing is at most $2^{-r^*} \leq 2^{-\omega(\log(\kappa))}$, which is negligible. (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 3.15. *For each 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 run the procedure $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m+1\Big)$ on $\mathcal{A}$'s matrix $(\mathbf{x}_i)_{i=0}^{l-1}$, where m is $(\mathbf{x}_i)_{i=0}^{l-1}$'s modesty; we note immediately that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m+1\Big) = \mathsf{true}$, by definition of m. We write $(x'_i)_{i=0}^{l-1}$ for the vector of bit assignments ultimately assembled internally during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m+1\Big)$. (Even when $m = \kappa$, $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m+1\Big)$ nonetheless makes sense; this procedure will trivially return true on its first iteration, and will assign the entire vector $(x_i)_{i=0}^{l-1}$ arbitrarily.) We finally assign the further components $(x'_i)_{i=l}^{l'-1}$ arbitrarily. For each $i \in \{0, \ldots, l'-1\}$, we write $\mathbf{e}'_i \coloneqq x'_i \cdot (1, \ldots, 1) + \mathbf{x}_i$.

We begin by further simplifying F'_x. Concretely, we perform the following procedure. We define:

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

where the vectors $(\mathbf{e}'_i)_{i=0}^{l'-1}$ are as defined above. We note that, for each $x \in \mathbb{F}_2^\kappa$ and each $\Delta \in \mathbb{F}_2^\kappa$, we have the identity:

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

where we abbreviate $x^* \coloneqq \sum_{i=0}^{l'-1} \chi_i \cdot x'_i$. In the first step, we use the definition of the vectors $(\mathbf{e}'_i)_{i=0}^{l'-1}$ and the distributive law; we moreover use the identity $(x'_i \cdot (1, \ldots, 1)) * \Delta = x'_i \cdot \Delta$. We conclude that:

$$r = \min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F'_x) = \min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''_{x-x^*}) = \min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''_x);$$

in the last step, we exploit the fact that subtraction by x^* acts as a permutation on the set $\mathbb{F}_{2^\kappa}$. We conclude that, to prove the proposition, we may instead bound the probability that $\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''_x) < r^*$.

We further write $(\mathbf{e}''_i)_{i=0}^{l'-1}$ for the reduced row-echelon form over $\mathbb{F}_2$ of $(\mathbf{e}'_i)_{i=0}^{l'-1}$, and define:

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

Clearly, the $l' \times \kappa$ matrices $(\mathbf{e}''_i)_{i=0}^{l'-1}$ and $(\mathbf{e}'_i)_{i=0}^{l'-1}$ differ by left-multiplication by an $l' \times l'$ invertible matrix M with entries in $\mathbb{F}_2$. We thus obtain the following matrix relationship over $\mathbb{F}_2$:

$$\overbrace{\begin{bmatrix} — & \mathbf{e}''_0 & — \\ & \vdots & \\ — & \mathbf{e}''_{l'-1} & — \end{bmatrix}}^{l'} = \overbrace{\begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix}}^{l'} \cdot \overbrace{\begin{bmatrix} — & \mathbf{e}'_0 & — \\ & \vdots & \\ — & \mathbf{e}'_{l'-1} & — \end{bmatrix}}^{\kappa}.$$

Interpreting, for each $i \in \{0, \ldots, l'-1\}$, the elements $\mathbf{e}'_i$ and $\mathbf{e}''_i$ of $\mathbb{F}_2^\kappa$ as elements of $\mathbb{F}_{2^\kappa}$, we obtain the following equivalent matrix identity, now defined over $\mathbb{F}_{2^\kappa}$:

$$\overbrace{\begin{bmatrix} \mathbf{e}''_0 \\ \vdots \\ \mathbf{e}''_{l'-1} \end{bmatrix}}^{l'} = \overbrace{\begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix}}^{l'} \cdot \overbrace{\begin{bmatrix} \mathbf{e}'_0 \\ \vdots \\ \mathbf{e}'_{l'-1} \end{bmatrix}}^{1}.$$

Viewed as an $\mathbb{F}_{2^\kappa}$-matrix in the natural way, M of course remains invertible. We argue that the further matrix identity below—which we again understand over $\mathbb{F}_{2^\kappa}$—likewise holds:

$$\overbrace{\begin{bmatrix} \mathbf{e}''_0 * \Delta \\ \vdots \\ \mathbf{e}''_{l'-1} * \Delta \end{bmatrix}}^{l'} = \overbrace{\begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix}}^{l'} \cdot \overbrace{\begin{bmatrix} \mathbf{e}'_0 * \Delta \\ \vdots \\ \mathbf{e}'_{l'-1} * \Delta \end{bmatrix}}^{1}. \tag{2}$$

To justify this claim, we pick an arbitrary row $i \in \{0, \dots, l' - 1\}$. From the previous equality above, we see that $\mathbf{e}''_i = \sum_{j=0}^{l'-1} M_{i,j} \cdot \mathbf{e}'_j$ holds. Using this guarantee, we obtain in turn the equality $\sum_{j=0}^{l'-1} M_{i,j} \cdot \left(\mathbf{e}'_j * \Delta\right) = \sum_{j=0}^{l'-1} \left(\left(M_{i,j} \cdot \mathbf{e}'_j\right) * \Delta\right) = \left(\sum_{j=0}^{l'-1} M_{i,j} \cdot \mathbf{e}'_j\right) * \Delta = \mathbf{e}''_i * \Delta$, as required. Here, we again use the fact that $M_{i,j} \cdot (\mathbf{e}'_j * \Delta) = (M_{i,j} \cdot \mathbf{e}'_j) * \Delta$ holds for each bit $M_{i,j} \in \{0, 1\}$.

Using the final matrix identity (2) above, we finally reëxpress $F'''_x(\Delta)$ in the following way, for parameters $x \in \mathbb{F}_2^\kappa$ and $\Delta \in \mathbb{F}_2^\kappa$ arbitrary. We understand all matrix expressions below over $\mathbb{F}_{2^\kappa}$.

$$F'''_x(\Delta) = \begin{bmatrix} \chi_0 & \cdots & \chi_{l'-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e}''_0 * \Delta \\ \vdots \\ \mathbf{e}''_{l'-1} * \Delta \end{bmatrix} + x \cdot \Delta = \begin{bmatrix} \chi_0 & \cdots & \chi_{l'-1} \end{bmatrix} \cdot \begin{bmatrix} \\ M \\ \\ \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e}'_0 * \Delta \\ \vdots \\ \mathbf{e}'_{l'-1} * \Delta \end{bmatrix} + x \cdot \Delta.$$

The first equality is simply the definition of $F'''_x(\Delta)$; in the second, we use (2). On the other hand, the right-hand quantity above is simply a variant of $F''_x(\Delta)$ in which the $\mathbb{F}_{2^\kappa}$-elements $(\chi_i)_{i=0}^{l'-1}$ are replaced by $(\chi_i^M)_{i=0}^{l'-1}$, where we write:

$$\begin{bmatrix} \chi_0^M & \cdots & \chi_{l'-1}^M \end{bmatrix} \coloneqq \begin{bmatrix} \chi_0 & \cdots & \chi_{l'-1} \end{bmatrix} \cdot \begin{bmatrix} \\ M \\ \\ \end{bmatrix}.$$

Using this observation, we claim that the distribution of the variable quantity $r = \min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''_x)$, viewed as a function of the random scalars $(\chi_i)_{i=0}^{l'-1}$, is identical to that of $\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F'''_x)$ over $(\chi_i)_{i=0}^{l'-1}$. To prove this, we simply note that the random vectors $(\chi_i)_{i=0}^{l'-1}$ and $(\chi_i^M)_{i=0}^{l'-1} = (\chi_i)_{i=0}^{l'-1} \cdot M$ are identically distributed in $\mathbb{F}_{2^\kappa}^{l'}$. This fact is an immediate consequence of the invertibility of M (as an $\mathbb{F}_{2^\kappa}$-matrix). To prove the proposition, it thus further suffices to instead bound the probability that $\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F'''_x) < r^*$.

Lemma 3.16. *The reduced row-echelon form $(\mathbf{e}''_i)_{i=0}^{l'-1}$ 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 $\textsc{Modest}\left((\mathbf{x}_i)_{i=0}^{l-1}, m+1\right)$ (deferring all white and grey rows, as well as rows indexed $i \in \{l, \dots, l'-1\}$). 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 Gaussian elimination 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. Likewise, using the new pivot row to clear the pivot column also preserves this invariant. We thus conclude that there are at least as many pivots as there are black rows.

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

We continue our study of the maps $F'''_x : \Delta \mapsto \sum_{i=0}^{l'-1} \chi_i \cdot (\mathbf{e}''_i * \Delta) + x \cdot \Delta$. For each $i \in \{0, \ldots, l'-1\}$, the field element $\chi_i \in \mathbb{F}_{2^\kappa}$ induces an $\mathbb{F}_2$-isomorphism on $\mathbb{F}_2^\kappa$, by multiplication. We thus naturally associate to each χ_i a $\kappa \times \kappa$ matrix, which we call $\left[\chi_i\right]$. Finally, we write $\left[\chi'_i\right]$ for the $\kappa \times \kappa$ matrix which we obtain upon striking from $\left[\chi_i\right]$ those columns $j \in \{0, \ldots, \kappa-1\}$ for which $\mathbf{e}''_{i,j} = 0$. In other words, for $i \in \{0, \ldots, l'-1\}$ and each $j \in \{0, \ldots, \kappa-1\}$, $\left[\chi'_i\right]$'s j^{th} column $\left(\left[\chi'_i\right]_{k,j}\right)_{k=0}^{\kappa-1}$ is $\left(\left[\chi_i\right]_{k,j}\right)_{k=0}^{\kappa-1}$ if $\mathbf{e}''_{i,j} = 1$ and $(0)_{k=0}^{\kappa-1}$ otherwise. We note immediately that, for each $i \in \{0, \ldots, l'-1\}$, $\chi_i \cdot (\mathbf{e}''_i * \Delta) = \left[\chi'_i\right] \cdot \Delta$ holds. To prove this claim, we let $i \in \{0, \ldots, l'-1\}$ be arbitrary, and observe the following matrix identity over $\mathbb{F}_2$:

$$\chi_i \cdot (\mathbf{e}''_i * \Delta) = \begin{bmatrix} & \left[\chi_i\right] & \end{bmatrix} \cdot \begin{bmatrix} | \\ \mathbf{e}''_i * \Delta \\ | \end{bmatrix} = \begin{bmatrix} & \left[\chi_i\right] & \end{bmatrix} \cdot \begin{bmatrix} & \operatorname{diag}(\mathbf{e}''_i) & \end{bmatrix} \cdot \begin{bmatrix} | \\ \Delta \\ | \end{bmatrix} = \begin{bmatrix} & \left[\chi'_i\right] & \end{bmatrix} \cdot \begin{bmatrix} | \\ \Delta \\ | \end{bmatrix},$$

which is exactly what we wanted to show.

We thus obtain the equivalent expression $F'''_x : \Delta \mapsto \left(\sum_{i=0}^{l'-1} \left[\chi'_i\right] + \left[x\right]\right) \cdot \Delta$; here, we interpret the left-hand sum as a sum of matrices, and moreover write $\left[x\right]$ for x's associated $\kappa \times \kappa$ $\mathbb{F}_2$-matrix. We depict this latter expression for F'''_x in Figure 2 below.

$$F'''_x : \Delta \mapsto \left(\left[\left[\chi'_0\right] \right] + \cdots + \left[\left[\chi'_{l'-1}\right] \right] + \left[\left[x\right] \right] \right) \cdot \left[\Delta \right]$$

Figure 2: A depiction of the linear map F'''_x.

In Figure 2, we highlight the columns of the matrices $\left(\left[\chi'_i\right]\right)_{i=0}^{l'-1}$ either lightly or darkly, in order to convey that these matrices' columns are either preserved from or stricken from those of $\left(\left[\chi_i\right]\right)_{i=0}^{l'-1}$.

Because the matrix $(\mathbf{e}''_i)_{i=0}^{l'-1}$ is row-reduced, each of $(\mathbf{e}''_i)_{i=0}^{l'-1}$'s pivots contributes an independent random column to the matrix expression $\sum_{i=0}^{l'-1} \left[\chi'_i\right]$. We argue that we may, conservatively, consider the pivot columns alone in our study of F'''_x. We write $\mathbf{p} \subset \{0, \ldots, \kappa-1\}$ for the subset consisting of those column indices at which $(\mathbf{e}''_i)_{i=0}^{l'-1}$ has a pivot. Finally, we write P for the diagonal, $\kappa \times \kappa$ matrix for which, for each $j \in \{0, \ldots, \kappa-1\}$, $P_{j,j} = 1$ if $j \in \mathbf{p}$ and $P_{j,j} = 0$ otherwise (in slightly fancier terms, $P \coloneqq \operatorname{diag}(\mathbf{p})$). Clearly, right-multiplying the $\kappa \times \kappa$ matrix $\sum_{i=0}^{l'-1} \left[\chi'_i\right] + \left[x\right]$ by P can only decrease its rank; that is, we have the rank inequality $\operatorname{rank}\left(\left(\sum_{i=0}^{l'-1} \left[\chi'_i\right] + \left[x\right]\right) \cdot P\right) \leq \operatorname{rank}\left(\sum_{i=0}^{l'-1} \left[\chi'_i\right] + \left[x\right]\right)$. We abbreviate $X \coloneqq \sum_{i=0}^{l'-1} \left[\chi'_i\right]$, and define the final linear map:

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

In particular, F''''_x's matrix differs from F'''_x's simply by right-multiplication by P. In words, it replaces those among that matrix's columns corresponding to the non-pivot indices $j \in \{0, \ldots, \kappa-1\} \setminus \mathbf{p}$ with identically zero columns. We depict the map F''''_x in Figure 3 below.

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

Figure 3: A depiction of our linear map F''''_x.

Again, we shade the columns of Figure 3's matrices lightly or darkly. We note that precisely the pivot columns $\mathbf{p} \subset \{0, \ldots, \kappa-1\}$ are shaded lightly; moreover, the same set of columns is preserved in both X and $\left[x\right]$.

We recall again that $\operatorname{rank}(F''''_x) \leq \operatorname{rank}(F'''_x)$ holds for each pair of parameters $(\chi_i)_{i=0}^{l'-1}$ and x. In particular, if $\operatorname{rank}(F'''_x) < r^*$, holds, then $\operatorname{rank}(F''''_x) < r^*$ also does. We see that, for each vector $(\chi_i)_{i=0}^{l'-1} \in \mathbb{F}_{2^\kappa}^{l'}$ for which at least one $x \in \mathbb{F}_{2^\kappa}$ satisfies $\operatorname{rank}(F'''_x) < r^*$, certainly at least one $x \in \mathbb{F}_{2^\kappa}$ also satisfies $\operatorname{rank}(F''''_x) < r^*$ (and in fact the same x can be chosen). We see that our replacement of F'''_x with F''''_x serves only to expand the set of vectors $(\chi_i)_{i=0}^{l'-1} \in \mathbb{F}_{2^\kappa}$ under consideration; that is,

$$\Pr_{(\chi_i)_{i=0}^{l'-1} \leftarrow \mathbb{F}_{2^\kappa}^{l'}} \left[\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F'''_x) < r^* \right] \leq \Pr_{(\chi_i)_{i=0}^{l'-1} \leftarrow \mathbb{F}_{2^\kappa}^{l'}} \left[\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''''_x) < r^* \right].$$

To prove the proposition, it thus suffices to bound the probability $\Pr_{(\chi_i)_{i=0}^{l'-1} \leftarrow \mathbb{F}_{2^\kappa}^{l'}} [\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''''_x) < r^*]$. We undertake this task now.

We write $\widehat{\kappa} \coloneqq |\mathbf{p}|$ for the number of pivots in the matrix $(\mathbf{e}''_i)_{i=0}^{l'-1}$. As a notational aid, we work in the space of matrices $\mathbb{F}_2^{\kappa \times \widehat{\kappa}}$, instead of in $\mathbb{F}_2^{\kappa \times \kappa}$. For each $\kappa \times \kappa$ matrix X as above, we write X' for the $\kappa \times \widehat{\kappa}$ restriction of X to those columns $j \in \mathbf{p}$ (i.e., for the non-stricken columns of $X \cdot P$). Similarly, for each $x \in \mathbb{F}_{2^\kappa}$, we write $\left[x'\right]$ for the $\kappa \times \widehat{\kappa}$ restriction of $\left[x\right]$ to the pivot columns $j \in \mathbf{p}$. For each X as above and each $x \in \mathbb{F}_{2^\kappa}$, the matrices $\left(X + \left[x\right]\right) \cdot P \in \mathbb{F}_2^{\kappa \times \kappa}$ and $X' + \left[x'\right] \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ clearly have identical ranks (the former matrix differs from the latter only by the augmentation of $\kappa - \widehat{\kappa}$ identically zero columns). We thus restate our task as follows. For $X' \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ uniformly random, we must bound the probability with which there exists a field element $x \in \mathbb{F}_{2^\kappa}$ for which $\operatorname{rank}\left(X' + \left[x'\right]\right) < r^*$.

We make use of a counting argument in $\mathbb{F}_2^{\kappa \times \widehat{\kappa}}$. There are exactly 2^κ distinct field elements $x \in \mathbb{F}_{2^\kappa}$, and hence at most 2^κ distinct corresponding matrices $\left[x'\right]$. On the other hand, for each matrix $X' \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ for which, for some $x \in \mathbb{F}_{2^\kappa}$, $\operatorname{rank}\left(X' + \left[x'\right]\right) < r^*$ holds, we necessarily have that $X' + \left[x'\right] = Y$—or, equivalently, $X' = \left[x'\right] + Y$—where the matrix $Y \in \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$ is of rank less than r^*. In the following lemma, we bound the number of such matrices Y.

Lemma 3.17. *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 $\operatorname{rank}(Y) < r^*$.*

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 $\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}\left(X' + \left[x'\right]\right) < r^*$ is exactly the union, taken over $x \in \mathbb{F}_{2^\kappa}$, of the subsets $\left\{ \left[x'\right] + Y \;\middle|\; \operatorname{rank}(Y) < r^* \right\} \subset \mathbb{F}_2^{\kappa \times \widehat{\kappa}}$. In light of Lemma 3.17, 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 $\min_{x \in \mathbb{F}_{2^\kappa}} \operatorname{rank}(F''''_x) < r^*$ is thus at most

$$2^{(\kappa+\widehat{\kappa}) \cdot (r^*-1)+\kappa-\kappa \cdot \widehat{\kappa}}. \tag{3}$$

We reëxpress the exponent of (3) as:

$$(\kappa + \widehat{\kappa}) \cdot (r^* - 1) + \kappa - \kappa \cdot \widehat{\kappa} = \kappa \cdot (r^* - 1) + \kappa + \widehat{\kappa} \cdot (r^* - 1 - \kappa);$$

since $r^* - 1 - \kappa < 0$ necessarily holds (a consequence of $r^* \leq \kappa$), we see that the above exponent is decreasing as the number of pivots $\widehat{\kappa} \in \{0, \ldots, \kappa\}$ increases. On the other hand, by Lemma 3.16, we have the guarantee $\widehat{\kappa} \geq \frac{\kappa}{m} - 1$. We thus conclude that we may freely substitute $\widehat{\kappa} = \frac{\kappa}{m} - 1$ into the expression (3); doing this, we bound that quantity further by:

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

which is exactly the desired expression. This completes the proof of the proposition. □

We now consider the distinguisher's advantage. 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—provided $\mathcal{A}$ passes the correlation check—D may learn information about $\Delta \in \mathbb{F}_2^\kappa$ by the means of brute-force queries of the form $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \mathbf{e}_i * \Delta)$, where $i \in \{0, \ldots, l-1\}$. Finally, D has one further source of information. That is, D may make use of the very fact that $\mathcal{A}$ passed the correlation check. This datum tells D precisely that $\Delta \in \ker(F'_x)$. As $\operatorname{rank}(F'_x)$ grows, this fact—provided $\mathcal{A}$ passes—comes to reveal more information to D.

Proposition 3.18. *For each computationally unbounded distinguisher D, which makes at most $Q(\kappa)$ oracle queries, say, D's advantage, conditioned on $\mathcal{A}$ passing the correlation check, is at most $Q(\kappa) \cdot 2^{-m+\mathrm{rank}(F'_x)}$.*

Proof. We must bound the probability with which D—or, for that matter, $\mathcal{A}$—makes any query of the form $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$, where $i \in \{0, \ldots, l-1\}$. We write $\mathbf{d}$ for the vector constructed during the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, and $\mathbf{w} \coloneqq \mathbf{d} * \Delta$ for the projection of the sender's or simulator'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 likewise give $\mathcal{A}$ $\mathbf{w}$ after it submits x and t and passes the correlation check.

Having given $\mathcal{A}$ and D the projection $\mathbf{w}$, we observe that the only information that remains privileged these machines is the collective set of oracle inputs positions $\mathbf{t}_i + \mathbf{e}_i * \Delta$ for which $\mathbf{e}_i \not\subset \mathbf{d}$ (queried either by $\mathcal{S}$ or S) and $\mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta$ (queried only by S). Indeed, these positions—alongside those of the form $\mathbf{t}_i + \mathbf{e}_i * \Delta$ for $\mathbf{e}_i \subset \mathbf{d}$, which $\mathcal{A}$ and D, given $\mathbf{w}$, already know—represent precisely those possibly ever queried by any entity other than $\mathcal{A}$ and D. If neither $\mathcal{A}$ nor D queries any of these positions during their respective executions, then D's real-world and ideal-world views clearly become identical; indeed, in this case, D could just as well simulate the random oracle itself. Throughout the remainder of the proof, we refer only to D, for succinctness. (We further compare $\mathcal{A}$'s and D's views in Remark 3.23 below.)

We thus bound the probability with which D submits any query of the form $H(i \parallel \mathbf{t}_i + \mathbf{r})$, where $i \in \{0, \ldots, l-1\}$, and where the offset $\mathbf{r} \in \mathbb{F}_2^\kappa$ is such that either of the following two conditions hold:

$$\begin{cases} \mathbf{e}_i \not\subset \mathbf{d} \text{ and } \mathbf{r} = \mathbf{e}_i * \Delta. \\ \mathbf{r} = \overline{\mathbf{e}_i} * \Delta. \end{cases}$$

We write $Y \coloneqq \{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w} \wedge \Delta \in \ker(F'_x)\}$. Upon just beginning to run—and in possession of $\mathbf{w}$—D, for all it knows, views Δ as uniform in Y. We use the symbol $\mathbf{f} \in \mathbb{F}_2^\kappa$ to denote some vector of the form $\mathbf{f} = \mathbf{e}_i$ (where we require moreover that $\mathbf{e}_i \not\subset \mathbf{d}$) or $\mathbf{f} = \overline{\mathbf{e}_i}$, for $i \in \{0, \ldots, l-1\}$ arbitrary. For each such $\mathbf{f}$, we write $Y_\mathbf{f} \coloneqq \{\mathbf{f} * \Delta \mid \Delta \in Y\}$ 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.

Definition 3.19. If D queries $H(i \parallel \mathbf{t}_i + \mathbf{r})$, where $\mathbf{r} \in Y_\mathbf{f}$, then we say D has *checked* the fiber $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$.

Reëxpressing our above discussion linear-algebraically, we see that that our goal is precisely to bound the probability with which D checks a fiber containing Δ (i.e., with which, for some fiber $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ checked by D, where $\mathbf{f}$ is of the above form, $\Delta \in \mathbf{f}^{-1}(\mathbf{r})$ holds). In the following key lemma, we bound the proportion of candidates $\Delta \in Y$ which each among D's individual queries serves to check. The tricky part is to handle the interaction between the bit-positions tested by D, on the one hand, and the hint $\Delta \in \ker(F'_x)$, on the other.

Lemma 3.20. *For each $\mathbf{f}$ of the above form and each $\mathbf{r} \in Y_\mathbf{f}$, we have $\dim(Y) - \dim(\mathbf{f}^{-1}(r)) \geq m - \mathrm{rank}(F'_x)$.*

Proof. We first claim that $\dim(Y) \geq \left|\overline{\mathbf{d}}\right| - \mathrm{rank}(F'_x)$. Indeed, by definition, Y is the intersection in $\mathbb{F}_2^\kappa$ between $\ker(F'_x)$ and the subspace $\{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w}\}$. These subspaces are of codimension $\mathrm{rank}(F'_x)$ and $|\mathbf{d}|$, respectively, in $\mathbb{F}_2^\kappa$. By the subadditivity of codimension under intersection, we conclude that Y is of codimension at most $\mathrm{rank}(F'_x) + |\mathbf{d}|$ in $\mathbb{F}_2^\kappa$; in other words, Y is of dimension at least $\kappa - |\mathbf{d}| - \mathrm{rank}(F'_x) = \left|\overline{\mathbf{d}}\right| - \mathrm{rank}(F'_x)$, as desired. Finally, each $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ is the intersection in $\mathbb{F}_2^\kappa$ between $\ker(F'_x)$ and the 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 $\left|\overline{\mathbf{d} \cup \mathbf{f}}\right| = \left|\overline{\mathbf{d}} \cap \overline{\mathbf{f}}\right|$. Combining these facts, we obtain:

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

If $\mathbf{f} = \mathbf{e}_i$ holds (where again $\mathbf{e}_i \not\subset \mathbf{d}$), then the index $i \in \{0, \ldots, l-1\}$ is certainly white (or else $\mathbf{e}_i \subset \mathbf{d}$ would hold); by Lemma 3.8, we conclude in this case that $|\mathbf{f} \setminus \mathbf{d}| = |\mathbf{e}_i \setminus \mathbf{d}| \geq m$. If on the other hand $\mathbf{f} = \overline{\mathbf{e}_i}$, then by Lemma 3.9, we again obtain $|\mathbf{f} \setminus \mathbf{d}| = |\overline{\mathbf{e}_i} \setminus \mathbf{d}| \geq m$. In any case, we conclude that $|\mathbf{f} \setminus \mathbf{d}| \geq m$, so that, by the above calculation, $\dim(Y) - \dim\big(\mathbf{f}^{-1}(\mathbf{r})\big) \geq m - \mathrm{rank}(F'_x)$ holds, as required. □

Applying Lemma 3.20, we see that each fiber $\mathbf{f}^{-1}(\mathbf{r}) \subset Y$ checked by D covers a proportion of at most $2^{-m+\mathrm{rank}(F'_x)}$ of Y's points. We conclude that, provided it makes at most $Q(\kappa)$ queries, D can check in total a proportion of at most $Q(\kappa) \cdot 2^{-m+\mathrm{rank}(F'_x)}$ among Y's points. This completes the proof of the proposition. □

We are now in a position to prove the theorem. Traversing the tree of Figure 1, and invoking Propositions 3.15 and 3.18, we see that for each distinguisher D as above, each modesty $m \in \{1, \ldots, \kappa\}$, and each rank cutoff $r^* \in \{1, \ldots, \kappa\}$, D's advantage is at most:

$$\min\left(1, 2^{\kappa \cdot (r^*+1) - \kappa \cdot \frac{\kappa - r^* + 1}{m}}\right) \cdot 2^{-\mathrm{rank}(F'_x)} \cdot \min\left(1, Q(\kappa) \cdot 2^{-m + \mathrm{rank}(F'_x)}\right) + 2^{-r^*}. \tag{4}$$

The left-hand summand of (4) represents, in descending order, the lowermost three edges of the main path of Figure 1. That is, its first factor represents the probability (over $(\chi_i)_{i=0}^{l'-1}$) that the minimal rank $r < r^*$ is low, and leverages Proposition 3.15. Its middle factor $2^{-\mathrm{rank}(F'_x)}$ is exactly the probability that $\mathcal{A}$ passes the correlation check. Its final factor represents D's advantage, and corresponds to Proposition 3.18. The right-hand summand 2^{-r^*} of (4) represents the probability that $r \geq r^*$ and $\mathcal{A}$ passes the correlation check.

For notational convenience, we absorb the middle term $2^{-\mathrm{rank}(F'_x)}$ of (4)'s left-hand summand into its right-hand term. Since $2^{-\mathrm{rank}(F'_x)} \leq 1$, $2^{-\mathrm{rank}(F'_x)} \cdot \min\left(1, Q(\kappa) \cdot 2^{-m + \mathrm{rank}(F'_x)}\right) \leq \min(1, Q(\kappa) \cdot 2^{-m})$ necessarily holds; we thus further upper-bound (4) by:

$$\min\left(1, 2^{\kappa \cdot (r^*+1) - \kappa \cdot \frac{\kappa - r^* + 1}{m}}\right) \cdot \min\left(1, Q(\kappa) \cdot 2^{-m}\right) + 2^{-r^*}. \tag{5}$$

We set $r^* := \sqrt{\kappa}$ for the rest of the proof. The exponent of the right-hand term 2^{-r^*} of (5) is $-\sqrt{\kappa}$, which is $-\Omega(\sqrt{\kappa})$; we conclude immediately that that term is negligible. As for (5)'s left-hand term, 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 this left-hand term's first factor's exponent is bounded from above by $\kappa^{3/2} + \kappa - \frac{\kappa^2 - \kappa^{3/2} + \kappa}{m} < \kappa^{3/2} + \kappa - 2 \cdot \kappa^{3/2} + 2 \cdot \kappa - 2 \cdot \sqrt{\kappa} = -\kappa^{3/2} + 3 \cdot \kappa - 2 \cdot \sqrt{\kappa}$, which is in $-\Omega(\kappa^{3/2})$, so that this term is likewise negligible, and the result is proved. If $m \geq \frac{1}{2} \cdot \sqrt{\kappa}$ instead holds, then (5)'s left-hand term's second factor's exponent is at most $-\frac{1}{2} \cdot \sqrt{\kappa}$, which is in $-\Omega(\sqrt{\kappa})$, so that the left-hand term of (5) is again negligible (here, we use our assumption whereby $Q(\kappa)$ is polynomial). This completes the proof of the theorem. □

Intuition and examples. In order to further explain our proof strategy above, we discuss its treatment of various special cases. As a side effect, we also compare KOS to its predecessor work Ishai, Kilian, Nissim, and Petrank [IKNP03], in which the correlation check is absent (see Protocol 2.7 above).

Example 3.21. If $\mathcal{A}$ behaves honestly—that is, if it submits a monochromatic matrix, as in Example 3.6—then, as we've already seen, our simulator $\mathcal{S}$ above will extract the bit x_i for each vector $\mathbf{x}_i = x_i \cdot (1, \ldots, 1)$. In this case, $\mathbf{v}_{i,x_i} = H(i \,\|\, \mathbf{t}_i + \mathbf{e}_i * \Delta) = H(i \,\|\, \mathbf{t}_i)$ will hold for each $i \in \{0, \ldots, l-1\}$; by querying on-vectors, D will thus learn nothing about Δ. D will thus win only if it queries $H(i \,\|\, \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta) = H(i \,\|\, \mathbf{t}_i + \Delta)$ for some $i \in \{0, \ldots, l-1\}$. Each attempt to do so on the part of D will succeed with probability $\frac{1}{2^\kappa}$. D's advantage, provided that it makes at most 2^λ queries, will thus be $2^{\lambda - \kappa}$. We see that, in this case, in order to attain s bits of statistical security, the security parameter $\kappa := \lambda + s$ would suffice. In particular, against a semi-honest adversary, KOS (and so also IKNP) is secure, with security parameter just $\kappa = \lambda + s$ no less.

Example 3.22. If $\mathcal{A}$ submits the pathological matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$ of Example 3.7 and manages to pass the correlation check, then D will attain a $O(\kappa)$-time attack, as we now explain. Indeed, we note that $\mathcal{S}$ will assign its bits $(x_i)_{i=0}^{\kappa-1}$ arbitrarily. If, for any $i \in \{0, \ldots, \kappa-1\}$, $\mathcal{S}$ extracts $x_i = 1$, then, by checking $\mathbf{v}_{i,\overline{x_i}} \stackrel{?}{=} H(i \,\|\, \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta) = H(i \,\|\, \mathbf{t}_i + \mathbf{x}_i * \Delta)$ for both possible values of $\mathbf{x}_i * \Delta$ (we recall that $w(\mathbf{x}_i) = 1$), D will immediately distinguish the two worlds. We thus assume that $x_i = 0$ for each $i \in \{0, \ldots, \kappa-1\}$. In this case, for each $i \in \{0, \ldots, l-1\}$, by checking $\mathbf{v}_{i,x_i} \stackrel{?}{=} H(i \,\|\, \mathbf{t}_i + \mathbf{e}_i * \Delta) = H(i \,\|\, \mathbf{t}_i + \mathbf{x}_i * \Delta)$ for both possible values of $\mathbf{x}_i * \Delta$, D will immediately learn the bit Δ_i. After doing this for each $i \in \{0, \ldots, l-1\}$, D will learn Δ in its entirety; D may at this point explicitly check $\mathbf{v}_{i,\overline{x_i}} \stackrel{?}{=} H(i \,\|\, \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$ for any $i \in \{0, \ldots, l-1\}$. On the other hand, if $\mathcal{A}$ submits this matrix $(\mathbf{x}_i)_{i=0}^{l'-1}$, then its chance of passing the correlation check will become vanishingly small. Indeed, in the language of our proof above, $\mathcal{A}$'s matrix $(\mathbf{e}'_i)_{i=0}^{l'-1}$—i.e., corresponding to $\textsc{Modest}\left((\mathbf{x}_i)_{i=0}^{l-1}, 2\right)$—will begin exactly as $(\mathbf{x}_i)_{i=0}^{l-1}$ does (i.e., with a $\kappa \times \kappa$ identity submatrix), as will its row-reduced variant $(\mathbf{e}''_i)_{i=0}^{l'-1}$. We see that $\mathcal{A}$'s task will amount to finding, given a uniformly random binary matrix $X \in \mathbb{F}_2^{\kappa \times \kappa}$, a field element $x \in \mathbb{F}_{2^\kappa}$ for which $\mathrm{rank}(X + [x])$ becomes low. As our proof above shows, the mere existence of such an $x \in \mathbb{F}_{2^\kappa}$ will become extremely improbable over X (leave aside $\mathcal{A}$'s finding it).

Example 3.22 also shows that IKNP fails to be maliciously secure in the nonprogrammable random oracle model. Indeed, if the correlation check were absent, then $\mathcal{A}$ could simply proceed exactly as in Example 3.22 at its leisure. That example shows that, in the face of this strategy, $\mathcal{S}$—regardless of how it extracts its bits $(x_i)_{i=0}^{\kappa-1}$—necessarily hands to the distinguisher D an efficient attack strategy. This attack in fact dates to Nielsen [Nie07, § 4].

Interestingly, in the *fully programmable random oracle model*—an excessively weak model, of essentially theoretical interest (see Subsection 2.1 above)—IKNP becomes maliciously "secure", as we explain below.

Remark 3.23. We claim that in the fully programmable random oracle model, IKNP becomes maliciously "secure", with $\kappa \coloneqq 2 \cdot \lambda + 2 \cdot s$ no less. In the fully programmable model, D can't further query the random oracle during its execution (or rather, its queries are independent of its protocol transcript, and so useless). Instead, D must hope that $\mathcal{A}$ itself makes the winning query (i.e., during its execution of the protocol). Since $\mathcal{A}$ lacks access to the honest party's outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$, its position is much weaker than D's. Critically, $\mathcal{A}$ can't use the "incremental" attack strategy at the heart of our Theorem 3.14 above. Rather, $\mathcal{A}$ must hope—seeing neither $\Delta \in \mathbb{F}_2^\kappa$ nor $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$—nonetheless to query $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$, for some $i \in \{0, \ldots, l-1\}$. By comparing $\mathcal{A}$'s queries, after the fact, to the honest party's outputs $(\mathbf{v}_{i,0}, \mathbf{v}_{i,1})_{i=0}^{l-1}$, D may hope to find one for which $\mathbf{v}_{i,\overline{x}_i} \stackrel{?}{=} H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$ holds. We thus see that $\mathcal{S}$ should opt simply to make the off-vectors $\overline{\mathbf{e}_i}$ as high-weight as possible. That is, $\mathcal{S}$ should use the *majority rule* extraction strategy, whereby, for each $i \in \{0, \ldots, l-1\}$, it sets $x_i \coloneqq \textsc{Maj}(\mathbf{x}_i)$ (as well as $\mathbf{v}_{i,x_i} \coloneqq H(i \parallel \mathbf{q}_i + x_i \cdot \Delta)$ as usual). This strategy guarantees that $w(\overline{\mathbf{e}_i}) \geq \frac{\kappa}{2}$ holds for each $i \in \{0, \ldots, l-1\}$. In particular, each of $\mathcal{A}$'s query attempts $H(i \parallel \mathbf{t}_i + \overline{\mathbf{e}_i} * \Delta)$ will succeed with probability at most $\frac{1}{2^{\kappa/2}}$; $\mathcal{A}$'s overall probability of success will thus be $2^{\lambda-\kappa/2}$. We see that by setting $\kappa \coloneqq 2 \cdot \lambda + 2 \cdot s$, we obtain the security bound $2^{\lambda-\kappa/2} = 2^{\lambda-(\lambda+s)} = 2^{-s}$.

We believe that this separation result may be of independent interest.

Concrete security. 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 3.24. *For given computational and statistical security parameters λ and s, respectively, in order for it to be the case that $\left|\Pr\left[D\left(\mathsf{Real}_{\Pi_{\mathsf{ROT}}^{\kappa,l},\mathcal{A},R}(\kappa)\right) = 1\right] - \Pr\left[D\left(\mathsf{Ideal}_{\mathcal{F}_{\mathsf{ROT}}^{\kappa,l},\mathcal{S},R}(\kappa)\right) = 1\right]\right| \leq 2^{-s}$ 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 $r^* \coloneqq s + 1$ once and for all. We also fix $Q(\kappa) \coloneqq 2^\lambda$, and moreover substitute this expression into (5). We describe a selection procedure for κ which serves to bound (5) from above by 2^{-s} (i.e., for each possible $m \in \{1, \ldots, \kappa\}$).

Indeed, we set κ so large 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 (5)'s left-hand term 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 $\kappa \cdot (r^* + 1) - \kappa \cdot \frac{\kappa - r^* + 1}{m^*} = 0$ holds (i.e., where the first, increasing function crosses 0). Since we chose κ so as to guarantee that $\kappa \cdot (r^* + 1) - \kappa \cdot \frac{\kappa - r^* + 1}{\lambda + r^*} \leq 0$ holds, and since $\kappa \cdot (r^* + 1) - \kappa \cdot \frac{\kappa - r^* + 1}{m}$ is increasing in m (i.e., regardless of κ), our intersection point m^* certainly satisfies $m^* \geq \lambda + r^*$, so that $\lambda - m^* \leq -r^*$ in turn holds. We conclude that (5) 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^* in fact maximizes (5).

Since, by our 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 (5) 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 of the theorem. □

Remark 3.25. Theorem 3.24 can be viewed as a precise variant of the final argument of Theorem 3.14, 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 (5) decay as quickly as possible). Indeed, for κ chosen as in Theorem 3.24, 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 3.26. For $s \coloneqq 30$, and $\lambda \coloneqq 60$, Theorem 3.24 guarantees security as long as $\kappa \geq 2{,}942$.

Example 3.27. For $s \coloneqq 40$ and $\lambda \coloneqq 80$, Theorem 3.24 guarantees security as long as $\kappa \geq 5{,}122$.

Example 3.28. For $s \coloneqq 80$ and $\lambda \coloneqq 128$, Theorem 3.24 guarantees security as long as $\kappa \geq 17{,}218$.

A matching upper-bound. Theorem 3.24 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 the introduction of new techniques. We explain this as follows.

Corollary 3.29. *For each κ large enough, 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 $s \coloneqq \frac{1}{2} \cdot \sqrt{\kappa}$ and $\lambda \coloneqq \frac{1}{2} \cdot \sqrt{\kappa} + \frac{1}{2} \cdot \log(\kappa) - 1$. 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 3.24 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 3.30. In Corollary 3.29, 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 each random $\chi_i \leftarrow \mathbb{F}_{2^\kappa}$, each individual column $\left(\left[\chi_i \right]_{k,j} \right)_{k=0}^{\kappa-1}$ of $\left[\chi_i \right]$, for $j \in \{0, \ldots, \kappa - 1\}$, is itself uniformly random in $\{0,1\}^\kappa$. (Of course, the columns, considered jointly, are not independently random.) This property holds also for PSS, though they construct their matrices differently (with a single random column repeated).

The lower-bound established by our Corollary 3.29, 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 3.24 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.

Sources of loss. Our proof makes a number of loose approximations. In light of these, the fact of our proof's having nonetheless managed to closely approach Roy's upper-bound is fairly impressive. We discuss now these sources of loss, with a view towards future attempts to sharpen our proof (and possibly to bypass the PSS upper-bound).

In Proposition 3.15 above, our replacements of $F_{x,t}$ by F'_x, of F'_x by F''_x, and finally of F''_x by F'''_x are entirely lossless. On the other hand, our replacement of F'''_x by F''''_x introduces a plausibly significant source of loss, as we now explain. Indeed, right-multiplying F'''_x's matrix by P serves, in general, to reduce that map's rank. In the following example, we discuss an extreme instance of this phenomenon.

Example 3.31. We fix κ even, and suppose that $\mathcal{A}$ submits the matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ whose first two rows equal $\mathbf{x}_0 \coloneqq (\underbrace{1,\ldots,1}_{\kappa/2 \text{ ones}},\underbrace{0,\ldots,0}_{\kappa/2 \text{ zeros}})$ and $\mathbf{x}_1 \coloneqq (\underbrace{0,\ldots,0}_{\kappa/2 \text{ zeros}},\underbrace{1,\ldots,1}_{\kappa/2 \text{ ones}})$, and which elsewhere satisfies $\mathbf{x}_i \coloneqq (0,\ldots,0)$ (i.e., for each $i \in \{2,\ldots,l-1\}$). We claim without proof that the modesty of this matrix $(\mathbf{x}_i)_{i=0}^{l-1}$ is $m = \frac{\kappa}{2}$. Our proof of Proposition 3.15, upon running $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, \frac{\kappa}{2}+1\Big)$, will assign each bit $(x'_i)_{i=0}^{l-1}$ arbitrarily. The matrix $(\mathbf{e}'_i)_{i=0}^{l-1}$ will thus have rows $\mathbf{e}'_0$ and $\mathbf{e}'_1$ individually equal either to $(1,\ldots,1,0,\ldots,0)$ or to $(0,\ldots,0,1,\ldots,1)$; each further row $\mathbf{e}'_i$, for $i \in \{2,\ldots,l-1\}$, will be either $(0,\ldots,0)$ or $(1,\ldots,1)$. Upon row-reducing $(\mathbf{e}'_i)_{i=0}^{l-1}$, our proof will thus obtain a matrix $(\mathbf{e}''_i)_{i=0}^{l-1}$ beginning with either one or two distinct half-and-half rows, and identically zero elsewhere. For simplicity, we fix $\mathbf{e}''_0 = (1,\ldots,1,0,\ldots,0)$ and $\mathbf{e}''_1 = (0,\ldots,0,1,\ldots,1)$ (similar analyses apply to the other possibilities). We see that the matrix $X \coloneqq \sum_{i=0}^{l'-1} \big[\chi'_i\big]$ of our proof will be a "chimera", equal to $\big[\chi_0\big]$ on its left half and to $\big[\chi_1\big]$ on its right (we refer also to Figure 2). It seems implausible that there should exist a field element $x \in \mathbb{F}_{2^\kappa}$ for which $\operatorname{rank}\big(X + \big[x\big]\big)$ is low (say, strictly less than $\frac{\kappa}{2}$). Indeed, for each $x \in \mathbb{F}_{2^\kappa}$, $X + \big[x\big]$ will equal $\big[\chi_0 + x\big]$ on its left half and $\big[\chi_1 + x\big]$ on its right. (We note that the elements χ_0 and χ_1 will be distinct, except with probability $\frac{1}{2^\kappa}$.) On the other hand, our proof above will proceed ultimately by right-multiplying $X + \big[x\big]$ by the matrix P, which—since $(\mathbf{e}''_i)_{i=0}^{l-1}$ has just two pivots (namely, at $\mathbf{p} = \{0, \frac{\kappa}{2}\}$)—will be of extremely low rank (i.e., 2). We conclude that our proof has almost nothing to say about the matrix $(\mathbf{x}_i)_{i=0}^{l-1}$, even though it will plausibly render $\mathcal{A}$ unable to pass.

The loss showcased by Example 3.31 reflects our proof's inability to handle dependently random columns. Our matrix $X \coloneqq \sum_{i=0}^{l'-1} \big[\chi'_i\big]$ will, in general, represent a complicated mix of the matrices $\big(\big[\chi_i\big]\big)_{i=0}^{l'-1}$. That is, for each non-pivot column $j \in \{0,\ldots,\kappa-1\}$, those elements χ_i for which $\mathbf{e}''_{i,j} = 1$ will contribute dependent randomness to the column $\Big(\big[X\big]_{k,j}\Big)_{k=0}^{\kappa-1}$. It seems difficult to argue rigorously that these columns—i.e., for which j is not a pivot, but for which some $\mathbf{e}''_{i,j} = 1$—must inevitably contribute to X's rank.

Any sharpening of Proposition 3.15 along these lines would serve to directly improve the upper-bound κ of Theorem 3.24. The ideal outcome, of course, would eliminate the quadratic dependency of κ on s and λ.

We mention a further source of loss. We have left completely untouched the computational task of finding, for $X \in \mathbb{F}_2^{\kappa\times\kappa}$ given, a field element $x \in \mathbb{F}_{2^\kappa}$ for which $\operatorname{rank}\big(X + \big[x\big]\big)$ is low. Rather, our proof proceeds solely statistically (concerning itself only with whether a suitable x exists). It seems plausible that this computational problem could be hard (or at least hard enough to sharpen our proof).

In Proposition 3.18 above, we ignored the cost to D of obtaining $\mathbf{w} = \mathbf{d} * \Delta$. We also assumed, conservatively, that the subspaces $\ker(F'_x)$ and $\{\Delta \in \mathbb{F}_2^\kappa \mid \mathbf{d} * \Delta = \mathbf{w}\}$ intersected transversely in $\mathbb{F}_2^\kappa$, thereby lower-bounding $\dim(Y) \geq \big|\overline{\mathbf{d}}\big| - \operatorname{rank}(F'_x)$. In general, $\dim(Y)$ will rather be as high as $\min(|\overline{\mathbf{d}}|, \dim(\ker(F'_x)))$ (which is likely to equal $|\overline{\mathbf{d}}|$ in the cases of interest to us).

References

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

[IKNP03] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending oblivious transfers efficiently. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[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.

[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.

[Nie02] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[Nie07] Jesper Buus Nielsen. Extending oblivious transfers efficiently - how to get robustness almost for free. Cryptology ePrint Archive, Paper 2007/215, 2007. `https://eprint.iacr.org/2007/215`.

[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.

A Deferred Proofs

Proof of Lemma 3.11. We fix inputs $(\mathbf{x}_i)_{i=0}^{l-1}$ and $m \in \{1, \dots, \kappa\}$, and suppose that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$. We fix an arbitrary permutation $\sigma \in \mathbf{S}_l$, and write $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ for the alternate execution of that algorithm which instead traverses the inner loop 4 in the order $i \in \{\sigma(0), \dots, \sigma(l-1)\}$. For contradiction, we suppose that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$. 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 $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. 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 $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. Throughout, we give meaning to the symbols $(\mathbf{e}'_i)_{i=0}^{l-1}$ by means the bit assignment $(x'_i)_{i=0}^{l-1}$. 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 3.9 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, $\Big|\mathbf{e}'_{j_0} \setminus \mathbf{d}'_{j_0}\Big| < m$ held. On other hand, we claim that $\Big|\mathbf{e}'_{j_0} \setminus \mathbf{d}\Big| \geq m$. If $c_{j_0} = \mathsf{white}$, then Lemma 3.8 immediately implies the result. We thus assume that $c_{j_0} \neq \mathsf{white}$. In this case, if $x'_{j_0} = x_{j_0}$ held, then we would obtain $\mathbf{e}'_{j_0} \subset \mathbf{d}$, which would contradict $\alpha_0 \notin \mathbf{d}$. We conclude that $x'_{j_0} \neq x_{j_0}$. In this setting, Lemma 3.9 again implies that $\Big|\mathbf{e}'_{j_0} \setminus \mathbf{d}\Big| \geq m$, as desired. Since $\Big|\mathbf{e}'_{j_0} \setminus \mathbf{d}'_{j_0}\Big| < m$ and $\Big|\mathbf{e}'_{j_0} \setminus \mathbf{d}\Big| \geq m$ both hold, we conclude that $\mathbf{d}'_{j_0} \not\subset \mathbf{d}$.

We iteratively repeat this process as follows. We select an element $\alpha_1 \in \mathbf{d}'_{j_0} \setminus \mathbf{d}$, and write $j_1 \in \{i_0, \ldots, i_{r-1}\}$ 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, if $c_{j_1} = \mathsf{white}$, then $|\mathbf{e}_{j_1} \setminus \mathbf{d}| \geq m$ necessarily holds, by Lemma 3.8. Otherwise, since $x'_{j_1} = x_{j_1}$ would imply $\mathbf{e}'_{j_1} \subset \mathbf{d}$, contradicting $\alpha_1 \notin \mathbf{d}$, we conclude that $x'_{j_1} \neq x_{j_1}$, and that, by Lemma 3.9, $|\mathbf{e}_{j_1} \setminus \mathbf{d}| \geq m$ holds in any case. This conclusion, in light of our guarantee $\left|\mathbf{e}'_{j_1} \setminus \mathbf{d}'_{j_1}\right| < m$, implies that $\mathbf{d}'_{j_1} \not\subset \mathbf{d}$.

Iteratively proceeding in this way, we obtain a descending sequence of indices $j_0 > \cdots > j_{s-1}$ in $\{i_0, \ldots, i_{r-1}\}$, each satisfying $\mathbf{d}'_{j_k} \not\subset \mathbf{d}$, for $k \in \{0, \ldots, s-1\}$. This sequence must eventually terminate, so that $j_{s-1} = i_0$. Since $\mathbf{d}'_{i_0} = \varnothing$, we see finally that $\varnothing \not\subset \mathbf{d}$, an absurdity. We conclude that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{false}$, as required. Arguing symmetrically, we conclude similarly that if $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, then each permuted execution $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$ likewise returns true. □

Proof of Corollary 3.12. We fix $(\mathbf{x}_i)_{i=0}^{l-1}$, m and m' as in the hypothesis of the corollary; we suppose that $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$. Informally, by Lemma 3.11, we may, during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$, opt freely to treat $(\mathbf{x}_i)_{i=0}^{l-1}$'s rows instead in the order in which they are marked black during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, thereby guaranteeing the outcome $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big) = \mathsf{true}$. We present a detailed proof below.

We give meaning to the symbols $(\mathbf{e}_i)_{i=0}^{l-1}$ by means of the bit assignment $(x_i)_{i=0}^{l-1}$ extracted during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. We write $(i_0, \ldots, i_{r-1})$ for the ordered sequence of indices $i \in \{0, \ldots, l-1\}$ marked black during the course of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$, and $\varnothing \subset \mathbf{d}_{i_0} \subset \cdots \subset \mathbf{d}_{i_{r-1}}$ for the sequence of values respectively taken by $\mathbf{d}$ immediately before the updates $\mathbf{d} \cup= \mathbf{e}_{i_j}$, for $j \in \{0, \ldots, r-1\}$. Clearly, $\mathbf{d}_{i_j} \cup \mathbf{e}_{i_j} = \mathbf{d}_{i_{j+1}}$ holds for each $j \in \{0, \ldots, r-2\}$. We note moreover that $\left|\mathbf{e}_{i_j} \setminus \mathbf{d}_{i_j}\right| < m$ holds for each $j \in \{0, \ldots, r-1\}$. We fix a permutation $\sigma \in \mathbf{S}_l$ for which, for each $j \in \{0, \ldots r-1\}$, $\sigma(j) = i_j$, and which maps each $j \in \{r, \ldots, l-1\}$ otherwise arbitrarily. We write $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ for the alternate execution of $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ which treats its rows in the order given by σ, and $\mathbf{d}'$ for the vector assembled internally during that execution. We claim that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big) = \mathsf{true}$. By Lemma 3.11, this claim suffices to establish the result.

For each $j \in \{0, \ldots, r-1\}$, as of the beginning of the j^{th} iteration of the outer loop of $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$, if that algorithm hasn't already terminated (in which case we're done), then, by induction, we will have the inductive hypothesis $\mathbf{d}' = \mathbf{d}_{i_j}$. In this iteration, the inner loop 4 will pass over the already-black indices $(\sigma(0), \ldots, \sigma(j-1))$, before coming to test the still-white index $\sigma(j) = i_j$. Since $m \leq m'$, our guarantee $\left|\mathbf{e}_{i_j} \setminus \mathbf{d}_{i_j}\right| < m$ above implies *a fortiori* that at least one of the inequalities $\left|\mathbf{x}_{i_j} \setminus \mathbf{d}'\right| < m'$ and $\left|\overline{\mathbf{x}_{i_j}} \setminus \mathbf{d}'\right| < m'$ will hold, so that the condition 5 will be fulfilled, and $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ will mark i_j black. If these inequalities in fact both hold, then the condition $\left|\overline{\mathbf{e}_{i_j}} \setminus \mathbf{d}'\right| < m'$ of line 9 will moreover be fulfilled, so that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ will return true in the j^{th} iteration (in which case our proof is complete). Otherwise, we claim that the bit assignment $x_{i_j} \in \{0, 1\}$ selected by $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ on line 6 necessarily will match that selected during $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big)$. Indeed, again using our guarantee $\left|\mathbf{e}_{i_j} \setminus \mathbf{d}_{i_j}\right| < m$, we see that $\left|\mathbf{e}_{i_j} \setminus \mathbf{d}'\right| < m'$ will necessarily hold (and that $\left|\overline{\mathbf{e}_{i_j}} \setminus \mathbf{d}'\right| < m'$ moreover won't, by what we just assumed). We conclude that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ will perform the update step $\mathbf{d}' \cup= \mathbf{e}_{i_j}$ on line 8, thereby setting $\mathbf{d}' \coloneqq \mathbf{d}_{i_j} \cup \mathbf{e}_{i_j} = \mathbf{d}_{i_{j+1}}$ and preserving our inductive hypothesis.

This process must terminate during or before the $r-1^{\text{th}}$ iteration. Indeed, if the algorithm hasn't already terminated before the $r-1^{\text{th}}$ iteration, then, as of the beginning of that iteration, by induction, we will have $\mathbf{d}' = \mathbf{d}_{i_{r-1}}$; moreover, because $\textsc{Modest}\Big((\mathbf{x}_i)_{i=0}^{l-1}, m\Big) = \mathsf{true}$, we will necessarily have both inequalities $\left|\mathbf{x}_{i_j} \setminus \mathbf{d}_{i_{r-1}}\right| < m$ and $\left|\overline{\mathbf{x}_{i_j}} \setminus \mathbf{d}_{i_{r-1}}\right| < m$. We conclude again *a fortiori* that $\left|\mathbf{x}_{i_j} \setminus \mathbf{d}'\right| < m'$ and $\left|\overline{\mathbf{x}_{i_j}} \setminus \mathbf{d}'\right| < m'$ themselves will both hold, so that $\textsc{Modest}'\Big((\mathbf{x}_i)_{i=0}^{l-1}, m'\Big)$ will return true upon testing line 9, as required. □