Optimally Secure Block Ciphers from Ideal Primitives[*]

Stefano Tessaro[**]

Department of Computer Science
University of California, Santa Barbara
tessaro@cs.ucsb.edu
http://www.cs.ucsb.edu/~tessaro/

Abstract. Recent advances in block-cipher theory deliver security analyses in models where one or more underlying components (e.g., a function or a permutation) are *ideal* (i.e., randomly chosen). This paper addresses the question of finding *new* constructions achieving the highest possible security level under minimal assumptions in such ideal models.

We present a new block-cipher construction, derived from the Swap-or-Not construction by Hoang et al. (CRYPTO '12). With n-bit block length, our construction is a secure pseudorandom permutation (PRP) against attackers making $2^{n-O(\log n)}$ block-cipher queries, and $2^{n-O(1)}$ queries to the underlying component (which has itself domain size roughly n). This security level is nearly optimal. So far, only key-alternating ciphers have been known to achieve comparable security using $O(n)$ independent random permutations. In contrast, we only use a *single function* or *permutation*, and still achieve similar efficiency.

Our second contribution is a generic method to enhance a block cipher, initially only secure as a PRP, to additionally withstand related-key attacks without substantial loss in terms of concrete security.

[*] © IACR 2015. This is the full version of a paper to appear in the proceedings of ASIACRYPT 2015

[**] Partially supported by NSF grant CNS-1423566 and by the Glen and Susanne Culler chair.

1 Introduction

Several recent works provide ideal-model security proofs for key-alternating (i.e., Even-Mansour like) ciphers [25,23,14,2,50,16,15,19,26,31,17] and for Feistel-like ciphers [29,20,34,42,38]. In these proofs, the underlying components (wich are either permutations or functions) are chosen uniformly at random, and are *public*, i.e., the attacker can evaluate them. At the very least, these proofs target pseudorandom permutation (PRP) security: The block cipher, under a secret key, must be indistinguishable from a random permutation, provided the attacker makes at most q queries to the cipher, and at most q_F queries to the underlying component, for q and q_F as large as possible.

Ideal-model proofs imply that the block cipher is secure against *generic* attacks (i.e., treating every component as a black box). Heuristically, however, one hopes for even more: Namely, that under a careful implementation of the underlying component, the construction retains the promised security level.

CONTRIBUTIONS. This paper contributes along two different axes:

- **Weaker assumptions.** We present a new block-cipher design achieving near-optimal security, i.e., it remains secure even when q and q_F approach the sizes of the block-cipher and component domains, respectively. Our construction can be instantiated from a *function* or, alternatively, from a *single* permutation. This is the first construction from a function with such security level, and previous permutation-based constructions all relied on multiple permutations to achieve such high security.
- **Related-key security.** We show how to enhance our construction to achieve related-key security without significantly impacting its efficiency and security. This is achieved via a *generic transformation* of independent interest.

This work should not be seen primarily as suggesting a new practical block-cipher construction, but rather as understanding the highest achievable security level in the model block ciphers are typically analyzed. The resulting technical questions are fairly involved, and resolving them is where we see our contributions.

Still, we hope that our approach may inspire designers. Our instantiation from a permutation gives a possible path for a first proof-of-concept implementation, where one simply takes a single-round of AES as the underlying permutation. (And in fact, even a simpler object may be sufficient.)

1.1 First contribution: Full-domain security

We start by explaining our construction from a (random) *function*. Concretely, we consider block-cipher constructions BC with block length n and key length κ using an underlying keyless *function* F with m-bit inputs. We say that BC is (q, q_F)*-secure* (as a PRP) if no attacker can distinguish with substantial advantage the *real world* – where it can query q_F times a randomly sampled function F and overall q times the block cipher BC_K^F (using the function F and a random secret key K) – from an *ideal world* where BC_K^F is replaced by an independent random permutation of the n-bit strings. (In fact, we typically also allow inverse queries to the block cipher and the permutation.)

OUR GOAL. Let us first look at what can we expect for q and q_F when a cipher is (q, q_F)-secure. Clearly, $q_F \leq 2^m$ and $q \leq 2^n$, assuming queries are distinct. However, one can also prove that (roughly) $q_F < 2^\kappa$ is necessary, otherwise, the adversary can mount a brute-force key search attack. Moreover, $q \leq 2^m$ must also hold (cf. e.g. [28] for a precise statement of these bounds).

Here, we target (near) optimal security, i.e., we would like to achieve security for q and q_F as close as possible to 2^n and 2^m, respectively, whenever $m \geq n$. That is, the construction should remain secure even if the adversary can query most of its domain, and of that of the underlying function F. We note that the question is meaningful for every value of $m \geq n$, but we specifically target the case where $m \approx n$, e.g., $m = n$, or $m = n + O(\log n)$.

Previous constructions from functions fall short of achieving this: Gentry and Ramzan [29], and the recent generalization of their work by Lampe and Seurin [38], use a Feistel-based approach with $m = n/2$, and this hence yields (at best) $(2^{n/2}, 2^{n/2})$-security. (The work of [38] approaches that security level for increasing number of rounds.) In contrast, key-alternating ciphers (KACs) have been studied in several works [23,14,2,50,16,15,19,26,31], and the tightest bounds show them to be $(2^{n(1-\varepsilon)}, 2^{n(1-\varepsilon)})$-secure, when using $O(1/\varepsilon)$ rounds calling each an (independent) n-bit random *permutation*. However, there is no way of making direct use of KACs given only a non-invertible *function*.

The WSN construction. Our construction – which we call *Whitened Swap-or-Not* (WSN) – adds simple whitening steps to the Swap-or-Not construction by Hoang, Morris, and Rogaway [33], which was designed for the (different) setting where the component functions are secret-key primitives. Concretely, the WSN construction, on input $X = X_0$, iterates R times a very simple round structure of the form

$$X_{i+1} \leftarrow X_i \oplus (F_{b(i)}(W_i \oplus \max\{X_i, X_i \oplus K_i\}) \cdot K_i) ,$$

where W_i and K_i are round keys, max of two strings returns the largest with respect to lexicographic ordering, and $F_{b(i)}(x)$ returns the first bit of $F(x)$ in the first half of the rounds, and the second bit in the second half. (Moreover, $\cdot$ denotes simple scalar multiplication with a bit, i.e., $b \cdot X = X$ if $b = 1$, and 0^n else.) *In particular, our construction requires F to only output 2 bits.* The round structure is very weak[1], and it differs from the construction of [33] in that the *same* round function is invoked over multiple rounds, and as this function is public, we use a key W_i to whiten the input. We prove the following:

> **Main Theorem. (Informal)** The WSN construction for $R = O(n)$ rounds is $(2^{n-O(\log n)}, 2^{n-O(1)})$-secure.

Note that $O(n)$ rounds are clearly asymptotically optimal.[2] For some parameter cases, techniques from [49,47] can in fact be used to obtain a $(2^n, 2^{n(1-\varepsilon)})$-secure PRP, at the cost of a higher number of rounds.

Functions vs. permutations. It is beyond the scope of this paper to assess whether a function is a better starting point than a permutation *in practice*. Independently of this, we believe that studying constructions from *functions* is a fundamental theoretical problem for at least two reasons.

Foremost, functions are combinatorially simpler than permutations, and thus, providing constructions from them (and thus enabling a secure permutation structure) is an important theoretical question, akin to (and harder than) the problem of building PRPs from PRFs covered by a multitude of papers. Also, practical designs from keyless round functions have been considered (cf. e.g. [1]).

[1] A single round can easily be distinguished from a random permutation with a constant number of queries, as every input x is mapped to either x or $x \oplus K_i$.

[2] Even for one single query, every internal call to F can supply at most one bit of randomness, and the output must be (information theoretically) indistinguishable from a random n-bit string, and thus $\Omega(n)$ calls are necessary.

In addition, our construction *only* requires $c = 2$ output bits, and it is worth investigating whether such short-output functions are also harder to devise than permutations. We in fact provide some theoretical evidence that this may not be the case. We prove that an elegant construction by Hall, Wagner, Kelsey, and Schneier [32] can be used to transform *any* permutation from $n + c$ bits to $n + c$ bits into a function from n bits to c bits which is *perfectly indifferentiable* [44] from a random function. This property ensures that the *concrete security* of every cipher using a function $F : \{0,1\}^n \to \{0,1\}^c$ is preserved if we replace F with the construction from π, and allow the adversary access to π and its inverse π^{-1}. The construction makes 2^c permutation calls, and thus makes only sense for small c. In contrast, it should be noted that the only indifferentiable construction of a permutation from functions is complex and weakly secure [34], and that no suitable constant-complexity high-security constructions of large-range functions from permutations exist, the most secure construction being [41,46].

A SINGLE-PERMUTATION INSTANTIATION. With $c = 2$, combining the WSN construction with the HWKS construction yields a secure cipher with n-bit block length from a single permutation on $(n + 2)$-bit strings. In contrast, we are not aware of any trick to instantiate KACs from a single permutation retaining provable nearly-optimal security, even by enlarging the domain of the permutation. The only exception is the work of [15], which however only considers two rounds and hence falls short of achieving full-domain security.

The complexity of the resulting construction matches (asymptotically) that of KACs when targeting $(2^{n-O(\log n)}, 2^{n-O(1)})$-security. Nonetheless, a clear advantage of KACs is that their security degrades smoothly when reducing the number of rounds, whereas here $O(n)$ rounds remain necessary even for $(1, 0)$-security. We note that in the setting of functions constructions with such smooth security degradations are not known, even in the simpler setting of [33].

REDUCING THE KEY LENGTH. Arguably, an obvious drawback of our construction is that the key length grows with the number of rounds. We note that this is also true for key-alternating ciphers, and it is not unique to our construction.

It is worth noting that the key length *can* be reduced via standard techniques without affecting security, by deriving the round keys from a single $(n-d)$-bit master key K as $K_i \leftarrow H(K \,\|\, \langle i-1 \rangle)$ and $W_i \leftarrow H(K \,\|\, \langle R+i-1 \rangle)$ for all $i \in [R]$ and a function $H : \{0,1\}^n \to \{0,1\}^n$ (to be modeled as random in the proof), where $\langle \cdot \rangle$ denotes the $(d = \lceil \log(2R) + 1 \rceil)$-bit binary encoding of an integer in $[2R]$. (Note that $d = O(\log n)$.) The security proof is fairly straightforward, and omitted – it essentially accounts to excluding the event that H is queried on one of the values related to the key, and the reducing the analysis to the one with large keys. This adds an additional $q_H \cdot R/2^{n-d}$ term to the bound, where q_H is the number of queries to H. H can in fact be built from the very same function F, but this requires a slightly more involved analysis.

1.2 Second contribution: Related-key security

In the second part, we show how to generically make *any* block-cipher construction secure against *related-key attacks* (or RKA secure, for short) while preserving full-domain security and *small* input length of the underlying function.

ON RKA SECURITY. Several attacks over the last two decades (cf. e.g. [8,35,9,10,13,12,11]) have motivated RKA security as the new golden standard for block-cipher security. As formalized by Bellare and Kohno [5], RKA security is parameterized by a class of key transformations Φ. Then, pseudorandomness security defined above is extended to allow the attacker for block-cipher queries of the form $(\phi, +, X)$ or $(\phi, -, Y)$ for $\phi \in \Phi$ and $X, Y \in \{0,1\}^n$, resulting in $\mathsf{BC}_{\phi(K)}(X)$ and $\mathsf{BC}^{-1}_{\phi(K)}(Y)$.

It is easy to see that WSN is *not* RKA secure if the class Φ allows for XORing chosen offsets to individual keys. Querying an input X (with the original key), and querying $X \oplus \Delta$ while adding Δ to K_1 results in the same output with probability $1/2$. In the random permutation model, two recent works [26,19] have shown that KACs are RKA secure (for appropriate key scheduling), yet the resulting construction is only $(2^{n/2}, 2^{n/2})$-secure. Here, in contrast, we target full-domain security of the cipher.

RELATED-KEY SECURE KEY-DERIVATION. We consider a generic approach to shield ciphers from related-key attacks using *related-key secure key-derivation functions* (or RKA-KDF, for short). These are functions $\mathsf{KDF} : \{0,1\}^\kappa \to \{0,1\}^\ell$ with the property that under a random secret key K, the outputs of $\mathsf{KDF}(\phi(K))$, for different $\phi \in \Phi$, look random and independent. A similar concept was proposed by Lucks [40], and further formalized by Barbosa and Farshim [3]. For any secure block cipher BC, the new block cipher computes, for key K and input X, the value $\mathsf{BC}_{\mathsf{KDF}(K)}(X)$, and is easily proved to be RKA-secure. Note that this approach is very different from the one used for standard-model RKA-secure PRF and PRP constructions (as in [4]), which leverage algebraic properties of PRF constructions.[3]

Building RKA-KDFs in *ideal* models may appear too easy: A hash function $H : \{0,1\}^\kappa \to \{0,1\}^\ell$, when modeled as a random oracle [6], *is* a secure RKA-KDF. However, such construction can be broken in $2^{\kappa/2}$ queries by a simple collision argument.[4] If our goal is to achieve security almost 2^n to preserve security of e.g. WSN above, then we need to set $\kappa \geq 2n$. But what if we are building our block cipher from a primitive with n-bit inputs, like the very same primitive used to build the block cipher, as in the WSN setting above?

One approach is to use a domain extender in the sense of indifferentiability [44]. The only known construction with (near) optimal security is due to Maurer and Tessaro [45] (MT), and further abstracted by Dodis and Steinberger [22]. Unfortunately, instantiations of the MT construction are very inefficient, and make $O(n^c)$ calls to the underlying function for some undetermined (and fairly large) c.

MT-BASED RKA-KDFS. As our second contribution, we provide a highly parallelizable construction of a RKA-KDF from a keyless function with nearly optimal security, i.e., its outputs are pseudorandom even when evaluated on $q = 2^{n(1-\varepsilon)}$ related keys, and the underlying function can be evaluated $q_F = 2^{n(1-\varepsilon)}$ times, where $\varepsilon > 0$. Our construction is a variant of the MT construction. However, while the latter is inefficient as it relies on a complex combinatorial object, called an input-restricting function family, here, we show that to achieve RKA-KDF security it is sufficient to use a much simpler *hitter* [30], which can for instance be built from suitable constant-degree expander graphs.

Overall, our construction needs $O(n)$ calls to *independent* n-to-n-bit functions. (It can also be reformulated to call a single n-to-n-bit function.) We see it as a challenging open problem to improve the complexity, but we note that this already yields the most efficient known approach to ensure high related-key security for block ciphers built from ideal primitives.

[3] Also, our requirements are stronger than those for non-malleable codes and non-malleable key-derivation [24,27].

[4] For example, for $Q := 2^{\kappa/2}$, and an additive RKA attack asking for random $\Delta_1, \ldots, \Delta_Q$, one of the values $H(K \oplus \Delta_i)$ is going to collide with constant probability with one of the values $H(X_i)$, for independent κ-bit strings $X_1, \ldots, X_Q$, allowing to distinguish.

Indifferentiability. The question of building a block cipher from a random function which is as secure as an ideal cipher (with respect to indifferentiability) was studied and solved by [20,34]. In the same vein, indifferentiable KAC-like cipher constructions from permutations have been given [2,37,31]. While these constructions are related-key secure, their concrete security is fairly weak.

2 Preliminaries

2.1 Notation

Throughout this paper, we let $[n] := \{1, \ldots, n\}$. Further, we denote by $\mathsf{Fcs}(m, n)$ the set of functions mapping m-bit strings to n-bit strings, and by $\mathsf{Fcs}(*, n)$ the set of functions $\{0,1\}^* \to \{0,1\}^n$. Similarly, we let $\mathsf{Perms}(n) \subset \mathsf{Fcs}(n, n)$ be the set of permutations on $\{0,1\}^n$. Given a string $X \in \{0,1\}^m$, we denote by $X[i \ldots j]$ (for $i < j$) the sub-string consisting of bits $i, i+1, \ldots, j-1, j$ of X. We also write $X^{\leq i}$ instead of $X[1 \ldots i]$. Further, given another string $X' \in \{0,1\}^n$, we denote by $X \,\|\, X'$ the $(m+n)$-bit concatenation of X and X'.

Algorithms, constructions, and adversaries in this paper are with respect to some (not further specified) RAM model of computation. We explicitly denote by $\mathsf{C}[F]$ the fact that a construction C (implementing a function) makes queries to another function F, and we denote by $\mathcal{A}^{\mathsf{O}}$ the fact that an adversary $\mathcal{A}$ accesses an oracle O. We denote by $x \xleftarrow{\$} \mathsf{S}$ the process of sampling x from the set S uniformly at random, and by $y \xleftarrow{\$} \mathcal{A}^{\mathsf{O}}$ the process of running the randomized algorithm $\mathcal{A}$ with access to a randomized oracle O, and sampling its output y. Also, we denote by $\mathcal{A}^{\mathsf{O}} \Rightarrow y$ the event that the concrete value y is output in the same experiment. In general, we use a notation close to the one of Bellare and Rogaway's Game Playing framework [7], which we hope to be self evident.

Additionally, we denote by $\mathsf{Pr}\left[X = x\right]$ the probability that the random variable takes the value x, and by $\mathsf{E}[X]$ its expected value. Also, the *statistical distance* between two random variables X and X' is $\mathsf{SD}(X, X') = \frac{1}{2}\sum_x |\mathsf{Pr}\left[X = x\right] - \mathsf{Pr}\left[X' = x\right]|$, where the sum is over all values which can be taken by X or X'. It is well known that if $\mathcal{X}^+$ is the set of values x such that $\mathsf{Pr}\left[X = x\right] \geq \mathsf{Pr}\left[X' = x\right]$, then $\mathsf{SD}(X, X') = \sum_{x \in \mathcal{X}^+} (\mathsf{Pr}\left[X = x\right] - \mathsf{Pr}\left[X' = x\right])$.

2.2 Ideal models

Our analyses are in the *random function model*, where algorithms and adversaries are relative to a randomly chosen function $F \xleftarrow{\$} \mathsf{Fcs}(m, \ell)$ for parameters m and ℓ. A variant of the model grants access to *multiple* independent random functions $F_1, \ldots, F_t \xleftarrow{\$} \mathsf{Fcs}(m, \ell)$, but these can equivalently be implemented in the *single* random function model for $m' = m + \lceil \log t \rceil$, where the individual functions F_i are obtained as $F_i(X) = F(\langle i \rangle \,\|\, X)$, with $\langle i \rangle$ representing a $\lceil \log t \rceil$-bit encoding of i. We often denote $F = (F_1, \ldots, F_t)$ to stress this dual representation explicitly. Therefore, all upcoming definitions are in the single random function model without loss of generality.

We also recall that we can build a function F from m bits to ℓ bits by making ℓ calls to a function from $m + \lceil \log \ell \rceil$ bits to a single bit, i.e., $F(X) = F'(\langle 0 \rangle \,\|\, X) \,\|\, \cdots \,\|\, F'(\langle \ell - 1 \rangle \,\|\, X)$. The statement can be made precise via the notion of perfect indifferentiability [44], which we review in Appendix A.

The definitions of this section also naturally extend to the *random permutation model*, where adversaries and algorithms can query one or more random permutations sampled uniformly from $\mathsf{Perms}(n)$. In particular, adversaries are also allowed query the inverses of these permutations.

2.3 Block ciphers and (related-key) pseudorandomness

Let $\mathsf{BC}[F] : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be an efficient construction making calls to a function $F \in \mathsf{Fcs}(m, \ell)$. (We generally omit F whenever clear from the context.) We say that $\mathsf{BC} = \mathsf{BC}[F]$ is a (κ, n)-block cipher if $\mathsf{BC}(K, \cdot)$ is a permutation for all κ-bit K and all $F \in \mathsf{Fcs}(m, \ell)$, and use the notation BC_K to refer to this permutation. Typically, we assume that BC_K and BC_K^{-1} are very efficient to compute given K, where efficiency in particular implies a small number of calls to F.

(Multi-user) PRPs. We require block ciphers to be secure *pseudorandom permutations (PRPs)* [39]. In particular, we consider a multi-user version of PRP security, which captures joint indistinguishability of an (a-priori unbounded) number of block-cipher instantiations under different independent keys. The traditional (single-user) PRP notion is recovered by considering adversaries making queries for one single key. While the single- and multi-user versions are related by a hybrid argument, sticking with the latter will allow potentially tighter bounds in the second part of this paper, as the standard hybrid argument cannot be made very tight given only an overall bound on the number of queries.

To this end, we consider two security games $\mathsf{PRP}\text{-}b^{\mathcal{A}}_{\mathsf{BC},F}$ for $b \in \{0,1\}$. In both, $F \xleftarrow{\$} \mathsf{Fcs}(m, \ell)$ is initially sampled, as well as independent keys $K_1, K_2, \ldots \xleftarrow{\$} \{0,1\}^\kappa$, and permutations $P_1, P_2, \ldots \xleftarrow{\$} \mathsf{Perms}(n)$.[5] Then, the adversary $\mathcal{A}$ is executed, and is allowed to issue two types of queries:

- **Function queries** x, returning $F(x)$
- **Construction queries** (i, σ, z), where $i \in \mathbb{N}$, $\sigma \in \{-, +\}$, $z \in \{0,1\}^n$. For $b = 1$, the query returns $\mathsf{BC}_{K_i}(z)$ (if $\sigma = +$, this is a *forward query*) or $\mathsf{BC}^{-1}_{K_i}(z)$ (if $\sigma = -$, and this is a *backward query*). For $b = 0$, the query returns $P_i(z)$ or $P_i^{-1}(z)$, respectively.

Finally, $\mathcal{A}$ outputs a bit, which is also the game's output. Then, PRP-security of BC is defined via the following advantage metric

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC},F}(\mathcal{A}) := \mathsf{Pr}\left[\mathsf{PRP}\text{-}1^{\mathcal{A}}_{\mathsf{BC},F} \Rightarrow 1\right] - \mathsf{Pr}\left[\mathsf{PRP}\text{-}0^{\mathcal{A}}_{\mathsf{BC},F} \Rightarrow 1\right] .$$

We also denote by $\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC},F}(q, q_F)$ the maximal advantage of an adversary $\mathcal{A}$ making at most q construction queries and q_F function queries. Informally, we say that BC is (q, q_F)-secure if $\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC},F}(q, q_F)$ is "small", i.e., negligible in κ.

Related-key secure PRPs. We target the traditional notion of a related-key secure (or RKA-secure) PRP introduced by Bellare and Kohno [5]. In particular, for a key length κ, we consider a family $\Phi \subseteq \mathsf{Fcs}(\kappa, \kappa)$ of *key transformations*. Given a (κ, n)-block cipher $\mathsf{BC} = \mathsf{BC}[F]$ as above, we define the following two games $\mathsf{RKA}\text{-}\mathsf{PRP}\text{-}1$ and $\mathsf{RKA}\text{-}\mathsf{PRP}\text{-}0$. The game $\mathsf{RKA}\text{-}\mathsf{PRP}\text{-}b^{\mathcal{A}}_{\mathsf{BC},F,\Phi}$ proceeds as follows: It first samples $F \xleftarrow{\$} \mathsf{Fcs}(m, \ell)$, a key $K \xleftarrow{\$} \{0,1\}^\kappa$, and 2^κ independent permutations $P_{k'} \xleftarrow{\$} \mathsf{Perms}(n)$ for all κ-bit k'. Then, $\mathcal{A}$ issues two types of queries:

- **Function queries** x, returning $F(x)$
- **Construction queries** (σ, ϕ, X), where $\sigma \in \{-, +\}$, $\phi \in \Phi$, $z \in \{0,1\}^n$. For $b = 1$, the query returns $\mathsf{BC}_{\phi(K)}(z)$ (if $\sigma = +$, this is a *forward query*) or $\mathsf{BC}^{-1}_{\phi(K)}(z)$ (if $\sigma = -$, and this is a *backward query*). For $b = 0$, the query returns $P_{\phi(K)}(z)$ or $P^{-1}_{\phi(K)}(z)$, respectively.

Finally, $\mathcal{A}$ outputs a bit, which is also the game's output. We define the $\mathsf{RKA}\text{-}\mathsf{PRP}$ advantage as

$$\mathsf{Adv}^{\mathsf{RKA}\text{-}\mathsf{PRP}}_{\mathsf{BC},F,\Phi}(\mathcal{A}) = \mathsf{Pr}\left[\mathsf{RKA}\text{-}\mathsf{PRP}\text{-}1^{\mathcal{A}}_{\mathsf{BC},F,\Phi} \Rightarrow 1\right] - \mathsf{Pr}\left[\mathsf{RKA}\text{-}\mathsf{PRP}\text{-}0^{\mathcal{A}}_{\mathsf{BC},F,\Phi} \Rightarrow 1\right] .$$

The advantage measure $\mathsf{Adv}^{\mathsf{RKA}\text{-}\mathsf{PRP}}_{\mathsf{BC},F,\Phi}(q, q_F)$ is defined by taking the maximum.

[5] As we are sampling infinitely many objects, once can think of sampling these lazily the first time they are needed.

3 The Whitened Swap-or-Not Construction

3.1 The construction

We present a construction of a block cipher using a function $F : \{0,1\}^n \to \{0,1\}^2$, which we refer to as the *Whitened Swap-or-Not construction*, or WSN for short. This construction naturally extends the Shuffle-or-Not construction by Hoang, Morris, and Rogaway [33] to the keyless-function setting.

For any even round number $R = 2r$, the construction $\mathsf{WSN} = \mathsf{WSN}^{(R)}$ expects *round* keys $K_1, \ldots, K_R$ and *whitening* keys $W_1, \ldots, W_R$, which are all n-bit strings. Its computation proceeds as follows, where $j(i) = 1$ if $i \leq r$, and $j(i) = 2$ else, and we interpret F as two functions F_1 and F_2 such that $F_j(x)$ returns the j-th bit of $F(x)$ for $j \in \{1,2\}$.

```
Construction $\mathsf{WSN}^{(R)}_{K_1,\ldots,K_R,W_1,\ldots,W_R}(X)$:        // $X \in \{0,1\}^n$
$X_0 \leftarrow X$
For $i = 1, \ldots, R$ do
  $X'_{i-1} \leftarrow \max\{X_{i-1}, X_{i-1} \oplus K_i\}$
  $B_i \leftarrow F_{j(i)}(W_i \oplus X'_{i-1})$
  If $B_i = 1$ then $X_i \leftarrow X_{i-1} \oplus K_i$ else $X_i \leftarrow X_{i-1}$
Return $X_R$
```

In the description, the max of two strings is with respect to the lexicographic order, and note that its purpose is to elect a unique representant for every pair $\{X, X \oplus K_i\}$. As in [33], the construction extends naturally to domains which are arbitrary abelian groups. However, we will stick with the special case of bit strings in the following.

It is easy to see that the construction can efficiently be inverted given the keys, simply by reversing the order of the rounds.

3.2 Security of the WSN construction

Compared with the original Swap-or-Not construction, WSN adds at each round a whitening key W_i to the input of a (publicly evaluable) round function $F_{j(i)}$, as opposed to using a *secret* independent random function F_i (which in particular *cannot* be queried directly by the adversary). It is a well-known folklore fact that for a function $F : \{0,1\}^n \to \{0,1\}$, the construction mapping a key W and an input X to $F(W \oplus X)$ is indistinguishable from a random function under a random secret key W when F is random and publicly evaluable.

However, the high security of WSN does not follow by simply composing this folklore fact with the original analysis [33]. This is because the folklore construction can easily be distinguished from a random function via $\Theta(2^{n/2})$ queries to $F(W \oplus \cdot)$ (or a random function $f \xleftarrow{\$} \mathsf{Fcs}(n,1)$), and $\Theta(2^{n/2})$ queries to F.[6] To overcome this, a valid black-box instantiation would use a more complex construction mapping X to $F(W_1 \oplus X) \oplus \cdots \oplus F(W_k \oplus X)$ (analyzed in [28]) for the round functions within Swap-or-Not. This would however result in roughly $\Theta(n^2)$ calls to F, as opposed to $\Theta(n)$ achieved by WSN.

[6] Roughly, pick $X_1, \ldots, X_Q, X'_1, \ldots, X'_Q$ to be independent uniform n-bit strings of length $n - k$, for some $k = \lceil \log n \rceil$ and $Q \approx 2^{n/2}$. Then one just queries $Y_{z,i} \leftarrow F(W \oplus (X_i \parallel z))$ and $Y'_{z,j} \leftarrow F(X'_j \parallel z)$ for all $i, j \in [Q]$ and $z \in \{0,1\}^k$. The distinguisher finally outputs one if and only if there exist i and j such that $Y_{i,z} = Y'_{j,z}$ for all $z \in \{0,1\}^k$.

Security of WSN. The following theorem establishes the concrete security of the WSN construction with $R = 2r$ rounds.

Theorem 1 (Security of WSN). *For all $q, q_F > 0$ and for all $r \in \mathbb{N}$, we have*

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{WSN}^{(2r)},F}(q, q_F) \leq 2\sqrt{2}\sqrt{q}2^{n/4}\left(\frac{1}{2} + \frac{q \cdot r + q_F}{2 \cdot 2^n}\right)^{r/4} .$$

The proof of Theorem 1 is given in Section 3.3 below. Note that if $r \cdot q + q_F = (1-\alpha)2^n$, then the above term can be made to be 2^{-n} for $r = O(n/\alpha)$. For example, this allows to infer security for $q = 2^{n - \log n - O(1)}$ and $q_F = 2^{n-2}$.

We also have no reason to believe that the construction would be insecure if we used a function with a single output bit throughout the evaluation, but we could not find a suitable proof and leave this analysis as an open problem.

Single-permutation instantiation. The WSN construction can be instantiated from a single permutation if we are ready to enlarge the domain of the permutation to $n+2$ bits. This follows from a result of independent interest, proved in Appendix B. Namely, we prove that a 2^c-call construction of a function $F^\pi \in \mathsf{Fcs}(n, c)$ from any permutation $\pi \in \mathsf{Perms}(n+c)$ due to Hall, Wagner, Kelsey, and Schneier [32] is *perfectly* indifferentiable [44] from a random function. This in particular implies (by the composition theorem in Appendix A) that we can replace the function F by our construction and still achieve the *same* security bound in the random permutation model.

Full-domain security. Two recently published works [49,47] enhance swap-or-not to *full-domain* security (i.e., security against $q = 2^n$ queries) at the cost of making $O(n^2)$ calls to the construction in the worst-case. (The later work [47] shows how to reduce the complexity to $O(n)$ in the *average* case.) One could hope to use their results generically to obtain $(2^n, 2^{n(1-\varepsilon)})$-security in our setting.

Unfortunately, these results require security for $q = 2^{n-1}$, which is unattainable by the above bound. By inspecting the proof of Theorem 1, it is however not hard to verify that a version of the WSN construction with *independent* round functions $F_1, \ldots, F_R$ can be made to achieve $(2^{n-1}, 2^{n(1-\varepsilon)})$-security (in essence, this is because one can easily reduce the exponential term in the bound to $\left(\frac{1}{2} + \frac{q_F + q}{2 \cdot 2^n}\right)^{r/4}$) and the results from [49,47] can be used in a black-box way.

Nevertheless, we point out that in contrast to the *small-domain* setting of [49,47], here we are mostly targeting a large n (e.g., $n = 128$), for which $2^{n(1-\varepsilon)}$ security can be largely sufficient. The additional cost may thus not be necessary.

3.3 Proof of Theorem 1

Our proof shares similarities with the original analysis of Swap-or-Not [33], but dealing with the setting where the function F is public requires a careful extension and different techniques. To this end, we follow an approach used in previous works by Lampe, Patarin, and Seurin [36], and by Lampe and Seurin [38] to reduce security analyses for PRP constructions in ideal models to a non-adaptive analysis. (With some extra care due to the fact that we deal with the multi-user PRP security notion.) In particular, we are first going to prove that the WSN construction, restricted to half of its rounds, satisfies a weaker non-adaptive security requirement, which we introduce in the following paragraph.

Non-adaptive security. Let $\mathsf{BC} = \mathsf{BC}[F]$ be a (κ, n)-block cipher construction based on some function $F : \{0,1\}^m \to \{0,1\}^\ell$. Now, let us fix a set of tuples $T_F = \{(x_i, y_i)\}_{i \in [q_F]}$ with $x_i \in \{0,1\}^m$ and $y_i \in \{0,1\}^\ell$ for all $i \in [q_F]$, and such that every x_i appears only in one pair in T_F. Moreover, let us fix a sequence $\mathbf{X}$ of q distinct inputs such that $\mathbf{X}[j] = (i_j, X_j)$ for all $j \in [q]$, where $i_j \in \mathbb{N}$ and $X_j \in \{0,1\}^n$.

Then we consider two processes – sampling two sequences $\mathbf{Y}$ and $\mathbf{Y}'$ of q n-bit strings – defined as follows:

- $\mathbf{Y}$ (the *real world* distribution) is obtained by sampling random κ-bit strings $K_1, K_2, \ldots \overset{\$}{\leftarrow} \{0,1\}^\kappa$, sampling a random $F \overset{\$}{\leftarrow} \mathsf{Fcs}(m, \ell)$ *conditioned on* satisfying $F(x_i) = y_i$ for all $i \in [q_F]$, and finally letting $\mathbf{Y}[j] \leftarrow \mathsf{BC}[F]_{K_{i_j}}(X_j)$ for all $j \in [q]$.
- $\mathbf{Y}'$ (the *ideal world* distribution) is obtained by sampling random permutations $P_1, P_2, \ldots \overset{\$}{\leftarrow} \mathsf{Perms}(n)$, and letting $\mathbf{Y}[j] \leftarrow P_{i_j}(X_j)$ for all $i \in [q]$.

Then, we define the advantage metric

$$\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC},F}(\mathbf{X}, T_F) := \mathsf{SD}(\mathbf{Y}, \mathbf{Y}') ,$$

where SD denotes statistical distance. Moreover, let $\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC},F}(q, q_F)$ denote the maximum of $\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC},F}(\mathbf{X}, T_F)$ taken over all q-sequences $\mathbf{X}$ and all sets T_F of size q_F.

From non-adaptive to adaptive security. We make use of the following lemma. The proof is very similar to previous works [36,38] and makes crucial use of Patarin's H-coefficient method [48]. The main difference is that our version deals with the multi-user PRP security notion. (A self-contained version of the proof is found in Appendix G.)

Given a (κ, n)-block cipher $\mathsf{BC}[F]$ relying on a function $F : \{0,1\}^m \to \{0,1\}^\ell$, then let $\mathsf{BC}[F_1] \circ \mathsf{BC}^{-1}[F_2]$ be the $(2\kappa, n)$-block cipher which relies on two functions $F_1, F_2 : \{0,1\}^m \to \{0,1\}^\ell$, and which on input $X \in \{0,1\}^n$ and given key $K_1 \,\|\, K_2 \in \{0,1\}^{2\kappa}$, returns $\mathsf{BC}[F_2]^{-1}_{K_2}(\mathsf{BC}[F_1]_{K_1}(X))$. The following lemma tells us that if BC is non-adaptively secure (as in the above notion), then $\mathsf{BC} \circ \mathsf{BC}^{-1}$ is *adaptively* secure in the sense of being a secure PRP for attackers making both forward and backward queries.

Lemma 1 (Non-adaptive ⇒ Adaptive Security). *For all q, q_F, we have*

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC}[F_1] \circ \mathsf{BC}^{-1}[F_2], (F_1, F_2)}(q, q_F) \leq 4 \cdot \sqrt{\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC}[F], F}(q, q_F)} .$$

Note that a stronger version of this statement (essentially without the square root) can be proved [43,18] in the setting where $q_F = 0$.

Non-adaptive analysis of WSN. We first adopt a slightly different representation of the WSN construction. In particular, let $\overline{\mathsf{WSN}}^{(r)} = \overline{\mathsf{WSN}}^{(r)}[F]$ be the construction relying on a function $F : \{0,1\}^n \to \{0,1\}$ which operates as the original WSN construction for r rounds, but always uses the the function F (instead of using one function F_1 for the first half, and the function F_2 for the second half of the evaluation). Then, it is easy to see that

$$\mathsf{WSN}^{(2r)}[F_1, F_2] = \overline{\mathsf{WSN}}^{(r)}[F_1] \circ \left(\overline{\mathsf{WSN}}^{(r)}[F_2]\right)^{-1} , \qquad (1)$$

where in particular we have used the fact that the inverse of $\overline{\mathsf{WSN}}$ is just the $\overline{\mathsf{WSN}}$ itself, with round and whitening keys scheduled in the opposite order.

The key element of our proof is the following lemma, which, combined with Lemma 1 and Equation (1) immediately yields Theorem 1.

Lemma 2 (Non-adaptive security of $\overline{\mathsf{WSN}}$). *For all q and q_F, and $N = 2^n$,*

$$\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\overline{\mathsf{WSN}}^{(r)}[F],F}(q, q_F) \leq \frac{1}{2} q\sqrt{N} \left(\frac{1}{2} + \frac{q \cdot r + q_F}{2N} \right)^{r/2} .$$

Proof (Of Lemma 2). We fix a sequence of q distinct queries $\mathbf{X}$, as well as a set T_F of q_F input-output pairs. For now, we only consider the single-key setting, i.e., all queries $\mathbf{X}[j]$ are of the *same* index $i_j = 1$, and thus we omit these indices i_j. (We argue below how the multi-user case follows easily from our proof.) Denote the randomly chosen round keys as $\mathbf{K} = (\mathbf{K}[1], \ldots, \mathbf{K}[r])$ and the corresponding whitening keys as $\mathbf{W} = (\mathbf{W}[1], \ldots, \mathbf{W}[r])$.

We are going to consider the evolution of the evaluation of $\overline{\mathsf{WSN}}$ on these inputs *simultaneously*, and denote the joint state after $t \in \{0\} \cup [r]$ rounds as $\mathbf{X}_t = (\mathbf{X}_t[1], \ldots, \mathbf{X}_t[q])$, with $\mathbf{X}_0 = \mathbf{X}$. With $\mathbf{U}$ uniformly distributed on the set of q distinct n-bit strings, we are going to upper bound

$$\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\overline{\mathsf{WSN}}^{(r)}[F],F}(\mathbf{X}, T_F) = \mathsf{SD}(\mathbf{X}_r, \mathbf{U}) .$$

For any $i \in [q]$, denote by $\mathbf{Q}_t[i]$ the set of input-output pairs corresponding to the t F queries made to compute $\mathbf{X}_t[i]$ from $\mathbf{X}_0[i]$. Let now $U_{t,i}$ be a uniformly distributed value on the set $S_{t,i} := \{0,1\}^n \setminus \{\mathbf{X}_t[1], \ldots, \mathbf{X}_t[i-1]\}$, and let $\mathbf{U}_{t,i}$ to be a uniform $(q-i)$-tuple of distinct strings from $S_{t,i+1}$. Then, for all $t \in \{0\} \cup [r]$,

$$\begin{aligned}
\mathsf{SD}(\mathbf{X}_t, \mathbf{U}) &\leq \sum_{i=1}^{q} \mathsf{SD}((\mathbf{X}_t^{\leq i-1}, \mathbf{U}_{t,i-1}), ((\mathbf{X}_t^{\leq i}, \mathbf{U}_{t,i})) \\
&\leq \sum_{i=1}^{q} \mathsf{SD}((\mathbf{Q}_t^{\leq i-1}, \mathbf{X}_t^{\leq i-1}, U_{t,i}, \mathbf{U}_{t,i}), (\mathbf{Q}_t^{\leq i-1}, \mathbf{X}_t^{\leq i}, \mathbf{U}_{t,i})) \\
&= \sum_{i=1}^{q} \mathsf{SD}((\mathbf{Q}_t^{\leq i-1}, \mathbf{X}_t^{\leq i}), (\mathbf{Q}_t^{\leq i-1}, \mathbf{X}_t^{\leq i-1}, U_{t,i})) = \sum_{i=1}^{q} \mathsf{E}\left[\mathsf{SD}(\mathbf{X}_t[i], U_{t,i})\right] . \qquad (2)
\end{aligned}$$

since $\mathsf{SD}(f(X), f(Y)) \leq \mathsf{SD}(X, Y)$ for all f, X, Y, and the i-th expectation in the sum is over $\mathbf{Q}_t^{\leq i-1}$, $\mathbf{X}_t^{\leq i-1}$, $\mathbf{W}^{\leq t}$, and $\mathbf{K}^{\leq t}$.

For all $a \in S_{t,i}$, we now we define the random variable $p_{t,i}(a)$ as the probability that $\mathbf{X}_t[i] = a$ conditioned on the actual values taken by the random variables $\mathbf{Q}_t^{\leq i-1}$, $\mathbf{X}_t^{\leq i-1}, \mathbf{W}^{\leq t}, \mathbf{K}^{\leq t}$. (In particular, $p_{t,i}(a)$ is a random variable itself, as it is a function of these random variables.) Also, let $N_i := N - i + 1$. Then, by Cauchy-Schwarz and Jensen's inequalities, we obtain

$$\begin{aligned}
\mathsf{E}\left[\mathsf{SD}(\mathbf{X}_t[i], U_{t,i})\right] &= \frac{1}{2} \cdot \mathsf{E}\left[\sum_{a \in S_{t,i}} \left| p_{t,i}(a) - \frac{1}{N_i} \right| \right] \\
&\leq \frac{1}{2} \cdot \sqrt{N} \sqrt{ \mathsf{E}\left[\sum_{a \in S_{t,i}} \left(p_{t,i}(a) - \frac{1}{N_i} \right)^2 \right] } .
\end{aligned} \qquad (3)$$

We are going to give a recursive formula for $\mathsf{E}[\mathit{\Delta}_{t,i}]$, where

$$\mathit{\Delta}_{t,i} := \sum_{a \in S_{t,i}} \left(p_{t,i}(a) - \frac{1}{N_i} \right)^2 .$$

Note that $\mathit{\Delta}_{0,i} = \mathsf{E}[\mathit{\Delta}_{0,i}] = 1 - \frac{1}{N_i}$. It is now convenient to assume that $\mathbf{Q}_t^{\leq i-1}$, $\mathbf{X}_t^{\leq i-1}$, $\mathbf{K}^{\leq t}$, $\mathbf{W}^{\leq t}$ are fixed to some values (and thus so are $\mathit{\Delta}_{t,i}$ and $p_{t,i}(a)$), and we are going to study $\mathsf{E}[\mathit{\Delta}_{t+1,i}]$, where the expectation is now over $\mathbf{X}_{t+1}^{\leq i-1}$, $\mathbf{K}[t+1]$, $\mathbf{W}[t+1]$ and $\mathbf{Q}_{t+1}^{\leq i-1}$. In particular, define $\mathcal{Q}_b$ (for $b \in \{0,1\}$) to be the set of all inputs of queries to F for which we know the corresponding output, i.e., $x \in \mathcal{Q}_b$ if $(x,b) \in T_F$ or $(x,b) \in \mathbf{Q}_t[j]$ for some $j \in [i-1]$. Moreover let $\mathcal{Q} := \mathcal{Q}_0 \cup \mathcal{Q}_1$ and $Q := |\mathcal{Q}|$, and note that $Q \leq t \cdot (i-1) + q_F$.

With the above being fixed, we are now considering the random experiment where we sample $\mathbf{K}[t+1]$ and $\mathbf{W}[t+1]$, and we are going to compute the expectation of $\mathit{\Delta}_{t+1,i}$ in this experiment. More concretely, we define a function $\varphi : S_{t,i} \to S_{t+1,i}$ (which is also a random variable, as it depends on $S_{t+1,i}$, $\mathbf{K}[t+1]$ and $\mathbf{W}[t+1]$) as follows:

$$\varphi(a) = \begin{cases} a & \text{if (1) } \max\{a \oplus \mathbf{K}[t+1], a\} \oplus \mathbf{W}[t+1] \in \mathcal{Q}_0 \text{, or} \\ & \text{(2) } a \oplus \mathbf{K}[t+1] \notin S_{t+1,i} \\ & \text{and } \max\{a \oplus \mathbf{K}[t+1], a\} \oplus \mathbf{W}[t+1] \notin \mathcal{Q} \text{, or} \\ & \text{(3) } a \oplus \mathbf{K}[t+1] \in S_{t,i} \text{ and } \max\{a \oplus \mathbf{K}[t+1], a\} \oplus \mathbf{W}[t+1] \notin \mathcal{Q}, \\ a \oplus \mathbf{K}[t+1] & \text{if (4) } \max\{a \oplus \mathbf{K}[t+1], a\} \oplus \mathbf{W}[t+1] \in \mathcal{Q}_1 \text{, or} \\ & \text{(5) } a \notin S_{t+1,i} \text{ and } \max\{a \oplus \mathbf{K}[t+1], a\} \oplus \mathbf{W}[t+1] \notin \mathcal{Q}. \end{cases}$$

Note that φ is a bijection. Indeed, if $\mathbf{X}_t[i] = a$ implies $\mathbf{X}_{t+1}[i] = a'$ (where $a' \in \{a, a \oplus \mathbf{K}[t+1]\}$), then $\varphi(a) = a'$ (this corresponds to exactly one of the first four cases), and otherwise we let $\varphi(a) = a$. Also note that φ does not depend (directly) on $\mathbf{Q}_{t+1}^{\leq i-1}$, only on $S_{t+1,i}$, $\mathbf{K}[t+1]$, $\mathbf{W}[t+1]$, and $\mathbf{Q}_t^{\leq i-1}$. Using both the bijectivity of φ as well as the linearity of expectation,

$$\mathsf{E}\left[\mathit{\Delta}_{t+1,i}\right] = \sum_{a \in S_{t,i}} \mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \frac{1}{N_i} \right)^2 \right] .$$

Recall that the expectation here is over the choice of $\mathbf{Q}_{t+1}^{\leq i-1}$, $\mathbf{K}[t+1]$ and $\mathbf{W}[t+1]$. We prove the following lemma in Appendix C.

Lemma 3. *For all $a \in S_{t,i}$,*

$$\mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \tfrac{1}{N_i} \right)^2 \right] = \left(1 - \tfrac{3}{4} \tfrac{N_i(N-Q)}{4 \cdot N^2} \right) \left(p_{t,i}(a) - \tfrac{1}{N_i} \right)^2 + \tfrac{1}{4} \tfrac{N-Q}{N^2} \mathit{\Delta}_{t,i} .$$

We can thus replace $\mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \tfrac{1}{N_i} \right)^2 \right]$ in the above, and using the fact that $\mathit{\Delta}_{t,i} = \sum_{a \in S_{t,i}} (p_{t,i}(a) - \frac{1}{N_i})^2$, this simplifies to

$$\mathsf{E}\left[\mathit{\Delta}_{t+1,i}\right] = \sum_{a \in S_{t,i}} \mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \frac{1}{N_i} \right)^2 \right] \leq \left(1 - \frac{N_i \cdot (N - Q_t)}{2 \cdot N^2} \right) \mathit{\Delta}_{t,i} ,$$

```
Procedure MAIN:
// Game RKA-KDF-$b$, $b \in \{0,1\}$
$F \xleftarrow{\$} \mathsf{Fcs}(m,n)$, $G \xleftarrow{\$} \mathsf{Fcs}(*,\ell)$
$K \xleftarrow{\$} \{0,1\}^\kappa$
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{F},\mathsf{Eval}}$
Return $b'$

Procedure Eval($\phi$):
// Game RKA-KDF-$b$, $b \in \{0,1\}$
If $b = 0$ then
  Return $\mathsf{KDF}[F](\phi(K))$
Else return $G(\phi)$

Procedure F($x$):
Return $F(x)$
```

Fig. 1. RKA-KDF security. The procedure Eval, in both games, takes as input a function $\phi \in \Phi$. Also, the notation $G(\phi)$ denotes G applied to some unique bit-encoding of the function ϕ.

where $Q_t = t(i-1)+q_F$. Now, we come back to thinking of $\mathbf{X}_t^{\leq i-1}$, $\mathbf{K}^{\leq t}$ and $\mathbf{W}^{\leq t}$ as being randomly chosen (rather than fixed), and evaluate $\mathsf{E}[\Delta_{t,i}]$ recursively. The above in particular implies that $\mathsf{E}[\Delta_{t,i}] \leq \left(1 - \frac{N_i \cdot (N-Q)}{2 \cdot N^2}\right) \mathsf{E}[\Delta_{t-1,i}]$, and thus

$$\mathsf{E}[\Delta_{r,i}] \leq \left(1 - \frac{N_i \cdot (N - Q_{r-1})}{2 \cdot N^2}\right)^r \leq \left(\frac{1}{2} + \frac{r \cdot q + q_F}{2N}\right)^r \ .$$

Now, we can put this together with (2) and (3), and see that

$$\mathsf{SD}(\mathbf{X}_r, \mathbf{U}_q) \leq \frac{1}{2} q \cdot \sqrt{N} \cdot \left(\frac{1}{2} + \frac{r \cdot q + q_F}{2N}\right)^{r/2} \ .$$

Note that for the multi-user case, the proof is essentially the same, with slightly more complex notation. The only difference is that we define $S_{t,i}$ and all related quantities only with respect to the previous queries for the same key / user. The upper bounds are the same however, as they only depend on N, q and q_F. This concludes the proof of Lemma 2. □

4 Related-Key Security

4.1 Related-key secure key derivation

We consider the general notion of a related-key secure key-derivation function, or RKA-KDF for short. Informally, for a class of key-transformation functions $\Phi \subseteq \mathsf{Fcs}(\kappa, \kappa)$, this is a function $\mathsf{KDF} : \{0,1\}^\kappa \to \{0,1\}^\ell$ such that $\mathsf{KDF}(\phi(K))$ gives independent, pseudorandom values for every $\phi \in \Phi$. A similar notion was considered by Lucks [40] and by Barbosa and Farshim [3].

FORMAL DEFINITION. Let $\mathsf{KDF}[F] : \{0,1\}^\kappa \to \{0,1\}^\ell$ be a construction that calls a function $F : \{0,1\}^m \to \{0,1\}^n$. In Figure 1, we define the security games RKA-KDF-0 and RKA-KDF-1 involving an adversary $\mathcal{A}$ and a class of key transformations $\Phi \subseteq \mathsf{Fcs}(\kappa, \kappa)$. In the real world (Game RKA-KDF-0), the adversary $\mathcal{A}$ makes queries to a random function F via the F oracle and can obtain evaluations of $\mathsf{KDF}[F](\phi(K))$ for multiple $\phi \in \Phi$ of its choice via the Eval oracle, and these values should be indistinguishable from random values, which are returned by the Eval oracle in the ideal world (i.e., in Game RKA-KDF-1). The RKA-KDF-advantage is then defined as

$$\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF},F,\Phi}(\mathcal{A}) = \mathsf{Pr}\left[\mathsf{RKA\text{-}KDF\text{-}0}^{\mathcal{A}}_{\mathsf{KDF},F,\Phi} \Rightarrow 1\right] - \mathsf{Pr}\left[\mathsf{RKA\text{-}KDF\text{-}1}^{\mathcal{A}}_{\mathsf{KDF},F,\Phi} \Rightarrow 1\right] \ ,$$

and $\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF},F,\Phi}(q, q_F)$ is obtained by maximizing the above over all adversaries making q queries to Eval and making q_F queries to F via the F oracle.

Remark 1. An alternative definition has the Eval oracle return $G(\phi(K))$ for a random function $G \xleftarrow{\$} \mathsf{Fcs}(\kappa, \ell)$. Our choice is better suited to the composition theorem below, and shifts the burden of dealing with the combinatorics of Φ to the RKA-KDF security proof.

<u>The composition theorem.</u> We can compose an arbitrary (ℓ, n)-block cipher construction $\mathsf{BC}[F]$ and a key-derivation function $\mathsf{KDF} : \{0,1\}^\kappa \to \{0,1\}^\ell$ using the same function F, into a new (κ, n) block cipher $\overline{\mathsf{BC}} = \overline{\mathsf{BC}}[F, \mathsf{KDF}]$ such that

$$\overline{\mathsf{BC}}_K(X) = \mathsf{BC}_{\mathsf{KDF}(K)}(X) \; . \tag{4}$$

for every $K \in \{0,1\}^\kappa$ and $X \in \{0,1\}^n$. The following theorem shows that if BC is a secure PRP and KDF is RKA-KDF secure, then the composition $\overline{\mathsf{BC}}$ is a related-key secure PRP. Note that the fact that we consider multi-user PRP security is central in allowing us a tight reduction.

Theorem 2 (The Composition Theorem). *Let $\overline{\mathsf{BC}} = \overline{\mathsf{BC}}[F, \mathsf{KDF}]$ be the (κ, n)-block cipher defined above, and assume that BC makes at most t calls to F upon each invocation. Let $\Phi \subset \mathsf{Fcs}(\kappa, \kappa)$ be a class of key transformations. Then, for all q, q_F,*

$$\mathsf{Adv}^{\mathsf{RKA\text{-}PRP}}_{\overline{\mathsf{BC}},F,\Phi}(q, q_F) \leq 2 \cdot \mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF},F,\Phi}(q, q_F + q \cdot t) + \mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC},F}(q, q_F) \; .$$

Proof (Sketch). One uses $\mathsf{RKA\text{-}KDF}$ security to transition from $\mathsf{RKA\text{-}PRP\text{-}1}$ to a setting where each query (ϕ, x) to the block cipher is replied with an independent key K_ϕ as $\mathsf{BC}_{K_\phi}(x)$, i.e., we map every ϕ with an independent κ-bit key K_ϕ. This is exactly $\mathsf{PRP\text{-}1}$ (except that users are now identified by elements of Φ) and results in the additive term $\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF}}(q, q_F + q \cdot t)$ in the bound by a standard reduction. Similarly, one uses $\mathsf{RKA\text{-}KDF}$ security to transition from $\mathsf{RKA\text{-}PRP\text{-}0}$ to a setting where each query (ϕ, x) to the block cipher is replied with an independent permutation P_ϕ, and this exactly maps to $\mathsf{PRP\text{-}0}$, and results in another additive term $\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF}}(q, q_F + q \cdot t)$. The final bound follows by the triangle inequality. □

Note that in a similar way, if KDF and BC use *different* functions F and F', then we can reduce $\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF},F,\Phi}(q, q_F + q \cdot t)$ to $\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{KDF},F,\Phi}(q, q_F)$.

4.2 Efficient RKA-KDF-secure construction

This section presents an $\mathsf{RKA\text{-}KDF}$-secure construction from a (small number of) random functions $F : \{0,1\}^n \to \{0,1\}^n$ approaching $(2^{n(1-\varepsilon)}, 2^{n(1-\varepsilon)})$-security. (As we argue below, this can be turned into a construction from a single function $F : \{0,1\}^n \to \{0,1\}$ with standard tricks.) Our construction will guarantee Φ-RKA-KDF-security for every class $\Phi \subseteq \mathsf{Fcs}(\kappa, \kappa)$ with the following two properties for (small) parameters $\gamma, \lambda \in [0,1]$:

γ-collision resistance. $\mathsf{Pr}\left[K \xleftarrow{\$} \{0,1\}^\kappa : \phi(K) = \phi'(K)\right] \leq \gamma$ for any two distinct $\phi, \phi' \in \Phi$.

λ-uniformity. For any $\phi \in \Phi$, we have that $\mathsf{SD}(K, \phi(K)) \leq \lambda$ for $K \xleftarrow{\$} \{0,1\}^\kappa$, i.e., $\phi(K)$ is λ-close to uniform for a random key K.

For example, $\Phi^\oplus = \{K \mapsto K \oplus \Delta \, : \, \Delta \in \{0,1\}^\kappa\}$ is both 0-collision-resistant and 0-uniform.

Combinatorial hitters. Our construction makes use of the standard combinatorial notion of a hitter [30], which we introduce with a slightly different parameterization than what used in the literature. Consider a family of functions $\mathsf{E} = (\mathsf{E}_1, \dots, \mathsf{E}_t)$ such that $\mathsf{E}_i : \{0,1\}^\kappa \rightarrow \{0,1\}^n$.

Definition 1 (Hitters). *The functions* $\mathsf{E} = (\mathsf{E}_1, \dots, \mathsf{E}_t)$ *with* $\mathsf{E}_i : \{0,1\}^\kappa \rightarrow \{0,1\}^n$ *are an* (α, β)-hitter *if for all subsets* $\mathcal{Q} \subseteq \{0,1\}^n$ *with* $|\mathcal{Q}| \leq \beta \cdot 2^n$, $\mathsf{Pr}\left[K \leftarrow \{0,1\}^\kappa : \forall i \in [t] : \mathsf{E}_i(K) \in \mathcal{Q}\right] \leq \alpha$.

In our setting, we are going to have $\beta = 2^{-n\varepsilon}$ (for some (small) $\varepsilon > 0$, and in particular $1 - \beta \geq \frac{1}{2}$) and $\alpha = 2^{-n}$. There are polynomially-computable *explicit* constructions of hitters (cf. e.g. [30] for an overview) with sufficiently good parameters for our purposes, where

$$\kappa = 2n + O(\log(1/\alpha)) = O(n) \ , \quad t = O(\log(1/\alpha)) = O(n) \ . \tag{5}$$

Appendix D gives further details about a concrete example of a “reasonably”’ cheap construction relying on random walks on constant-degree expander graphs. We will require our hitters to be *injective*, i.e., for any two inputs X and X', there must exist i such that $\mathsf{E}_i(X) \neq \mathsf{E}_i(X')$. It is easy to enforce injectivity for any hitter by just adding $O(\kappa/n)$ functions to the family.

The MT Construction. We now present our construction of an RKA-KDF-secure function, which follows the framework of Maurer and Tessaro [45]. Let $\mathsf{E} = (\mathsf{E}_1, \dots, \mathsf{E}_t)$ be such that $\mathsf{E}_i : \{0,1\}^\kappa \rightarrow \{0,1\}^n$. Moreover, let $F_{i,j} : \{0,1\}^n \rightarrow \{0,1\}^{2\kappa+n}$ for $i \in [t]$ and $j \in [r]$, $G_j : \{0,1\}^n \rightarrow \{0,1\}^\ell$ for $j \in [r]$. For simplicity, denote $F = (F_{i,j})_{i \in [t], j \in [r]}$ and $G = (G_i)_{i \in [t]}$.

The $\mathsf{MT}[\mathsf{E}, F, G]$ construction operates as follow. (Here, $\odot$ denotes multiplication of $(2\kappa + n)$-bit-strings interpreted as elements of the corresponding extension field $\mathbb{F}_{2^{2\kappa+n}}$.)

```
Construction $\mathsf{MT}[F, G](K)$:                         // $K \in \{0,1\}^\kappa$
(1) For all $j \in [r]$, compute
        $S[j] \leftarrow \left( \bigodot_{i=1}^{t} F_{i,j}(\mathsf{E}_i(K)) \right) [1 \dots n]$ .
(2) Compute $K' \leftarrow \bigoplus_{j=1}^{r} G_i(S[i])$ .
(3) Return $K'$.
```

RKA-KDF security. The above construction is indifferentiable from a random oracle [45,22] whenever E is a so-called *input-restricting function family.* While this combinatorial property would also imply RKA-KDF security, explicit constructions of such function families require a very large $t = O(n^c)$ for a large constant c, as discussed in [22].

Here, in contrast, we show that for RKA-KDF security it is sufficient if E is a good hitter. The following theorem summarizes the concrete parameters of our result. The complete proof is found in Appendix E. We give some intuition further below.

Theorem 3 (RKA-KDF-Security of MT). *Let* E *be an* $(\alpha, \beta = q_F/2^n)$*-injective hitter. Moreover, let* $\Phi \subseteq \mathsf{Fcs}(\kappa, \kappa)$ *be a* (γ, λ)*-well behaved set of key transformations. Then, for all adversaries* $\mathcal{A}$ *making* q *queries to* Eval, q_F *queries to the* F*-functions, and* q_G *queries to the* G*-functions,*

$$\mathsf{Adv}^{\mathsf{RKA\text{-}KDF}}_{\mathsf{MT},(F,G),\Phi}(\mathcal{A}) \leq \frac{4rt}{2^n} + q(\alpha + \lambda) + q^2\gamma + q \cdot \left(\frac{q_G + q}{2^n}\right)^r \ .$$

INSTANTIATIONS. Let us target security for $q_F = q = 2^{n(1-\varepsilon)}$ (e.g., $\varepsilon = O(1/n)$), $\ell = n$, and additive attacks $\Phi = \Phi^{\oplus}$ with $\gamma = \lambda = 0$. First note that because we want $\alpha \approx 2^{-n}$ and $\beta = 2^{-\varepsilon n}$, then we can use E with $\kappa = O(n)$ and $t = O(n)$ by (5). Moreover, we need to ensure that $2^{r(1-n)} \cdot 2^{n(1-\varepsilon)(r+1)} < 1$ or alternatively $r(n\varepsilon - 1) > n(1-\varepsilon)$, which is true for $r = r(\varepsilon) = \Omega(\frac{1-\varepsilon}{\varepsilon})$, and $r = O(n)$ for $\varepsilon = O(1/n)$.

Therefore, the construction evaluates a linear number of functions with linear output $O(n)$, or alternatively, $O(n^2)$ single-bit functions $\{0,1\}^n \to \{0,1\}$. This can be turned into evaluating $O(n^2)$ one single function $\{0,1\}^{n+2\log n+O(1)} \to \{0,1\}$.[7] Improving upon this appears a significant barrier.

The MT construction can be combined with the WSN construction above to obtain an RKA-secure block cipher with $(2^{n(1-\varepsilon)}, 2^{n(1-\varepsilon)})$-security via Theorem 2 for any class Φ with small λ, γ.

OVERVIEW OF THE PROOF OF THEOREM 3. We explain here the basic ideas behind the proof of Theorem 3.

To start with, it is convenient to first consider a toy construction, using only t functions $F = (F_i)_{i\in[t]}$ with $F_i \in \mathsf{Fcs}(n, \ell)$, in conjunction with a hitter $\mathsf{E} = (\mathsf{E}_1, \dots, \mathsf{E}_t)$ as above. On input $K \in \{0,1\}^{\kappa}$, it outputs $\bigoplus_{i=1}^{t} F_i(\mathsf{E}_i(K))$. Also, let us only consider $\mathsf{RKA\text{-}KDF}$ attackers which make all q_F of their F queries *beforehand*, and only then query Eval on inputs $\phi_1, \dots, \phi_q$, where the ϕ_i functions are such that $\phi_i(K)$ is uniform for a uniform K.

Assume without loss of generality the uniform key K is sampled after the F-queries have been made. Since E is an $(\alpha, \beta = q_F/2^n)$-hitter, then by the union bound, for every $k \in [q]$ there exists some $i^*(k)$ such that $\mathsf{E}_{i^*(k)}(\phi_k(K))$ was *not* queried to $F_{i^*(k)}$ in the first phase, except with probability $q \cdot \alpha$. Therefore, for all $k \in [q]$, the value $\bigoplus_{i=1}^{t} F_i(\mathsf{E}_i(\phi_k(K)))$ is *individually* uniform, even given the transcript of the F queries, but unfortunately, this does not guarantee independence of these outputs. Indeed, for two k and k', we may well have $i^*(k) = i^*(k')$, and we cannot exclude that for all $i \neq i^*(k)$ both values $F_i(\mathsf{E}_i(\phi_k(K)))$ and $F_i(\mathsf{E}_i(\phi_{k'}(K)))$ are known as part of the F-queries made in the first phase. Then, the output values for k and k' are clearly correlated.

Instead, by using two rounds with functions $(F_{i,j})_{i\in[t],j\in[r]}$ and $(G_j)_{j\in[r]}$ (where $F_{i,j} \in \mathsf{Fcs}(n, n)$ and $G_j \in \mathsf{Fcs}(n, \ell)$), we would generate values $S_k[j] \leftarrow \bigoplus_{i=1}^{t} F_{i,j}(\mathsf{E}_i(\phi_k(K)))$ hoping that, in addition to being individually uniform as above, $S_k[j]$ and $S_{k'}[j]$ are unlikely to collide for any $k \neq k'$.

If the final output of the construction is $\bigoplus_{j=1}^{r} G_j(S_k[j])$, the above would imply security: Indeed, with very high probability, we can show that for every k, there is going to always exist some j^* such that $S_k[j^*]$ was never queried to G_{j^*} previously directly by the attacker (because of the individual uniformity of the value) and that no other $k' \neq k$ is such that $S_{k'}[j^*] = S_k[j^*]$. (Exploiting independence of the $S_k[j]$'s, the probability that such j^* does not exist can be made very small, of the order $\left(\frac{q_G+q}{2^n}\right)^r$.)

There is a final catch. Imagine we are in the above "unfortunate" setting, i.e., for two k and k' and $j \in [r]$, we have $i^*(k) = i^*(k')$, and for all $i \neq i^*(k)$, $F_{i,j}(\mathsf{E}_i(\phi_k(K)))$ and $F_{i,j}(\mathsf{E}_i(\phi_{k'}(K)))$ are known. Then, the fact that $S_k[j]$ and $S_{k'}[j]$ collided is already determined by the transcript of the F queries, independent of $F_{i^*(k),j}(\mathsf{E}_{i^*(k)}(\phi_k(K)))$. Our approach to address this problem is to make the output of the F-values larger (roughly $2\kappa + n$ bits) and to use multiplication. This will make sure that given that any two partial product defined by the F queries as above will not collide (over $2\kappa + n$ bits), and thus (by the fact that multiplication with truncation gives a universal hash function), the final products, truncated at n bits, will also be unlikely to collide.

[7] Note that we can play a bit with parameters, and given a function $F : \{0,1\}^n \to \{0,1\}$, interpret it as a function $\{0,1\}^{n'+2\log(n')} \to \{0,1\}$ for a suitable n' only marginally smaller than n, and obtain an instantiation of our construction with respect to n' still making roughly $O(n^2)$ calls to F.

References

1. C. Adams, *RFC 2144 - The CAST-128 Encryption Algorithm.* Internet Activities Board, May 1997.
2. E. Andreeva, A. Bogdanov, Y. Dodis, B. Mennink, and J. P. Steinberger, "On the indifferentiability of key-alternating ciphers," in *CRYPTO 2013, Part I*, vol. 8042 of *LNCS*, pp. 531–550, Aug. 2013.
3. M. Barbosa and P. Farshim, "The related-key analysis of Feistel constructions," in *FSE 2014*, vol. 8540 of *LNCS*, pp. 265–284, Mar. 2015.
4. M. Bellare and D. Cash, "Pseudorandom functions and permutations provably secure against related-key attacks," in *CRYPTO 2010*, vol. 6223 of *LNCS*, pp. 666–684, Aug. 2010.
5. M. Bellare and T. Kohno, "A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications," in *EUROCRYPT 2003*, vol. 2656 of *LNCS*, pp. 491–506, May 2003.
6. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *ACM CCS 93*, pp. 62–73, Nov. 1993.
7. M. Bellare and P. Rogaway, "The security of triple encryption and a framework for code-based game-playing proofs," in *EUROCRYPT 2006*, vol. 4004 of *LNCS*, pp. 409–426, May / June 2006.
8. E. Biham, "New types of cryptanalytic attacks using related keys," *Journal of Cryptology*, vol. 7, no. 4, pp. 229–246, 1994.
9. E. Biham, O. Dunkelman, and N. Keller, "A related-key rectangle attack on the full KASUMI," in *ASIACRYPT 2005*, vol. 3788 of *LNCS*, pp. 443–461, Dec. 2005.
10. E. Biham, O. Dunkelman, and N. Keller, "Related-key impossible differential attacks on 8-round AES-192," in *CT-RSA 2006*, vol. 3860 of *LNCS*, pp. 21–33, Feb. 2006.
11. A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir, "Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds," in *EUROCRYPT 2010*, vol. 6110 of *LNCS*, pp. 299–319, May 2010.
12. A. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," in *ASIACRYPT 2009*, vol. 5912 of *LNCS*, pp. 1–18, Dec. 2009.
13. A. Biryukov, D. Khovratovich, and I. Nikolic, "Distinguisher and related-key attack on the full AES-256," in *CRYPTO 2009*, vol. 5677 of *LNCS*, pp. 231–249, Aug. 2009.
14. A. Bogdanov, L. R. Knudsen, G. Leander, F.-X. Standaert, J. P. Steinberger, and E. Tischhauser, "Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations - (extended abstract)," in *EUROCRYPT 2012*, vol. 7237 of *LNCS*, pp. 45–62, Apr. 2012.
15. S. Chen, R. Lampe, J. Lee, Y. Seurin, and J. P. Steinberger, "Minimizing the two-round Even-Mansour cipher," in *CRYPTO 2014, Part I*, vol. 8616 of *LNCS*, pp. 39–56, Aug. 2014.
16. S. Chen and J. P. Steinberger, "Tight security bounds for key-alternating ciphers," in *EUROCRYPT 2014*, vol. 8441 of *LNCS*, pp. 327–350, May 2014.
17. B. Cogliati, R. Lampe, and Y. Seurin, "Tweaking Even-Mansour ciphers," in *CRYPTO 2015, Part I*, LNCS, pp. 189–208, Aug. 2015.
18. B. Cogliati, J. Patarin, and Y. Seurin, "Security amplification for the composition of block ciphers: Simpler proofs and new results," in *SAC 2014*, vol. 8781 of *LNCS*, pp. 129–146, Aug. 2014.
19. B. Cogliati and Y. Seurin, "On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks," in *EUROCRYPT 2015, Part I*, vol. 9056 of *LNCS*, pp. 584–613, Apr. 2015.
20. J.-S. Coron, J. Patarin, and Y. Seurin, "The random oracle model and the ideal cipher model are equivalent," in *CRYPTO 2008*, vol. 5157 of *LNCS*, pp. 1–20, Aug. 2008.
21. Y. Dodis, T. Ristenpart, and T. Shrimpton, "Salvaging Merkle-Damgård for practical applications," in *EUROCRYPT 2009*, vol. 5479 of *LNCS*, pp. 371–388, Apr. 2009.
22. Y. Dodis and J. P. Steinberger, "Domain extension for MACs beyond the birthday barrier," in *EUROCRYPT 2011*, vol. 6632 of *LNCS*, pp. 323–342, May 2011.
23. O. Dunkelman, N. Keller, and A. Shamir, "Minimalism in cryptography: The Even-Mansour scheme revisited," in *EUROCRYPT 2012*, vol. 7237 of *LNCS*, pp. 336–354, Apr. 2012.
24. S. Dziembowski, K. Pietrzak, and D. Wichs, "Non-malleable codes," in *ICS 2010*, pp. 434–452, Jan. 2010.
25. S. Even and Y. Mansour, "A construction of a cipher from a single pseudorandom permutation," *Journal of Cryptology*, vol. 10, no. 3, pp. 151–162, 1997.
26. P. Farshim and G. Procter, "The related-key security of iterated even-mansour ciphers," in *FSE 2015*, LNCS, 2015.
27. S. Faust, P. Mukherjee, D. Venturi, and D. Wichs, "Efficient non-malleable codes and key-derivation for poly-size tampering circuits," in *EUROCRYPT 2014*, vol. 8441 of *LNCS*, pp. 111–128, May 2014.

28. P. Gaži and S. Tessaro, “Secret-key cryptography from ideal primitives: A systematic overview,” in *IEEE Information Theory Workshop - ITW 2015*, 2015.
29. C. Gentry and Z. Ramzan, “Eliminating random permutation oracles in the Even-Mansour cipher,” in *ASIACRYPT 2004*, vol. 3329 of *LNCS*, pp. 32–47, Dec. 2004.
30. O. Goldreich, “A sample of samplers - A computational perspective on sampling (survey),” *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 4, no. 20, 1997.
31. C. Guo and D. Lin, “On the indifferentiability of key-alternating Feistel ciphers with no key derivation,” in *TCC 2015, Part I*, vol. 9014 of *LNCS*, pp. 110–133, Mar. 2015.
32. C. Hall, D. Wagner, J. Kelsey, and B. Schneier, “Building PRFs from PRPs,” in *CRYPTO’98*, vol. 1462 of *LNCS*, pp. 370–389, Aug. 1998.
33. V. T. Hoang, B. Morris, and P. Rogaway, “An enciphering scheme based on a card shuffle,” in *CRYPTO 2012*, vol. 7417 of *LNCS*, pp. 1–13, Aug. 2012.
34. T. Holenstein, R. Künzler, and S. Tessaro, “The equivalence of the random oracle model and the ideal cipher model, revisited,” in *43rd ACM STOC*, pp. 89–98, June 2011.
35. J. Kelsey, B. Schneier, and D. Wagner, “Key-schedule cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES,” in *CRYPTO’96*, vol. 1109 of *LNCS*, pp. 237–251, Aug. 1996.
36. R. Lampe, J. Patarin, and Y. Seurin, “An asymptotically tight security analysis of the iterated even-mansour cipher,” in *ASIACRYPT 2012*, vol. 7658 of *LNCS*, pp. 278–295, Dec. 2012.
37. R. Lampe and Y. Seurin, “How to construct an ideal cipher from a small set of public permutations,” in *ASIACRYPT 2013, Part I*, vol. 8269 of *LNCS*, pp. 444–463, Dec. 2013.
38. R. Lampe and Y. Seurin, “Security analysis of key-alternating Feistel ciphers,” in *FSE 2014*, vol. 8540 of *LNCS*, pp. 243–264, Mar. 2015.
39. M. Luby and C. Rackoff, “How to construct pseudo-random permutations from pseudo-random functions (abstract),” in *CRYPTO’85*, vol. 218 of *LNCS*, p. 447, Aug. 1986.
40. S. Lucks, “Ciphers secure against related-key attacks,” in *FSE 2004*, vol. 3017 of *LNCS*, pp. 359–370, Feb. 2004.
41. A. Mandal, J. Patarin, and V. Nachef, “Indifferentiability beyond the birthday bound for the xor of two public random permutations,” in *INDOCRYPT 2010*, vol. 6498 of *LNCS*, pp. 69–81, Dec. 2010.
42. A. Mandal, J. Patarin, and Y. Seurin, “On the public indifferentiability and correlation intractability of the 6-round Feistel construction,” in *TCC 2012*, vol. 7194 of *LNCS*, pp. 285–302, Mar. 2012.
43. U. M. Maurer, K. Pietrzak, and R. Renner, “Indistinguishability amplification,” in *CRYPTO 2007*, vol. 4622 of *LNCS*, pp. 130–149, Aug. 2007.
44. U. M. Maurer, R. Renner, and C. Holenstein, “Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology,” in *TCC 2004*, vol. 2951 of *LNCS*, pp. 21–39, Feb. 2004.
45. U. M. Maurer and S. Tessaro, “Domain extension of public random functions: Beyond the birthday barrier,” in *CRYPTO 2007*, vol. 4622 of *LNCS*, pp. 187–204, Aug. 2007.
46. B. Mennink and B. Preneel, “On the xor of multiple random permutations,” in *Applied Cryptography and Network Security – ACNS 2015*, 2015.
47. B. Morris and P. Rogaway, “Sometimes-recurse shuffle - almost-random permutations in logarithmic expected time,” in *EUROCRYPT 2014*, vol. 8441 of *LNCS*, pp. 311–326, May 2014.
48. J. Patarin, “The “coefficients H” technique (invited talk),” in *SAC 2008*, vol. 5381 of *LNCS*, pp. 328–345, Aug. 2009.
49. T. Ristenpart and S. Yilek, “The mix-and-cut shuffle: Small-domain encryption secure against N queries,” in *CRYPTO 2013, Part I*, vol. 8042 of *LNCS*, pp. 392–409, Aug. 2013.
50. J. Steinberger, “Improved security bounds for key-alternating ciphers via hellinger distance.” Cryptology ePrint Archive, Report 2012/481, 2012. http://eprint.iacr.org/2012/481.
51. S. P. Vadhan, “Pseudorandomness,” *Foundations and Trends in Theoretical Computer Science*, vol. 7, no. 1-3, pp. 1–336, 2012.

A Indifferentiability

We briefly review the notion of indifferentiability by Maurer et al [44] as needed in this paper.

Let $\mathsf{C}[G] : \{0,1\}^m \to \{0,1\}^\ell$ be a construction from a function $G : \{0,1\}^a \to \{0,1\}^b$. We say that C is *indifferentiable* from a random function if $\mathsf{C}[G]$, for $G \xleftarrow{\$} \mathsf{Fcs}(a,b)$, is “as good as” a randomly chosen function $F \xleftarrow{\$} \mathsf{Fcs}(m,\ell)$ in a setting where an adversary is given access to *both* $\mathsf{C}[G]$ and the underlying function G. This is formalized by requiring the existence of a simulator $\mathcal{S}$, accessing F, which mimics the behavior of G in a way that makes real and ideal worlds indistinguishable.

FORMAL DEFINITION. For an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$, the *indifferentiability* advantage is

$$\mathsf{Adv}^{\mathsf{indiff}}_{\mathsf{C}[G],G,\mathcal{S}}(\mathcal{A}) = \mathsf{Pr}\left[G \xleftarrow{\$} \mathsf{Fcs}(a,b) : \mathcal{A}^{\mathsf{C}[G],G} \Rightarrow 1\right] - \mathsf{Pr}\left[F \xleftarrow{\$} \mathsf{Fcs}(m,\ell) : \mathcal{A}^{F,\mathcal{S}^F} \Rightarrow 1\right] .$$

Similarly, for a construction $\mathsf{C}[\pi]$ from a permutation $\pi \in \mathsf{Perms}(a)$, we define

$$\mathsf{Adv}^{\mathsf{indiff}}_{\mathsf{C}[\pi],\pi,\mathcal{S}}(\mathcal{A}) = \mathsf{Pr}\left[\pi \xleftarrow{\$} \mathsf{Perms}(a) : \mathcal{A}^{\mathsf{C}[\pi],\pi,\pi^{-1}} \Rightarrow 1\right] - \mathsf{Pr}\left[F \xleftarrow{\$} \mathsf{Fcs}(m,\ell) : \mathcal{A}^{F,\mathcal{S}^F} \Rightarrow 1\right] .$$

Note that in the latter case, the simulator $\mathcal{S}$ simulates both the behavior of π and π^{-1} queries. We are going to call queries to the first oracle (i.e., either $\mathsf{C}[G]$, $\mathsf{C}[\pi]$ or F) *construction queries*, and queries to the second oracle (either G, π, π^{-1}, or $\mathcal{S}^F$) *primitive queries.*

In this paper, we are going to only consider an information-theoretic version of indifferentiability.

Definition 2 (Indifferentiability). *A construction $\mathsf{C}[\Sigma]$ (where Σ is either a permutation or a function) is (ε, s)-*indifferentiable *from a random function if there exists a simulator $\mathcal{S}$ such that for all adversary $\mathcal{A}$ making q construction queries, and q_Σ primitive queries, $\mathsf{Adv}^{\mathsf{indiff}}_{\mathsf{C}[\Sigma],\Sigma,\mathcal{S}}(\mathcal{A}) \leq \varepsilon(q, q_\Sigma)$, and where additionally, upon each invocation via a primitive queries, the simulator Σ makes at most s queries. Moreover, the simulator answers each query in time polynomial in q_Σ.*

We say that $\mathsf{C}[\Sigma]$ is *perfectly indifferentiable* if it is $(0, 1)$-indifferentiable.

COMPOSITION THEOREM. We use the following fact below, which follows from general composition theorems [44,21] adapted to the specific case of block ciphers considered in this paper.

Theorem 4 (Composition theorem for block ciphers). *Let $\mathsf{BC} = \mathsf{BC}[F]$ be a (κ, n)-block cipher making at most t calls to a function $F : \{0,1\}^m \to \{0,1\}^\ell$, and let $\mathsf{C}[\Sigma]$ be a construction using a primitive Σ which is (ε, s)-indifferentiable from a random function. Consider the (κ, n)-block cipher $\mathsf{BC}' = \mathsf{BC}'[\Sigma] = \mathsf{BC}[\mathsf{C}[\Sigma]]$, i.e., calls to F are replaced by calls to $\mathsf{C}[\Sigma]$. Then,*

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC}'[\Sigma],\Sigma}(q, q_\Sigma) \leq \mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC}[F],F}(q, s \cdot q_\Sigma) + 2 \cdot \varepsilon(t \cdot q, q_\Sigma) .$$

B From Permutations to Functions

In this section, we revisit the security of a construction by Hall, Wagner, Kelsey, and Schneier [32] to build a random function $F : \{0,1\}^n \to \{0,1\}^c$ from a permutation $\pi : \{0,1\}^{n+c} \to \{0,1\}^{n+c}$. In particular, here we show that their construction achieves the stronger notion of *perfect indifferentiability* defined above in Appendix A, and thus can be used to replace (in a black-box way) the function F in the WSN construction. Note that in [32], only indistinguishability was shown. We believe that this result is of interest beyond the scope of this paper.

THE CONSTRUCTION. Let $\pi : \{0,1\}^{n+c} \to \{0,1\}^{n+c}$ be a permutation. The 2^c-query construction $\mathsf{F}_C[\pi] : \{0,1\}^n \to \{0,1\}^c$ proceeds as follows, on input $X \in \{0,1\}^n$: It outputs the c-bit value Z^* such that $\pi(X \parallel Z^*)$ is the smallest element in $\{\pi(X \parallel Z) : Z \in \{0,1\}^c\}$, where smallest is according to lexicographic order. (Or any other total order on strings.)

SECURITY. The following theorem establishes security of F in terms of indifferentiability.[8]

Theorem 5 (Indifferentiability of F). *The construction* $\mathsf{F}_c = \mathsf{F}_c[\pi]$ *is* perfectly *indifferentiable from a random function.*

Proof. We need to prove that there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{indiff}}_{\mathsf{F},\pi,\mathcal{S}}(\mathcal{A}) = 0$ for all adversaries $\mathcal{A}$, and moreover, $\mathcal{S}$ simulates a permutation from $\mathsf{Perms}(n+c)$, together with its inverse, and makes at most one single query to a given function $F \xleftarrow{\$} \mathsf{Fcs}(n,c)$ upon each invocation.

To help with the definition of the simulator, for a function $f \in \mathsf{Fcs}(n,c)$ and a permutation $\tau \in \mathsf{Perms}(n+c)$, we define a new permutation $\pi[\tau,f] \in \mathsf{Perms}(n+c)$. To this end, for every $x \in \{0,1\}^n$, we define

$$y^*_x = \min\{\tau(x \,\|\, z) \;:\; z \in \{0,1\}^c\}$$

and $y_x = \tau(x \,\|\, f(x))$. Note that y^*_x is the output of τ on input $x \,\|\, \mathsf{F}_c[\tau](x)$ and thus if $f = \mathsf{F}_c[\tau]$, $y_x = y^*_x$. The permutation $\pi[\tau,f]$ is such that

$$\pi[\tau,f](x \,\|\, z) = \begin{cases} y^*_x & \text{if } \tau(x \,\|\, z) = y_x, \text{ i.e., } f(x) = z \\ y_x & \text{if } \tau(x \,\|\, z) = y^*_x \\ \tau(x \,\|\, z) & \text{else.} \end{cases}$$

In other words, $\pi[\tau,f]$ re-arranges τ to assign $\pi[\tau,f](x \,\|\, f(x))$ the *smallest* value among $\tau(x \,\|\, z')$ for $z' \in \{0,1\}^c$. Clearly, given τ, $\pi[\tau,f](x \,\|\, z)$ can be computed with a single query to f and 2^c queries to τ. Moreover, note that the inverse $\pi^{-1}[\tau,f]$ is

$$\pi^{-1}[\tau,f](y) = \begin{cases} \tau^{-1}(y_x) & \text{if } y = y^*_x \\ \tau^{-1}(y^*_x) & \text{if } y = y_x \\ \tau^{-1}(y) & \text{else.} \end{cases}$$

Note that the check $y = y^*_x$ and $y = y_x$ can be implemented by first computing $\tau^{-1}(y)$, which returns $x \,\|\, z$, and then querying $\tau(x \,\|\, z')$ for all $z' \neq z$, as well as $f(x)$. In particular, $\pi^{-1}[\tau,f]$ can also be evaluated with one query to f, given τ.

The simulator $\mathcal{S}$ now simply does the following when given oracle access to f: It maintains a random permutation $\tau \xleftarrow{\$} \mathsf{Perms}(n+c)$ (implemented via lazy sampling), and on a forward query $x \,\|\, z$, replies as $\pi[\tau,f](x \,\|\, z)$, and on inverse query y it replies as $\pi^{-1}[\tau,f](y)$. By the above, this requires one f query per evaluation.

Therefore, to prove perfect indifferentiability, it is enough to prove that $(\mathsf{F}_c[\pi],\pi)$ (for $\pi \xleftarrow{\$} \mathsf{Perms}(n+c)$) and $(f,\pi[\tau,f])$ (for $f \xleftarrow{\$} \mathsf{Fcs}(n,c)$ and $\tau \xleftarrow{\$} \mathsf{Perms}(n+c)$) are identically distributed. This can be done in two steps:

1. First, note that $\mathsf{F}_c[\pi[\tau,f]] = f$. This is because on input x, F_c outputs z such that $\pi[\tau,f](x \,\|\, z)$ is smallest. This must be $z = f(x)$, because $\pi[\tau,f]$ is such that $\pi[\tau,f](x \,\|\, f(x)) = y^*_x$, which is the smallest value among $\tau(x \,\|\, z')$, and thus also among $\pi[\tau,f](x \,\|\, z')$.
2. Therefore, it suffices to show that the permutation $\pi[\tau,f]$ is uniformly distributed. This is because $\pi[\tau,f]$ is obtained by sampling a random permutation τ, and then for all x, swapping y^*_x with the output of $x \,\|\, z$ for a randomly chosen $z = f(x)$. This gives a uniform random permutation.

This concludes the proof. □

[8] We note that a previous version of this paper had a somewhat more cumbersome yet equivalent description of the simulator. The simpler and far more elegant description using $\pi[\tau,f]$ was suggested to us by an anonymous reviewer we wish to thank.

C Proof of Lemma 3

For every $a \in S_{t,i}$, we now define now two subsets partitioning $\{0,1\}^n \times \{0,1\}^n$, i.e., the key space for round $t+1$:

$$\mathcal{WK}_a^+ := \{(w,k) \, : \, a \oplus k \in S_{t,i} \; \wedge \; \max\{a \oplus k, a\} \oplus w \notin \mathcal{Q}\}$$
$$\mathcal{WK}_a^- := \{(w,k) \, : \, a \oplus k \notin S_{t,i} \; \vee \; \max\{a \oplus k, a\} \oplus w \in \mathcal{Q}\}$$

It is easy to see that

$$\left|\mathcal{WK}_a^+\right| = N_i \cdot (N-Q) \ , \quad \left|\mathcal{WK}_a^-\right| = N^2 - N_i \cdot (N-Q)$$

because for every a we have exactly $|S_{t,i}| = N_i$ values of k such that $a \oplus k \in S_{t,i}$, and moreover, we have (for each such value k) exactly $N-Q$ possible values of w with $\max\{a, a \oplus k\} \oplus w \notin \mathcal{Q}$. Also, note that for $(w,k) \in \mathcal{WK}_a^-$,

$$\mathsf{E}\left[(p_{t+1,i}(\varphi(a)) - 1/N_i)^2 \;\middle|\; \mathbf{K}[t+1] = k, \mathbf{W}[t+1] = w\right] = p_{t,i}(a)^2 \ ,$$

whereas for $(w,k) \in \mathcal{WK}_a^+$,

$$\mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \frac{1}{N_i}\right)^2 \;\middle|\; \mathbf{K}[t+1] = k, \mathbf{W}[t+1] = w\right] \quad = \quad \left(\frac{p_{t,i}(a) + p_{t,i}(a \oplus k)}{2} - \frac{1}{N_i}\right)^2 \quad .$$

Putting all of this together, we obtain

$$\begin{aligned}
&\mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \tfrac{1}{N_i}\right)^2\right] = \\
&\qquad = \tfrac{1}{N^2} \sum_{k,w} \mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \tfrac{1}{N_i}\right)^2 \;\middle|\; \mathbf{K}[t+1] = k, \mathbf{W}[t+1] = w\right] \\
&\qquad = \frac{1}{N^2}\left[\sum_{(w,k) \in \mathcal{WK}_a^-} \left(p_{t,i}(a) - \tfrac{1}{N_i}\right)^2 + \sum_{(w,k) \in \mathcal{WK}_a^+} \left(\tfrac{p_{t,i}(a) + p_{t,i}(a \oplus k)}{2} - \tfrac{1}{N_i}\right)^2\right] \\
&\qquad\qquad = \left(1 - \tfrac{N_i(N-Q)}{N^2}\right)\left(p_{t,i}(a) - \tfrac{1}{N_i}\right)^2 + \tfrac{N-Q}{N^2} \sum_{y \in S_{t,i}} \left(\tfrac{p_{t,i}(a) + p_{t,i}(y)}{2} - \tfrac{1}{N_i}\right)^2 \ ,
\end{aligned}$$

where we have used the structure of $\mathcal{WK}_a^+$, and the fact that for every $y \in S_{t,i}$ there exists k such that $a \oplus k = y$, and corresponding $N-Q$ values of w. In particular, we can expand

$$\begin{aligned}
\sum_{y \in S_{t,i}} \left(\tfrac{p_{t,i}(a) + p_{t,i}(y)}{2} - \tfrac{1}{N_i}\right)^2 &= \tfrac{1}{4} \sum_{y \in S_{t,i}} \left(\left(p_{t,i}(a) - \tfrac{1}{N_i}\right) + \left(p_{t,i}(y) - \tfrac{1}{N_i}\right)\right)^2 \\
&= \tfrac{N_i}{4} \cdot \left(p_{t,i}(a) - \tfrac{1}{N_i}\right)^2 + \tfrac{1}{4} \Delta_{t,i} \ ,
\end{aligned}$$

where we have used in passing the fact that $\sum_{y \in S_{t,i}} (p_{t,i}(a) - \frac{1}{N_i}) = 0$. When we plug this back into the above, we then get

$$\mathsf{E}\left[\left(p_{t+1,i}(\varphi(a)) - \tfrac{1}{N_i}\right)^2\right] \quad = \quad \left(1 - \tfrac{3}{4} \tfrac{N_i(N-Q)}{4 \cdot N^2}\right)\left(p_{t,i}(a) - \frac{1}{N_i}\right)^2 \quad + \quad \tfrac{1}{4} \tfrac{N-Q}{N^2} \Delta_{t,i} \quad .$$

This concludes the proof of Lemma 3.

D Hitters based on Random Walks

One fairly efficient way to build suitable hitters for our purposes is to use random walks on a constant-degree expander graph.

Expanders and random walks. In particular, we consider D-regular (I.e., every vertex has degree D) undirected graphs $G = (V, E)$ where $V = \{0,1\}^n$, which in particular are explicit, i.e., given $v \in V$ and $i \in [D]$, the i-th neighbor of v can be computed efficiently. As usual, we denote by $\lambda(G)$ the *second* largest eigenvalue (in absolute value, and normalized by D) of the adjacency matrix of G, and call $1 - \lambda(G)$ the *spectral gap*. (Note that we always have $\lambda(G) \leq 1$, with $\lambda(G) = 1$ corresponding to the situation that G is bipartite.)

A *random walk of length* t on G is a sequence of random variables $X_1, \ldots, X_t$, where $X_1 \overset{\$}{\leftarrow} \{0,1\}^n$, and X_i is obtained by selecting uniformly a random neighbor of X_{i-1} for all $i \in [t]$. In particular, representing such a random walk requires only $n + (t-1)\log D$ random bits, because the graph is D-regular.

The following theorem is a standard result in expander-graph theory. (Cf. e.g. Vadhan's excellent monograph on pseudorandomness for a proof [51].)

Theorem 6. *Let $G = (V, E)$ be a D-regular graph, then for any set $B \subseteq V$ with $|B|/|V| \leq \delta$, and a random walk $X_1, X_2, \ldots, X_t$ on G,*

$$\mathsf{Pr}\left[\forall i \in [t] : X_i \in B\right] \leq \left(\delta + \lambda(G)(1-\delta)\right)^t .$$

Example: Hitter constructions. One particular example of a computationally simple graph which is sufficient to give us a good hitter is given by the Margulis-Gabber-Galil construction already. The parameters are not great, but as this perhaps the *simplest* expander constructions, we show that obtain (asymptotically) reasonable parameters even for this simple construction.

The graph is over $V = \{0,1\}^{n/2} \times \{0,1\}^{n/2}$, and we interpret its vertices (x, y) as elements of $\mathbb{Z}_{2^{n/2}} \times \mathbb{Z}_{2^{n/2}}$, and each vertex (x, y) is connected to the 8 vertices

$$(x \pm 2y, y), (x \pm (2y+1), y), (x, y \pm 2x), (x, y \pm (2x+1)).$$

It is well known that for this graph we have $\lambda(G) \leq 5/8\sqrt{2} < 0.9$. Note that representing a random walk of length t takes $n + 3t$ bits, and given a set B such that $|B| \leq \frac{1}{2} \cdot 2^n$, in order to ensure the (δ, ε)-hitting property for say $\varepsilon = 2^n$, we need

$$(0.95)^t \leq \varepsilon ,$$

and hence it is sufficient to set

$$t = \log(1/\varepsilon)/\log(1/0.95) \leq 14n .$$

In particular, this gives a family of $t \leq 14n$ functions, where $\mathsf{E}_i : \{0,1\}^{43n} \to \{0,1\}^n$, and $\mathsf{E}_i(X)$ outputs the i-th step of the random walk on G.

Note that the above parameters can be substantially improved by using better expander graphs.

E Proof of Theorem 3

The proof of Theorem 3 proceeds via a sequence of games, which we first describe and then analyze. A formal pseudo-code description of these games is provided in Figures 2 and 3. In these descriptions, the notation $X \overset{\cup}{\leftarrow} \{a\}$ is a shorthand for $X \leftarrow X \cup \{a\}$.

```
Procedure MAIN:                                    // $G_0$ - $G_3$
$\mathcal{D} \leftarrow \emptyset$
$K \xleftarrow{\$} \{0,1\}^\kappa$
$b \xleftarrow{\$} \mathcal{A}^{\mathsf{F},\mathsf{G},\mathsf{Eval}}$
Return $b$

Procedure $\mathsf{Eval}(\phi)$:                   // $G_0$
If $\phi \notin \mathcal{D}$ then
  $\mathcal{D} \xleftarrow{\cup} \{\phi\}$
  For all $j \in [r]$ do
    $S[j] \leftarrow \left(\bigodot_{i=1}^{t} \mathsf{F}_{i,j}(\mathsf{E}_i(\phi(K)))\right)^{\leq n}$
    If $\mathsf{T}_G[j,S[j]] = \perp$ then
      $\mathsf{T}_G[j,S[j]] \xleftarrow{\$} \{0,1\}^\ell$
  $\mathsf{T}[\phi] \leftarrow \bigoplus_{j=1}^{r} \mathsf{T}_G[j,S[j]]$
Return $\mathsf{T}[\phi]$

Procedure $\mathsf{F}_{i,j}(x)$:                   // $G_0$ - $G_3$
If $\mathsf{T}_F[i,j,x] = \perp$ then
  $\mathsf{T}_F[i,j,x] \xleftarrow{\$} \{0,1\}^{2\kappa+n}$
Return $\mathsf{T}_F[i,j,x]$
```

```
Procedure $\mathsf{Eval}(\phi)$:                   // $G_1$
If $\phi \notin \mathcal{D}$ then
  $\mathcal{D} \xleftarrow{\cup} \{\phi\}$
  For all $j \in [r]$ do
    $S[j] \leftarrow \left(\bigodot_{i=1}^{t} \mathsf{F}_{i,j}(\mathsf{E}_i(\phi(K)))\right)^{\leq n}$
    $\mathcal{J} \leftarrow \{j \in [r] \,:\, \mathsf{T}_G[j,S[j]] = \perp\}$
  $\mathsf{T}[\phi] \xleftarrow{\$} \{0,1\}^\ell$
  If $\mathcal{J} = \emptyset$ then
    $\mathsf{T}[\phi] \leftarrow \bigoplus_{j=1}^{r} \mathsf{T}_G[j,S[j]]$
  Else
    $Z \leftarrow \mathsf{T}[\phi] \oplus \bigoplus_{j \notin \mathcal{J}} \mathsf{T}_G[j,S[j]]$
    $(Y_j)_{j \in \mathcal{J}} \xleftarrow{\$} \left\{(y_j)_{j \in \mathcal{J}} \,:\, \bigoplus_{j \in \mathcal{J}} y_j = Z\right\}$
    For all $j \in \mathcal{J}$ do
      $\mathsf{T}_G[j,S[j]] \leftarrow Y_j$
Return $\mathsf{T}[\phi]$

Procedure $\mathsf{G}_j(y)$:                       // $G_0, G_1$
If $\mathsf{T}_G[j,y] = \perp$ then
  $\mathsf{T}_G[j,y] \xleftarrow{\$} \{0,1\}^\ell$
Return $\mathsf{T}_G[j,y]$
```

Fig. 2. Description of games in proof of Theorem 3. Procedures MAIN and $\mathsf{F}_{i,j}$ for $i \in [t]$ and $j \in [r]$ are common to all games G_0 - G_3. Procedures Eval and G are specified for games G_0 and G_1 only.

The main sequence of games. We start with Game G_0, which represents the real world. There, T_F and T_G are the function tables of the F and G functions, i.e., $\mathsf{T}_F[i,j,x]$ and $\mathsf{T}_G[j,y]$ for $x, y \in \{0,1\}^n$, $i \in [t]$ and $j \in [r]$, are meant to store the output values of $F_{i,j}(x)$ and $G_j(y)$ sampled lazily by the procedures $\mathsf{F}_{i,j}$ and G_j. Similarly, $\mathsf{T}[\phi]$ keeps the value returned by Eval on input $\phi \in \Phi$. The set $\mathcal{D}$ stores the values ϕ for which $\mathsf{T}[\phi]$ has been set, and note that for ease of presentation we explicitly set T_G-values within Eval, even though we could equally well call G_j for $j \in [r]$.

The next game, Game G_1, although seemingly more involved, preserves the functionality of G_0 while moving closer to the ideal world. Only the way in which $\mathsf{Eval}(\phi)$ queries are replied is changed: Now, the value $\mathsf{T}[\phi]$ is set foremost to a random value. Then, we compute the set $\mathcal{J}$ of indices $j \in [r]$ for which the value $\mathsf{T}_G[j,S[j]]$ is unset. If $\mathcal{J}$ happens to be empty, then $\mathsf{T}[\phi]$ is *rewritten* to its actual value. However, otherwise, the procedure samples random values for the unset $\mathsf{T}_G[j,S[j]]$'s with the property that they all add up (together with the $\mathsf{T}_G[j,S[j]]$ values which are already set) to the randomly chosen $\mathsf{T}[\phi]$.

Game G_2 is similar to Game G_1, with two differences: The values $\mathsf{T}_G[j,S[j]]$ for $j \in \mathcal{J}$ are not set directly in the evaluation of $\mathsf{Eval}(\phi)$ any more. Rather, we first set $\mathsf{T}'_G[j,S[j]]$ to the corresponding Y_j, and only within a later $\mathsf{G}_j(y)$ query, if $\mathsf{T}'_G[j,y] \neq \perp$ but $\mathsf{T}_G[j,y] = \perp$, we set $\mathsf{T}_G[j,y] \leftarrow \mathsf{T}'_G[j,y]$. Note in particular that $\mathsf{T}_G[j,y]$ and $\mathsf{T}'_G[j,y]$, at any point in time, are either equal or at least one of them is $\perp$. We use the notation $\mathsf{T}_G[j,y] \vee \mathsf{T}'_G[j,y]$ to denote $\perp$ if both of the values are $\perp$, and otherwise to equal $z \in \{0,1\}^\ell$ if (at least) one of the two values equal z. Our change does not affect the behavior, provided we re-redefine $\mathcal{J}$ to include those values for which $\mathsf{T}_G[j,j] = \mathsf{T}'_G[j,y] = \perp$ and we replace usages of $\mathsf{T}_G[j,y]$ by $\mathsf{T}_G[j,y] \vee \mathsf{T}'_G[j,y]$.

Another modification in G_2 is the introduction of a flag bad, which is initially set to $\mathtt{false}$. The flag bad turns true when one of the two following things happen:

```
Procedure $\mathsf{Eval}(\phi)$:                              // Game $\boxed{G_2}$, $G_3$
If $\phi \notin \mathcal{D}$ then
 $\mathcal{D} \xleftarrow{\cup} \{\phi\}$
 For all $j \in [r]$ do
  $S[j] \leftarrow \left(\bigodot_{i=1}^{t} \mathsf{F}_{i,j}(\mathsf{E}_i(\phi(K)))\right)^{\leq n}$
 $\mathcal{J} \leftarrow \{j \in [r] \,:\, \mathsf{T}_G[j, S[j]] = \mathsf{T}'_G[j, S[j]] = \perp\}$
 $\mathsf{T}[\phi] \xleftarrow{\$} \{0,1\}^{\ell}$
 If $\mathcal{J} = \emptyset$ then
  $\mathsf{bad} \leftarrow \mathtt{true}$
  $\boxed{\mathsf{T}[\phi] \leftarrow \bigoplus_{j=1}^{r}(\mathsf{T}_G[j, S[j]] \vee \mathsf{T}'_G[j, S[j]])}$
 Else
  $Z \leftarrow \mathsf{T}[\phi] \oplus \bigoplus_{j \notin \mathcal{J}} (\mathsf{T}_G[j, S[j]] \vee \mathsf{T}'_G[j, S[j]])$
  $(Y_j)_{j \in \mathcal{J}} \xleftarrow{\$} \left\{(y_j)_{j \in \mathcal{I}} \,:\, \bigoplus_{j \in \mathcal{J}} y_j = Z\right\}$
  For all $j \in \mathcal{J}$ do
   $\mathsf{T}'_G[j, S[j]] \leftarrow Y_j$
   $\mathsf{Q}_G[j, S[j]] \leftarrow \phi$
Return $\mathsf{T}[\phi]$

Procedure $\mathsf{G}_j(y)$:                                 // Game $G_2$
If $\mathsf{T}_G[j, y] = \perp$ then
 $\mathsf{T}_G[j, y] \xleftarrow{\$} \{0,1\}^{\ell}$
 If $\mathsf{T}'_G[j, y] \neq \perp$ then
  $\phi \leftarrow \mathsf{Q}_G[j, y]$
  If $\forall j' \in [r] \setminus \{j\}$ :
                        $\mathsf{T}_G[j', S_\phi[j']] \neq \perp$ then
   $\mathsf{bad} \leftarrow \mathtt{true}$
  $\mathsf{T}_G[j, y] \leftarrow \mathsf{T}'_G[j, y]$
Return $\mathsf{T}_G[j, y]$

Procedure $\mathsf{G}_j(y)$:                                 // Game $G_3$
If $\mathsf{T}_G[j, y] = \perp$ then
 $\mathsf{T}_G[j, y] \xleftarrow{\$} \{0,1\}^{\ell}$
 If $\mathsf{T}'_G[j, y] \neq \perp$ then
  $\phi \leftarrow \mathsf{Q}_G[j, y]$
  If $\forall j' \in [r] \setminus \{j\}$ :
                        $\mathsf{T}_G[j', S_\phi[j']] \neq \perp$ then
   $\mathsf{bad} \leftarrow \mathtt{true}$
  Else
   $\mathsf{T}_G[j, y] \leftarrow \mathsf{T}'_G[j, y]$
Return $\mathsf{T}_G[j, y]$
```

Fig. 3. Description of games in proof of Theorem 3 (Cont'd). Procedures Eval and G in the definition of games G_2 and G_3. The boxed instruction is only executed in Game G_2, but not in G_3. For $\phi \in \mathcal{D}$, we use the notation $S_\phi[j]$ to denote the n-bit value $S[j]$ defined in the process of computing $\mathsf{Eval}(\phi)$.

- When answering an $\mathsf{Eval}(\phi)$ query, the set $\mathcal{J}$ is empty.
- When answering a query $\mathsf{G}_j(y)$ such that $\mathsf{T}_G[j, y] = \perp$ but $\mathsf{T}'_G[j, y] \neq \perp$ (i.e., this value was set when answering an earlier $\mathsf{Eval}(\phi)$ query, but was not queried directly), all other values $S_\phi[j']$ set while answering the *same* $\mathsf{Eval}(\phi)$ query have been queried already directly to $\mathsf{G}_{j'}$.

Note that in the description in Figure 3, we use the notation $\mathsf{Q}_G[j, y]$ to keep track, for every y such that $\mathsf{T}'_G[j, y] \neq \perp$, of the input ϕ of the eval query which has set this value.

Finally, Game G_3 just modifies G_2 so that during an $\mathsf{Eval}(\phi)$ query $\mathsf{T}[\phi]$ is never overwritten and is always random, and similarly an G_j query always returns a random value, and in particular when bad is set to $\mathtt{true}$, the value $\mathsf{T}_G[j, y]$ is *not* overwritten with $\mathsf{T}'_G[j, y]$.

<u>GAME TRANSITIONS.</u> First off, note that G_0 behaves as the real world, i.e., we have $\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathsf{RKA\text{-}KDF\text{-}0}_{\mathsf{MT}}^{\mathcal{A}} \Rightarrow 1\right]$. Furthermore, despite substantial syntactical differences, Games G_0 and G_1 are identical in their behavior. Indeed, if $\mathcal{J} = \emptyset$, it is clear that an Eval query is answered as in G_0. However, this is true even if $\mathcal{J} \neq 0$. In G_0, we would have sampled random $\mathsf{T}_G[j, S[j]]$ for all $j \in \mathcal{J}$, and then computed $\mathsf{T}[\phi]$ as $\bigoplus_{j=1}^{r} \mathsf{T}_G[j, S[j]]$. However, in this case, the joint distribution of $\mathsf{T}[\phi]$ and $\{\mathsf{T}_G[j, S[j]]\}_{j \in \mathcal{J}}$ is exactly uniform over the set of possible tuples of $|\mathcal{J}| + 1$ values such that $\bigoplus_{j \in \mathcal{J}} \mathsf{T}_G[j, S[j]] \oplus \mathsf{T}[\phi] = \bigoplus_{j \notin \mathcal{J}} \mathsf{T}_G[j, S[j]]$. This distribution can be sampled by choosing $|\mathcal{J}|$ out of $|\mathcal{J}| + 1$ values uniformly at random and independently, and setting the remaining one to satisfy the above constraint. In particular, the choice of which $|\mathcal{J}|$ values are chosen at random is irrelevant, and this is exactly what we exploit to move from G_0 to G_1. In other words, we have $\Pr\left[G_0^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[G_1^{\mathcal{A}} \Rightarrow 1\right]$. It is also not hard to see that, as we argue above, G_2 only postpones the setting of $\mathsf{T}_G[j, S[j]]$ to the first point in time where G_j is queried directly on input $S[j]$, but

otherwise the game behaves identically to G_1, and thus $\mathsf{Pr}\left[G_1^{\mathcal{A}} \Rightarrow 1\right] = \mathsf{Pr}\left[G_2^{\mathcal{A}} \Rightarrow 1\right]$. Finally, the game G_3 behaves as the ideal world, since every query is replied randomly. This is because whenever we pick a random vector $(Y_j)_{j\in\mathcal{J}}$, then at most all but one of these values are ever output as part of G-queries, and thus these answers are also random and independent by the above. Therefore,

$$\mathsf{Pr}\left[G_3^{\mathcal{A}} \Rightarrow 1\right] = \mathsf{Pr}\left[\mathsf{RKA\text{-}KDF\text{-}1}_{\mathsf{MT}}^{\mathcal{A}} \Rightarrow 1\right] \ .$$

Thus,

$$\mathsf{Adv}_{\mathsf{MT}}^{\mathsf{RKA\text{-}KDF}}(\mathcal{A}) = \mathsf{Pr}\left[G_2^{\mathcal{A}} \Rightarrow 1\right] - \mathsf{Pr}\left[G_3^{\mathcal{A}} \Rightarrow 1\right] \ .$$

On the other hand, Games G_2 and G_3 behave identically as long as bad is not set to $\mathtt{true}$, and thus by the fundamental lemma of game playing [7],

$$\mathsf{Adv}_{\mathsf{MT}}^{\mathsf{RKA\text{-}KDF}}(\mathcal{A}) = \mathsf{Pr}\left[G_2^{\mathcal{A}} \Rightarrow 1\right] - \mathsf{Pr}\left[G_3^{\mathcal{A}} \Rightarrow 1\right] \leq \mathsf{Pr}\left[G_3^{\mathcal{A}} \text{ sets } \mathsf{bad}\right] \ . \tag{6}$$

Therefore, the rest of the analysis is devoted to upper bounding $\mathsf{Pr}\left[G_3^{\mathcal{A}} \text{ sets } \mathsf{bad}\right]$. In particular, we prove the following lemma, which is the core of the proof.

Lemma 4 (Bad-event Analysis). *For $\mathcal{A}$ making q Eval queries, q_F F-queries and q_G G-queries,*

$$\mathsf{Pr}\left[G_3^{\mathcal{A}} \textit{ sets } \mathsf{bad}\right] \leq \frac{4rt}{2^n} + q(\alpha + \lambda) + q^2\gamma + q \cdot \left(\frac{q_G + q}{2^n}\right)^r \ .$$

Proof. One key observation to upper bound the probability that $G_3^{\mathcal{A}}$ sets bad is that in Game G_3, all Eval-, F- and G-queries are answered randomly and independently, regardless of whether bad has been set to $\mathtt{true}$ or not during the game. Therefore, we can *equivalently* think of executing Game G_3 until $\mathcal{A}$ outputs its decision bit, and only then check (according to the history) whether bad was set to $\mathtt{true}$ at some point. Now, assume the queries $\phi_1, \ldots, \phi_q$ have been made to Eval in this order, and denote by $S_k[j]$ the value $S[j]$ computed when processing ϕ_k, and let $\mathcal{G}_j$ be the set of inputs of G_j queries made by $\mathcal{A}$. Then, if bad is $\mathtt{true}$ at the end of G_3, there must be a $k \in [q]$ such that for all $j \in [r]$ one of the following two facts is true: (1) There exists $k' < k$ such that $S_{k'}[j] = S_k[j]$, (2) Or $S_k[j] \in \mathcal{G}_j$.

Therefore, it will be simpler to consider Game G depicted in Figure 4 – which captures exactly the above – and the probability that it outputs $\mathtt{true}$. Note in particular that we can defer the computation of the $S_l[j]$ values to the end of $\mathcal{A}$'s interaction, as $\mathcal{A}$'s view is independent of those, as $\mathsf{Eval}(\phi)$ queries are answered randomly. Thus $\mathsf{Pr}\left[G_3^{\mathcal{A}} \text{ sets } \mathsf{bad}\right] \leq \mathsf{Pr}\left[G^{\mathcal{A}} \Rightarrow \mathtt{true}\right]$.

A bit more formally, assume $\mathcal{A}$ makes without loss of generality q Eval-queries, q_F F-queries and q_G G-queries. Then, for every $k \in [q]$ and $j \in [r]$, denote by $\mathsf{BAD}_{k,j}$ the event that there exists $k' < k$ such that $S_k[j] = S_{k'}[j]$ or $S_k[j] \in \mathcal{G}_j$. Denote $\mathsf{BAD}_k := \bigwedge_{j=1}^r \mathsf{BAD}_{k,j}$ and $\mathsf{BAD} = \bigvee_{k=1}^q \mathsf{BAD}_k$. Clearly, $\mathsf{Pr}\left[G^{\mathcal{A}} \Rightarrow \mathtt{true}\right] = \mathsf{Pr}\left[\mathsf{BAD}\right]$.

We are going to define a series of other events that are going to help us in the analysis of $\mathsf{Pr}\left[\mathsf{BAD}\right]$:

The collision event. For all $j \in [r]$, at any point of time during $\mathcal{A}$'s execution, for an input $X \in \{0,1\}^\kappa$, let $\mathcal{I}_{F,j}(X)$ be the set of indices $i \in [t]$ such that $\mathsf{E}_i(X) \in \mathcal{F}_{i,j}$. Here, we define the event COLL_j which occurs if at the end of $\mathcal{A}$'s execution in Game G, right before sampling K, there exist two *distinct* κ-bit inputs $X \neq X'$ with the properties that (1) $\mathcal{I} = \mathcal{I}_{F,j}(X) = \mathcal{I}_{F,j}(X') \neq \emptyset$, (2) there exists $i \in \mathcal{I}$ such that $\mathsf{E}_i(X) \neq \mathsf{E}_i(X')$, and (3) we have a "partial" collision

$$\bigodot_{i\in\mathcal{I}} \mathsf{T}_F[i, j, \mathsf{E}_i(X)] = \bigodot_{i\in\mathcal{I}} \mathsf{T}_F[i, j, \mathsf{E}_i(X')] \ .$$

Moreover, we let $\mathsf{COLL} := \bigvee_{j=1}^r \mathsf{COLL}_j$, i.e., the event that COLL_j occurs for some $j \in [r]$.

```
Procedure MAIN:                                   // Game $G$
$q \leftarrow 0$
For all $i \in [t]$ and $j \in [r]$ do $\mathcal{F}_{i,j} \leftarrow \emptyset$
For all $j \in [r]$ do $\mathcal{G}_j \leftarrow \emptyset$
$b \overset{\$}{\leftarrow} \mathcal{A}^{\mathsf{F},\mathsf{G},\mathsf{Eval}}$
$K \overset{\$}{\leftarrow} \{0,1\}^{\kappa}$
For all $k = 1, \ldots, q$ do
 For all $j \in [r]$ do
  For all $i \in [t]$ do
   If $\mathsf{T}_F[i, j, \mathsf{E}_i(\phi_k(K))] = \perp$ then
    $\mathsf{T}_F[i, j, \mathsf{E}_i(\phi_k(K))] \overset{\$}{\leftarrow} \{0,1\}^{2\kappa+n}$
  $S_k[j] \leftarrow \left(\bigodot_{i=1}^{t} \mathsf{T}_F[i, j, \mathsf{E}_i(\phi_k(K))]\right)[1 \ldots n]$
 If $\forall j \in [r]$: $(\exists k' < k : S_{k'}[j] = S_k[j]) \vee S_k[j] \in \mathcal{G}_j$ then
  Return true
Return false

Procedure $\mathsf{F}_{i,j}(x)$:                  // Game $G$
If $\mathsf{T}_F[i, j, x] = \perp$ then
 $\mathcal{F}_{i,j} \overset{\cup}{\leftarrow} \{x\}$, $\mathsf{T}_F[i, j, x] \overset{\$}{\leftarrow} \{0,1\}^{2\kappa+n}$
Return $\mathsf{T}_F[i, j, x]$

Procedure $\mathsf{G}_j(y)$:                      // Game $G$
If $\mathsf{T}_G[j, y] = \perp$ then
 $\mathcal{G}_j \overset{\cup}{\leftarrow} \{y\}$, $\mathsf{T}_G[j, y] \overset{\$}{\leftarrow} \{0,1\}^{\ell}$
Return $\mathsf{T}_G[j, y]$

Procedure $\mathsf{Eval}(\phi)$:                  // Game $G$
If $\mathsf{T}[\phi] \neq \perp$ then
 $q \leftarrow q + 1$, $\phi_q \leftarrow \phi$, $\mathsf{T}[\phi] \overset{\$}{\leftarrow} \{0,1\}^{\ell}$
Return $\mathsf{T}[\phi]$
```

Fig. 4. Description of Game G in the proof of Theorem 3.

The hitting event. For every $k \in [q]$, consider the event HIT_k that for all $i \in [t]$, we have that $\mathsf{E}_i(\phi(K)) \in \mathcal{F}$, where $\mathcal{F} = \cup_{i,j} \mathcal{F}_{i,j}$. (Note that $|\mathcal{F}| \leq q_F$.) Moreover, denote by $\mathsf{HIT} := \bigvee_{k=1}^{q} \mathsf{HIT}_k$ the event that HIT_k occurs for some $k \in [q]$.

We prove the following three lemmas in Section F, which are used to conclude the proof.

Lemma 5 (Bounding the collision probability). $\Pr[\mathsf{COLL}] \leq \frac{4rt}{2^n}$.

Lemma 6 (Bounding the hitting probability). $\Pr[\mathsf{HIT}] \leq q(\alpha + \lambda)$.

Lemma 7 (Bounding the BAD-probability). $\Pr\left[\mathsf{BAD} \,\middle|\, \overline{\mathsf{COLL}} \wedge \overline{\mathsf{HIT}}\right] \leq q^2\gamma + q \cdot \left(\frac{q_G+q}{2^n}\right)^r$.

To conclude our proof, it is enough now to combined the three lemmas, observing that

$$\Pr[\mathsf{BAD}] \leq \frac{4rt}{2^n} + q(\alpha + \lambda) + q^2\gamma + q \cdot \left(\frac{q_G + q}{2^n}\right)^r \ , \ ,$$

concluding the proof of Lemma 4. □

F Proof of Lemmas

Proof (Of Lemma 5). Fix $j \in [r]$ and fix two κ-bit inputs $X \neq X'$. Assume that COLL_j occurs, i.e., at the end of $\mathcal{A}$'s execution because of a partial collision between X and X', we have (1) $\mathcal{I} = \mathcal{I}_{F,j}(X) = \mathcal{I}_{F,j}(X') \neq \emptyset$, (2) there exists $i \in \mathcal{I}$ such that $\mathsf{E}_i(X) \neq \mathsf{E}_i(X')$, and (3) we have a "partial" collision

$$\bigodot_{i \in \mathcal{I}} \mathsf{T}_F[i, j, \mathsf{E}_i(X)] = \bigodot_{i \in \mathcal{I}} \mathsf{T}_F[i, j, \mathsf{E}_i(X')] \ . \tag{7}$$

There are two cases. First, assume that the above collision in (7) is *not* on $0^{2\kappa+n}$. Then, there must have been a point in time, during $\mathcal{A}$'s execution, where $\mathcal{A}$ has made an F query of the form

$\mathsf{F}_{i^*,j}(\mathsf{E}_{i^*}(X))$ for some $i^* \in [t]$ such that $\mathcal{I}_{F,j}(X) = \mathcal{I}_{F,j}(X') \setminus \{i^*\}$, $\mathsf{E}_{i^*}(X) \neq \mathsf{E}_{i^*}(X')$, and after answering the query

$$\bigodot_{i \in \mathcal{I}_{F,j}(X) \cup \{i^*\}} \mathsf{T}_F[i, j, \mathsf{E}_i(X)] = \bigodot_{i \in \mathcal{I}_{F,j}(X')} \mathsf{T}_F[i, j, \mathsf{E}_i(X')]$$

Or this happen symmetrically for X'. Note that the equality is satisfied in the above situation with probability at most $2^{-2\kappa-n}$ because

$$\bigodot_{i \in \mathcal{I}_{F,j}(X)} \mathsf{T}_F[i, j, \mathsf{E}_i(X)] \neq 0$$

and $\mathsf{T}_F[i^*, j, \mathsf{E}_{i^*}(X)]$ is chosen uniformly. Moreover, there are at most $2t$ chances for this to happen (t chances for X and X' each). Thus, by the union bound, the probability of a partial collision for X and X' on a non-zero value is $2t2^{-2\kappa-n}$. The case that both sides of (7) are 0 is easy to exclude: This means that at least one (in fact two) of the involved $2t$ T_F values is 0. This happens with probability at most $2t2^{-2\kappa-n}$.

Hence, the overall probability that COLL_j occurs because of X and X' is at most $4t2^{-2\kappa-n}$. The final bound on COLL follows by taking the union bound over all X, X' and over all $j \in [r]$. □

Proof (Of Lemma 6). By the union bound, we have

$$\mathsf{Pr}[\mathsf{HIT}] \leq \sum_{k=1}^{q} \mathsf{Pr}[\mathsf{HIT}_k] = \sum_{k=1}^{q} \mathsf{Pr}[\forall i \in [t] : \mathsf{E}_i(\phi(K)) \in \mathcal{F}] \ .$$

However, because Φ is λ-uniform, we can upper bound

$$\mathsf{Pr}[\forall i \in [r] : E_i(\phi(K)) \in \mathcal{F}] \leq \mathsf{Pr}[\forall \in [y : \mathsf{E}_i(K) \in \mathcal{F}] + \lambda \leq \alpha + \lambda \ ,$$

where the last property comes from the fact that $|\mathcal{F}| \leq q_F$ and that E is a $(q_F/2^n, \varepsilon)$-hitter. And therefore, overall, we have

$$\mathsf{Pr}[\mathsf{HIT}] \leq q(\alpha + \lambda) \ ,$$

as desired. □

Proof (Of Lemma 7). To save on space, we assume here tacitly that HIT and COLL do not happen, without mentioning this conditioning explicitly. In fact, we assume that T_F and T_G have been partially fixed arbitrarily during the execution of $\mathcal{A}$ in some arbitrary way (which will be irrelevant for the proof) with the sole constraint that both HIT and COLL have not occurred. In the following, we also assume without loss of generality that $\phi_k(K) \neq \phi_{k'}(K)$ for all $k \neq k'$, and this just results in the term $q^2\gamma$ being added to the bound.

We first upper bound the probability that $\mathsf{BAD}_{k,j}$ happens for some fixed $k \in [q]$ and $j \in [r]$. By the union bound and the definition of $\mathsf{BAD}_{k,j}$,

$$\mathsf{Pr}[\mathsf{BAD}_{k,j}] \leq \sum_{y \in \mathcal{G}_j} \mathsf{Pr}[S_k[j] = y] + \sum_{k' < k} \mathsf{Pr}[S_k[j] = S_{k'}[j]] \ .$$

Let $\mathcal{I}_{k,j}$ be the the set of $i \in [t]$ such that $\mathsf{E}_i(\phi_k(K)) \in \mathcal{F}_{i,j}$. Note that since HIT_j did not happen, then $\mathcal{I}_{k,j}$ is a *proper* subset of $[t]$. Then, also note that

$$S_k[j] = \left(A_{j,k} \odot \bigodot_{i \notin \mathcal{I}_{k,j}} X_{k,j,i} \right) [1 \ldots n] \; .$$

where $X_{k,j,i} := \mathsf{T}_F[i, j, \mathsf{E}_i(\phi_k(K))]$ and $A_{k,j} := \bigodot_{i \in \mathcal{I}_{k,j}} X_{k,j,i}$. Therefore, when computing $S_k[j]$, all values $X_{k,j,i}$ for $i \notin \mathcal{I}_{k,j}$ are generated randomly and freshly *after* $\mathcal{A}$'s execution is terminated, and in particular independent of $\mathcal{G}_j$. Therefore, for any $y \in \mathcal{G}_j$ (and conditioned on $\mathcal{G}_j$ and $\mathcal{F}_{i,j}$ being given for all $i \in [t]$)

$$\mathsf{Pr}\left[S_k[j] = y\right] = \sum_{z \in \{0,1\}^{2\kappa}} \mathsf{Pr}\left[A_{j,k} \odot \bigodot_{i \notin \mathcal{I}_{k,j}} X_{k,j,i} = y \parallel z \right] \leq 2^{2\kappa} \cdot \frac{t}{2^{2\kappa+n}} = \frac{t}{2^n} \; ,$$

by the Schwartz-Zippel Lemma and the fact that $A_{j,k} \odot \bigodot_{i \notin \mathcal{I}_{k,j}} X_{k,j,i} - y \| z$ is a polynomial of degree at most t in the $X_{k,j,i}$'s over the extension field $\mathbb{F}_{2^{2\kappa+n}}$.

Now, for $k' < k$, we note that

$$\mathsf{Pr}\left[S_{k'}[j] = S_k[j]\right] = \sum_{z \in \{0,1\}^{2\kappa}} \mathsf{Pr}\left[A_{j,k} \odot \bigodot_{i \notin \mathcal{I}_{k,j}} X_{k,j,i} \oplus A_{j,k'} \odot \bigodot_{i \notin \mathcal{I}_{k',j}} X_{k',j,i} = 0^n \parallel z \right] \; .$$

Note that we have two cases here: If $\mathcal{I}_{k,j} = \mathcal{I}_{k',j}$ and $\mathsf{E}_i(\phi_k(K)) = \mathsf{E}_i(\phi_{k'}(K))$ for all $i \in \mathcal{I}_{k,j}$, then we must have $A_{j,k} \neq A_{j,k'}$ for otherwise COLL would have happened. Otherwise, the two products differ in at least one variable. Either way, $A_{j,k} \odot \bigodot_{i \notin \mathcal{I}_{k,j}} X_{k,j,i} \oplus A_{j,k'} \odot \bigodot_{i \notin \mathcal{I}_{k',j}} X_{k',j,i} \oplus 0^n \parallel z$ is non-zero polynomial of degree at most $2t$ in the extension field $\mathbb{F}_{2^{2\kappa+n}}$, and thus

$$\mathsf{Pr}\left[S_{k'}[j] = S_k[j]\right] \leq 2^{2\kappa} \cdot \frac{2t}{2^{2\kappa+n}} = \frac{2t}{2^n}$$

by the Schwartz-Zippel Lemma. Therefore, overall we get

$$\mathsf{Pr}\left[\mathsf{BAD}_{k,j}\right] \leq \frac{t(2q + q_G)}{2^n} \; .$$

Now note that for fixed k, the events $\mathsf{BAD}_{k,j}$ are independent, and thus

$$\mathsf{Pr}\left[\mathsf{BAD}_k\right] = \prod_{j=1}^{r} \mathsf{Pr}\left[\mathsf{BAD}_{k,j}\right] \leq \left(\frac{t(2q + q_G)}{2^n} \right)^r \; .$$

And by the union bound, $\mathsf{Pr}\left[\mathsf{BAD}\right] \leq q \left(\frac{t(2q+q_G)}{2^n}\right)^r$. □

G Proof of Lemma 1

The proof is based on the H-coefficient method by Patarin [48], using the notation of Chen and Steinberger [16], and is very similar to the approach of Lempe and Seurin [38].

Here, we give a new proof for completeness considering the slightly more general notion of multi-user PRP security we deal with in this paper.

Setup of the lemma. First, fix a sequence $T_F = (x_1, y_1), \ldots, (x_{q_F}, y_{q_F})$ such that $x_i \in \{0,1\}^m$ and $y_i \in \{0,1\}^\ell$ for all $i \in [q_F]$, and let

$$\boldsymbol{X} = ((i_1, X_1), \ldots, (i_q, X_q))$$

be another sequence such that $i_j \in \mathbb{N}$ and $X_j \in \{0,1\}^n$ for all $j \in [q]$, and for all j, j' such that $i_j = i_{j'}$ and $j \neq j'$, we have $X_j \neq X_{j'}$. Then, for every sequence $\boldsymbol{Y} = (Y_1, \ldots, Y_q)$ such that $Y_j \neq Y_{j'}$ whenever $i_j = i_{j'}$, let $p(\boldsymbol{Y}|\boldsymbol{X}, T_F)$ be the probability $\mathsf{BC}_{K_{i_j}}[F](X_j) = Y_j$ for all $j \in [q]$, where F is a randomly sampled function from $\mathsf{Fcs}(m, \ell)$ with the constraint that $F(x_i) = y_i$ for all $i \in [q_F]$, and $K_1, K_2, \ldots \xleftarrow{\$} \{0,1\}^\kappa$ are independent keys. Also, let $\mathcal{Y}(\boldsymbol{X})$ be the set of such sequences $\boldsymbol{Y}$, and $N(\boldsymbol{X}) = |\mathcal{Y}(\boldsymbol{X})|$.

Note that we can connect these quantities with the statement of the Lemma by noting that for all T_F and for all $\boldsymbol{X}$,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC},F}(\boldsymbol{X}, T_F) &= \frac{1}{2} \sum_{\boldsymbol{Y} \in \mathcal{Y}(\boldsymbol{X})} \left| p(\boldsymbol{Y}|\boldsymbol{X}, T_F) - \frac{1}{N(\mathbf{X})} \right| \\ &\leq \mathsf{Adv}^{\mathsf{NCPAPRP}}_{\mathsf{BC},F}(q, q_F) =: \varepsilon \ .\end{aligned}$$

By an averaging argument this implies that for every $\boldsymbol{X}$ and T_F, there exists some set $\mathcal{Y}'(\boldsymbol{X}, T_F) \subseteq \mathcal{Y}(\boldsymbol{X})$ such that:

- $|\mathcal{Y}'(\boldsymbol{X}, T_F)| \geq (1 - \sqrt{\varepsilon}) \cdot |\mathcal{Y}(\boldsymbol{X})|$
- For all $\boldsymbol{Y} \in \mathcal{Y}'(\boldsymbol{X}, T_F)$,
$$p(\boldsymbol{Y}|\boldsymbol{X}, T_F) \geq (1 - \sqrt{\varepsilon}) \cdot \frac{1}{N(\boldsymbol{X})} \ .$$

We can now continue with the proof.

Transcripts. Let $\mathcal{A}$ an attacker in the (multi-user) PRP game for $\mathsf{BC}[F_1] \circ \mathsf{BC}^{-1}[F_2]$, i.e., in either of PRP-1 or PRP-0. We assume without loss of generality that the attacker is deterministic. Moreover, it makes overall exactly q block cipher queries. Similarly, it makes exactly q_F queries to F_1 and q_F queries to F_2. Finally, it never makes a redundant block cipher query, i.e., if it has made a forward query (i, X), returning Y, it never makes a later backward query (i, Y), and vice versa.

Then, its interaction can be represented by a transcript

$$\tau = ((x^1_1, y^1_1), \ldots, (x^1_{q_F}, y^1_{q_F}), (x^2_1, y^2_1) \ldots, (x^2_{q_F}, y^2_{q_F}), (i_1, \sigma_1, X_1, Y_1), \ldots, (i_q, \sigma_q, X_q, Y_q)) \ ,$$

where $x_1, \ldots, x_q \in \{0,1\}^m$, $y_1, \ldots, y_q \in \{0,1\}^\ell$, $i_1, \ldots, i_q \in \mathbb{N}$, $X_1, Y_1, \ldots, X_q, Y_q \in \{0,1\}^n$, and $\sigma_1, \ldots, \sigma_q \in \{+, -\}$. The meaning of this is that $\mathcal{A}$, in its interaction, has made queries $x^1_1, \ldots, x^1_{q_F}$ to F_1, $x^2_1, \ldots, x^2_{q_F}$ to F_2 (receiving answers $y^1_1, \ldots, y^1_{q_F}$ and $y^2_1, \ldots, y^2_{q_F}$, respectively), and queried the block cipher for all $j \in [q]$ either on (i_j, X_j) as a forward query (if $\sigma_j = +$) and received answer Y_j, or on (i_j, Y_j) as a backward query (if $\sigma_j = -$) and received answer X_j. Note that the order of the queries in the transcript is irrelevant, as the actual ordering can be reconstructed inductively due to the uniqueness of the queries and $\mathcal{A}$ being deterministic.

We now denote by T_{real} and T_{ideal} the random variables representing the transcripts occurring in an interaction with $\mathcal{A}$ in games $\mathsf{PRP}\text{-}1^{\mathcal{A}}_{\mathsf{BC}[F_1] \circ \mathsf{BC}^{-1}[F_2], (F_1, F_2)}$ and $\mathsf{PRP}\text{-}0^{\mathcal{A}}_{\mathsf{BC}[F_1] \circ \mathsf{BC}^{-1}[F_2], (F_1, F_2)}$, respectively. Then, fix the transcript

$$\tau = ((x^1_1, y^1_1), \ldots, (x^1_{q_F}, y^1_{q_F}), (x^2_1, y^2_1) \ldots, (x^2_{q_F}, y^2_{q_F}), (i_1, \sigma_1, X_1, Y_1), \ldots, (i_q, \sigma_q, X_q, Y_q))$$

which is compatible with $\mathcal{A}$ (i.e., it can occur in an interaction with $\mathcal{A}$). Denote

$$T_{F_i} = ((x_1^i, y_1^i), \ldots, (x_{q_F}^i, y_{q_F}^i))$$

for $i = 1, 2$, as well as

$$\boldsymbol{X} = ((i_1, X_1), \ldots, (i_q, X_q)) \ , \quad \boldsymbol{Y} = ((i_1, Y_1), \ldots, (i_q, Y_q)) \ .$$

Moreover, we use the shorthand $\mathsf{Pr}\left[T_{F_i}\right] = \mathsf{Pr}\left[\forall j \in [q_F] : F_j(x_j^i) = y_j^i\right]$ for $i = 1, 2$. Also, let $\mathcal{Z} = \mathcal{Y}'(\boldsymbol{X}, T_{F_1}) \cap \mathcal{Y}'(\boldsymbol{Y}, T_{F_2})$. Then, using the independent of F_1 and F_2, as well as of the keys

$$\begin{aligned}
\mathsf{Pr}\left[T_{\mathsf{real}} = \tau\right] &= \mathsf{Pr}\left[T_{F_1}\right] \cdot \mathsf{Pr}\left[T_{F_2}\right] \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, T_{F_1}) \cdot p(\boldsymbol{Z}|\boldsymbol{Y}, T_{F_2}) \\
&\geq \mathsf{Pr}\left[T_{F_1}\right] \cdot \mathsf{Pr}\left[T_{F_2}\right] \sum_{\boldsymbol{Z} \in \mathcal{Z}} p(\boldsymbol{Z}|\boldsymbol{X}, T_{F_1}) \cdot p(\boldsymbol{Z}|\boldsymbol{Y}, T_{F_2}) \\
&\geq \mathsf{Pr}\left[T_{F_1}\right] \cdot \mathsf{Pr}\left[T_{F_2}\right] (1 - 2\sqrt{\varepsilon}) \frac{|\mathcal{Z}|}{N(\boldsymbol{X})^2} \\
&\geq \mathsf{Pr}\left[T_{F_1}\right] \cdot \mathsf{Pr}\left[T_{F_2}\right] (1 - 2\sqrt{\varepsilon}) \frac{N(\boldsymbol{X})}{N(\boldsymbol{X})^2} \\
&\geq (1 - 4\sqrt{\varepsilon}) \frac{\mathsf{Pr}\left[T_{F_1}\right] \cdot \mathsf{Pr}\left[T_{F_2}\right]}{N(\boldsymbol{X})} = (1 - 4\sqrt{\varepsilon}) \mathsf{Pr}\left[T_{\mathsf{ideal}} = \tau\right] \ .
\end{aligned}$$

To conclude the proof, we use the following simple argument, which is at the core of Patarin's H-coefficient method. Let $\mathcal{T}^+$ be the set of valid transcripts for $\mathcal{A}$ such that $\mathsf{Pr}\left[T_{\mathsf{ideal}} = \tau\right] \geq \mathsf{Pr}\left[T_{\mathsf{real}} = \tau\right]$. Then, using well-known properties of the statistical distance

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{BC}[F_1] \circ \mathsf{BC}[F_2]^{-1}, (F_1, F_2)}(\mathcal{A}) &\leq \mathsf{SD}(T_{\mathsf{real}}, T_{\mathsf{ideal}}) \\
&= \sum_{\tau \in \mathcal{T}^+} \left(\mathsf{Pr}\left[T_{\mathsf{ideal}} = \tau\right] - \mathsf{Pr}\left[T_{\mathsf{real}} = \tau\right]\right) \\
&\leq 4\sqrt{\varepsilon} \sum_{\tau \in \mathcal{T}^+} \mathsf{Pr}\left[T_{\mathsf{ideal}} = \tau\right] \leq 4\sqrt{\varepsilon} \ .
\end{aligned}$$

This concludes the proof of Lemma 1.