Key-alternating Ciphers and Key-length Extension: Exact Bounds and Multi-user Security

Viet Tung Hoang and Stefano Tessaro

Dept. of Computer Science, University of California Santa Barbara

September 6, 2016

Abstract. This paper revisits the concrete security of key-alternating ciphers and key-length extension schemes, with respect to *tightness* and *multi-user security.* The best existing bounds on the concrete security of key-alternating ciphers (Chen and Steinberger, EUROCRYPT '14) are only *asymptotically* tight, and the quantitative gap with the best existing attacks remains numerically substantial for concrete parameters. Here, we prove *exact* bounds on the security of key-alternating ciphers and extend them to XOR cascades, the most efficient construction for key-length extension. Our bounds essentially match, for any possible query regime, the advantage achieved by the best existing attack.

Our treatment also extends to the multi-user regime. We show that the multi-user security of key-alternating ciphers and XOR cascades is very close to the single-user case, i.e., given enough rounds, it does not substantially decrease as the number of users increases. On the way, we also provide the first explicit treatment of multi-user security for key-length extension, which is particularly relevant given the significant security loss of block ciphers (even if ideal) in the multi-user setting.

The common denominator behind our results are new techniques for information-theoretic indistinguishability proofs that both extend and refine existing proof techniques like the H-coefficient method.

Keywords: Symmetric cryptography, block ciphers, provable security, tightness, multi-user security

Table of Contents

1 Introduction . . . 1

2 Preliminaries . . . 5

3 Indistinguishability Proofs via Point-wise Proximity . . . 6

3.1 The indistinguishability framework . . . 6

3.2 Point-wise proximity . . . 8

3.3 From single-user to multi-user security . . . 10

4 Exact Bounds for Key-Alternating Ciphers . . . 12

4.1 Results and Discussion . . . 12

4.2 Proof of Theorem 1 . . . 15

4.3 Multi-user security of KAC . . . 20

5 XOR Cascades . . . 22

A Proof of Lemma 4 . . . 28

B Proof of Lemma 5 . . . 29

C Proof of Theorem 3 . . . 31

D Proof of Proposition 1 . . . 32

E XC's relation with Gaži and Tessaro's 2XOR . . . 36

1 Introduction

Precise bounds on the security of symmetric constructions are essential in establishing when and whether these constructions are to be deployed. This paper revisits the question of proving *best-possible* security bounds for *key-alternating ciphers* and *key-length extension schemes*.

Our contribution is twofold. First, we prove *exact* bounds on the security of key-alternating ciphers and related methods for key-length extensions (i.e, XOR cascades) which essentially match what is achieved by the best-known attack. This is a substantial improvement over previous bounds, which are only *asymptotically* optimal. Second, we extend our treatment to the multi-user setting, where no non-trivial bounds are known to date for these constructions.

Our results are built on top of new *conceptual* insights in information-theoretic indistinguishability proofs, generalizing previous approaches such as the H-coefficient technique [9, 28].

Key-alternating ciphers. *Key-alternating ciphers* (KACs) generalize the Even-Mansour construction [15] over multiple rounds. They abstract the structure of AES, and this fact has made them the object of several recent analyses [1, 7–9, 13, 29]. Given t permutations $\pi = (\pi_1, \ldots, \pi_t)$ on n-bit strings, as well as n-bit subkeys $L_0, L_1, \ldots, L_t$, the t-round KAC construction $\mathrm{KAC}[\pi, t]$ outputs, on input M, the value

$$L_t \oplus \pi_t(L_{t-1} \oplus \pi_{t-1}(\cdots \pi_1(M \oplus L_0) \cdots)) \; . \tag{1}$$

Here, we are specifically interested in (strong) prp security of $\mathrm{KAC}[\pi, t]$, i.e., its indistinguishability from a random permutation (under random secret sub-keys) for adversaries that can query both the construction and its inverse. Analyses here are in the random-permutation model: The permutations $\pi_1, \ldots, \pi_t$ are independent and random, and the distinguisher is given a budget of q on-line construction queries, and $p_1, \ldots, p_t$ queries to each of the permutations. The currently best bound is by Chen and Steinberger (CS) [9], who prove that the distinguishing advantage of any such distinguisher A satisfies (using $N = 2^n$ and $p_1 = \cdots = p_t = p$)

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \leq (t+2)\left(\frac{q(6p)^t}{N^t} \cdot t^2(t+1)^{t+1}\right)^{1/(t+2)} \; . \tag{2}$$

Note that the best known distinguishing attack achieves advantage roughly qp^t/N^t. The bound from (2) is asymptotically "tight", i.e., the attacker needs to spend about $\Omega\left(N^{t/(t+1)}\right)$ queries for the bound to become constant, as in the attack. However, there is a substantial gap between the curve given by the bound and the advantage achieved by the best attack, and the constant hidden inside the Ω notation (which depends on t) is fairly significant.

Exact bounds for KACs. Our first contribution is a (near-)exact bound for KACs which matches the best-known attack (up to a small factor-four loss in the number of primitive queries necessary to achieve the same advantage). Concretely, we show that for A as above,

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \leq \frac{q(4p)^t}{N^t} \; . \tag{3}$$

The core of our proof inherits some of the combinatorial tools from CS's proof. However, we use them in a different (and simpler) way to give a much sharper bound. We elaborate further at the end of this introduction. Clearly, our new bound substantially improves upon the CS bound from (2). For example, for realistic AES-like parameters ($n = 128$ and $t = 10$), and $q = p = 2^{110}$, the CS bound is already vacuous (indeed, the advantage starts becoming substantial at around 2^{100}), and in contrast, our new bound still gives us 2^{-50}. Another feature is that our bound does not make any assumptions on q and p — we can for example set $q = N$ and still infer security as long as p is sufficiently small. In contrast, the CS bound (and the technique behind it) assumes that $p, q \leq N/3$.

We note in passing that Lampe, Patarin, and Seurin [22] already proved a similar bound for the (simpler) case of a specific non-adaptive distinguisher. If one wants however to extend their bound to the adaptive case, a factor-two loss in the number of rounds becomes necessary.

Multi-user security. Similar to all prior works, the above results only consider a single user. Yet, block ciphers are typically deployed *en masse* and attackers are often satisfied with compromising *some* user among many. This can be substantially easier. For example, given multiple ciphertexts encrypted with a single k-bit key, a brute-force key-search attack takes effort roughly 2^k to succeed. However, if the ciphertexts are encrypted with u different keys, the effort is reduced to $2^k/u$. Overall we effectively lose $\log(u)$ bits of security, which can be substantial. Note that this loss is only inherent if exhaustive key-search *is* the best attack — it may be that a given design is subject to better degradation, and assessing what is true is crucial to fix concrete parameters.

The notion of multi-user (mu) security was introduced and formalized by Bellare, Boldyreva, and Micali [2] in the context of public-key encryption. Unfortunately, until recently, research on *provable* mu security for block-cipher designs has been somewhat lacking, despite significant evidence of this being the right metric (cf. e.g. [6] for an overview). Recent notable exceptions are the works of Mouha and Luykx [25] and Tessaro [30]. The former, in particular, provided a tight analysis of the Even-Mansour cipher in the mu setting, and is a special case of our general analysis for $t = 1$.

Multi-user security for KACs. First recall that in the mu setting, the adversary makes q queries to multiple instances of $\mathrm{KAC}[\pi, t]$ (and their inverses), each with an independent key (but all accessing the same π), and needs to distinguish these from the case where they are replaced by independent random permutations. The crucial point is that *we do not know* a per-instance upper bound on the number of the distinguisher queries, which are distributed adaptively across these instances. Thus, in the *worst-case*, at most q queries are made on some instance and by a naive hybrid argument,[1]

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\mathrm{KAC}[\pi,t]}(A) \leq \frac{u \cdot q(4(p+qt))^t}{N^t} \leq \frac{q^2(4(p+qt))^t}{N^t} \quad , \tag{4}$$

where u is an upper bound on the number of different instances (or "users") for which A makes a query, which again can be at most q. Note that such additional multiplicative factor q is significant: e.g., for $t = 1$, it would enforce $q < N^{1/3}$. As our second contribution, we show that this loss is not necessary, and that in fact essentially the same bound as in the single-user case holds, i.e.,

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\mathrm{KAC}[\pi,t]}(A) \leq 2\frac{q(4(p+qt))^t}{N^t} \quad . \tag{5}$$

[1] The increase from p to $p + qt$ is due to the fact that in the reduction to su prp security, the adversary needs to simulate queries to all but one of the instances with direct permutation queries.

To get a sense of why the statement holds true, note that we could prove this bound easily *if we knew* that the adversary makes at most q_i queries for the i-th user, and $q = \sum_i q_i$. In this case, the naive hybrid argument would yield the bound from (5), but we do not have such q_i's. Our proof relies on a "transcript-centric" hybrid argument, i.e., we use a hybrid argument to relate the real-world and ideal-world probabilities that the oracles of the security game behave according to a certain *a-priori fixed transcript*, for which the quantities q_i *are* defined. The fact that looking at these probabilities suffice will be at the core of our approach, discussed below.

Key-length extension and multi-user security. A fundamental problem in symmetric cryptography, first considered in the design of "Triple-DES" (3DES), is that of building a cipher with a "long" key from one with a "short" key to mitigate the effects of exhaustive key search. Analyses of such schemes (in the ideal-cipher model) have received substantial attention [4,11,16–19,23], yet the practical relevance of these works is often put in question given existing designs have already sufficient security margins. However, *the question gains substantial relevance in the multi-user setting* – indeed, the mu PRP security of an ideal cipher with key length k is at most $2^{k/2}$, i.e., 64 bits for AES-128.

In this paper, we analyze XOR-cascades [16,23], which have been shown to give the best possible trade-off between number of rounds and achievable security. Given a block cipher E with k-bit keys and n-bit blocks, the t-round XOR cascade $\mathrm{XC}[E,t]$ uses sub-keys $J_1, \ldots, J_t, L_0, \ldots, L_t$, and on input M, outputs

$$L_t \oplus E_{J_t}(L_{t-1} \oplus E_{J_{t-1}}(\cdots E_{J_1}(M \oplus L_0)\cdots)) \ . \tag{6}$$

A connection between analyzing XC in the ideal-cipher model and KAC in the random permutation model was already noticed [16,17], but the resulting reduction is far from tight. Here, we give a tight reduction, and use our result on $\mathrm{KAC}[\pi,t]$ to show that for every adversary making q construction queries and at most p queries to an ideal cipher, *if the keys* $J_1, \ldots, J_t$ *are distinct*,

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t]}(A) \leq q\left(\frac{4p}{2^{k+n}}\right)^t \ . \tag{7}$$

Our bound does not make any assumption on q (which can be as high as 2^n) and p. If the keys are independent (and may collide), an additional term needs to be added to the bound — a naive analysis gives $t^2/2^k$, which is usually good enough, and this is what done in prior works. This becomes interesting when moving to the multi-user case. For the distinct-key case, we can apply our techniques to inherit the bound from (7) (replacing p with $p + q \cdot t$), noting that we are allowing keys to collide across multiple users, just same-user keys need to be distinct. An important feature of this bound (which is only possible thanks to the fact that we are not imposing any restrictions on query numbers in our original bound for $\mathrm{KAC}[\pi,t]$) is that it also gives guarantees when $q \gg 2^n$ and queries are necessarily spread across multiple users. This is particularly interesting when n is small (e.g., $n = 64$ for DES, or even smaller if E is a format-preserving encryption (FPE) scheme).

However, for the independent-key case, the naive analysis here gives us a term $ut^2/2^k$, where u is the number of users (and $u = q$ may hold). This term is unacceptably large – in particular, if $u = q \gg 2^n$. To this end, we significantly improve (in the single-user case already) the additive term $t^2/2^k$. In the multi-user setting, the resulting bound is going to be extremely close to the one for distinct keys, if $t \neq 3$.[2] We leave the question open of reducing the gap (or proving its necessity) for $t = 3$.

Our techniques. A substantial contribution of our work is conceptual. Section 3.1 below presents our tools in a general fashion, making them amenable to future re-use. We give an overview here.

All of our results rely on establishing a condition we call *point-wise proximity*: That is, we show that there exists an $\epsilon = \epsilon(q)$ such that for all possible transcripts τ describing a possible ideal- or real-world interaction (say with q queries), the probabilities $\mathsf{p}_0(\tau)$ and $\mathsf{p}_1(\tau)$ that the ideal and real systems, respectively, answer consistently with τ (when asked the queries in τ) satisfy

$$\mathsf{p}_0(\tau) - \mathsf{p}_1(\tau) \leq \epsilon \cdot \mathsf{p}_0(\tau) \; .$$

This directly implies that the distinguishing advantage of any q-query distinguisher is at most ϵ. This method was first used by Bernstein [5], and can be seen as a special case of Patarin's H-coefficient method [28] (recently revisited and re-popularized by Chen and Steinberger [9]) and Nandi's "interpolation method" [26], where we do not need to consider the possibility of some transcripts "being bad". It turns out that when we do not need such bad set, the notion becomes robust enough to easily allow for a number of arguments.

Transcript-centric reductions. Our first observation is that point-wise proximity makes a number of classical proof techniques *transcript-centric*, such as hybrid arguments and reductions. For example, assume that for a pair of systems with transcript probabilities p_0 and p_1, we have already established that $\mathsf{p}_0(\tau) - \mathsf{p}_1(\tau) \leq \epsilon \cdot \mathsf{p}_0(\tau)$. Now, to establish that for some other p'_0 and p'_1 we also have $\mathsf{p}'_0(\tau) - \mathsf{p}'_1(\tau) \leq \epsilon \cdot \mathsf{p}'_0(\tau)$, it is enough to exhibit a function φ, mapping transcripts into transcripts, such that

$$\frac{\mathsf{p}'_1(\tau)}{\mathsf{p}'_0(\tau)} = \frac{\mathsf{p}_1(\varphi(\tau))}{\mathsf{p}_0(\varphi(\tau))}$$

for every τ such that $\mathsf{p}'_0(\tau) > 0$. This is effectively a reduction, but the key point is that the reduction φ maps *executions* into *executions* (i.e., transcripts), and thus can exploit some global after-the-fact properties of this execution, such as the number of queries of a certain particular type. This technique will be central e.g. to transition (fairly generically) from single-user to multi-user security in a tight way. Indeed, while a hybrid argument does not give a tight reduction from single-user to multi-user security, such a reduction can be given when we have established the stronger property of single-user point-wise proximity.

The expectation method. Our main quantitative improvement over the CS bound is due to a generalization of the H-coefficient method that we call the *expectation method.*

To better understand what we do, we first note that through a fairly involved combinatorial analysis, the proof of the CS bound [9] gives (implicitly) an exact formula for the ratio $\epsilon(\tau) = 1 - \frac{\mathsf{p}_1(\tau)}{\mathsf{p}_0(\tau)}$ for every "good transcript" τ. The issue here is that $\epsilon(\tau)$ depends on the transcript τ, in particular, on numbers of paths of different types in a transcript-dependent graph $G = G(\tau)$. To obtain a sharp bound, CS enlarge the set of bad transcripts to include those where these path numbers excessively deviate from their expectations, and prove a unique bound $\epsilon^* \geq \epsilon(\tau)$ for all good transcripts. As these quantities do not admit overly strong concentration bounds, only

[2] We note that in practice, it is easy for a user to enforce that her t keys are distinct, making this part of the key sampling algorithm. Still, our bound shows that this is not really necessary for $t \neq 3$.

Markov's inequality applies, and this results in excessive slackness. In particular, an additional parameter appears in the bound, allowing for a trade-off between the probability δ^* of τ being bad and the quality of the upper bound ϵ^*, and this parameter needs to be optimized to give the sharpest bound, which however still falls short of being exact.

The problem here is that the H-coefficient technique takes a worst-case approach, by unnecessarily requiring one *single* ϵ^* to give us an upper bound for *all* (good) transcripts. What we use here is that given a *transcript-dependent* $\epsilon = \epsilon(\tau)$ for which the above upper bound on the ratio holds, then one can simply replace ϵ^* in the final bound with the *expected value* of $\epsilon(\tau)$ for an *ideal-world* transcript τ. This expected value is typically fairly straightforward to compute, since the ideal-world distribution is very simple.

We in fact do even more than this, noticing that for KACs point-wise proximity can be established, and this will allow us to obtain many of the applications of this paper. In fact, once we do not need to enlarge the set of bad transcripts any more as in CS, we observe that every transcript is potentially good. Only in combination with the key (which is exposed as part of the transcript in CS) transcripts can be good or bad. We will actually apply the expectation method on every *fixed* transcript τ, the argument now being only over the choice of the random sub-keys $L_0, L_1, \ldots, L_t$ – this makes it even simpler.

A PERSPECTIVE. The above techniques are all fairly simple in retrospect, but they all indicate a conceptual departure from the standard "good versus bad" paradigm employed in information-theoretic indistinguishability proofs. CS already suggested that one can generalize their methods beyond a two-set partition, but in a way, what we are doing here is an extreme case of this, where every set in the partition is a singleton set.

It also seems that the issue of using Markov's inequality has seriously affected the issue of proving "exact bounds" (as opposed to asymptotically tight ones). Another example (which we also revisit) is the reduction of security of XOR cascades to that of KACs [16,17].

2 Preliminaries

NOTATION. For a finite set S, we let $x \leftarrow\$ S$ denote the uniform sampling from S and assigning the value to x. Let $|x|$ denote the length of the string x, and for $1 \leq i < j \leq |x|$, let $x[i,j]$ denote the substring from the ith bit to the jth bit (inclusive) of x. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with randomness r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow\$ A(x_1, \ldots)$ be the resulting of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$.

MULTI-USER PRP SECURITY OF BLOCKCIPHERS. Let $\Pi : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ be a blockcipher, which is built on a family of independent, random permutations π : Index × Dom → Dom. (Note that here Index could be a secret key, in this case π will model an ideal cipher, or just a small set of indices, in which case π models a (small) family of random permutations.) We associate with Π a key-sampling algorithm Sample. Let A be an adversary. Define

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\Pi[\pi],\text{Sample}}(A) = \Pr[\text{Real}^A_{\Pi[\pi],\text{Sample}} \Rightarrow 1] - \Pr[\text{Rand}^A_{\Pi[\pi],\text{Sample}} \Rightarrow 1]$$

where games Real and Rand are defined in Fig. 1. In these games, we first use Sample to sample keys $K_1, K_2, \ldots \in \mathcal{K}$ for Π, and independent, random permutations $f_1, f_2, \ldots$ on $\mathcal{M}$. The adversary is given four oracles PRIM, PRIMINV, ENC, and DEC. In both games, the oracles PRIM and

```
proc Initialize()    $\boxed{\text{Real}^{A}_{\Pi[\pi],\text{Sample}}}$
for $i = 1, 2, \ldots$ do $K_i \leftarrow\!\!\$\ \text{Sample}()$
proc Enc$(i, x)$ {return $\Pi_{K_i}[\pi](x)$}
proc Dec$(i, y)$ {return $\Pi^{-1}_{K_i}[\pi](y)$}
proc Prim$(J, u)$ {return $\pi_J(u)$}
proc PrimInv$(J, v)$ {return $\pi^{-1}_J(v)$ }
```

```
proc Initialize()    $\boxed{\text{Rand}^{A}_{\Pi[\pi],\text{Sample}}}$
for $i = 1, 2, \ldots$ do $f_i \leftarrow\!\!\$\ \text{Perm}(\mathcal{M})$
proc Enc$(i, x)$ {return $f_i(x)$}
proc Dec$(i, y)$ {return $f_i^{-1}(y)$}
proc Prim$(J, u)$ {return $\pi_J(u)$}
proc PrimInv$(J, v)$ {return $\pi^{-1}_J(v)$}
```

Fig. 1: **Games defining the multi-user security of a blockcipher $\Pi : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$.** This blockcipher is based on a family of independent, random permutations π : Index × Dom → Dom. The game is associated with a key-sampling algorithm Sample. Here $\text{Perm}(\mathcal{M})$ denotes the set of all permutations on $\mathcal{M}$.

PrimInv always give access to the primitive π and its inverse respectively. The Enc and Dec oracles gives access to $f_1(\cdot), f_2(\cdot), \ldots$ and their inverses respectively in game Rand, and access to $\Pi[\pi](K_1, \cdot), \Pi[\pi](K_2, \cdot), \ldots$ and their inverses in game Real. The adversary finally needs to output a bit to tell which game it's interacting.

For the special case that and adversary A only queries $\text{Prim}(\cdot), \text{Enc}(1, \cdot)$, and their inverses, we write $\mathsf{Adv}^{\pm\text{prp}}_{\Pi[\pi],\text{Sample}}(A)$ to denote the advantage of A.

If Sample is the uniform sampling of $\mathcal{K}$ then we only write $\mathsf{Adv}^{\pm\text{prp}}_{\Pi[\pi]}(A)$ and $\mathsf{Adv}^{\pm\text{mu-prp}}_{\Pi[\pi]}(A)$. If Π doesn't use π then $\mathsf{Adv}^{\pm\text{prp}}_{\Pi}(A)$ coincides with the conventional (strong) PRP advantage of A against Π.

Maclaurin's inequality. In some proofs, we'll need to use the following inequality.

Lemma 1 (Maclaurin's inequality). Let $m \ge t \ge 1$ be integers, and let $a_1, \cdots, a_m$ be non-negative real numbers. Then,

$$\frac{1}{\binom{m}{t}} \sum_{1 \le \ell_1 < \cdots < \ell_t \le m} a_{\ell_1} \cdots a_{\ell_t} \le \frac{1}{m^t} \Big(\sum_{i=1}^{m} a_i \Big)^t .$$

3 Indistinguishability Proofs via Point-wise Proximity

This section discusses techniques for information-theoretic indistinguishability proofs. A reader merely interested in our theorems can jump ahead to the next sections — the following tools are not needed to understand the actual statements, only their proofs. Here we start with an abstract framework for indistinguishability proofs in Section 3.1, where we also revise the H-coefficient method within this framework. We then present the notion of point-wise proximity in Section 3.2, together with techniques used to prove it, and conclude in Section 3.3 with an application of point-wise proximity to generically infer tight bounds for multi-user security.

3.1 The indistinguishability framework

Let us consider the setting of a distinguisher A (outputting a decision bit) interacting with one of two "systems" $\mathbf{S}_0$ and $\mathbf{S}_1$. These systems take inputs and produce outputs, and are randomized and possibly stateful. We dispense with a formalization of the concept of a system, as an intuitive understanding will be sufficient. Still, this can be done via games [4], random systems [24], ITMs, or whichever other language permits doing so. In this paper, these systems will provide a construction oracle Enc with a corresponding inversion oracle Dec, and a primitive oracle Prim with a corresponding inversion oracle PrimInv, but our treatment here is general, and thus does not assume this form.

The interaction between $\mathbf{S}_b$ and A (for $b \in \{0,1\}$) defines a *transcript* $\tau = ((u_1, v_1), \dots, (u_q, v_q))$ containing the ordered sequence of query-answer pairs describing this interaction. We denote by X_b the random variable representing this transcript. In the following, we consider the problem of upper bounding the statistical distance

$$\mathsf{SD}(X_0, X_1) = \sum_{\tau} \max\{0, \Pr[X_1 = \tau] - \Pr[X_0 = \tau]\} , \tag{8}$$

of the transcripts, where the sum is over all possible transcripts. It is well known that $\mathsf{SD}(X_0, X_1)$ is an upper bound on the distinguishing advantage of A, i.e., the difference between the probabilities of A outputting one when interacting with $\mathbf{S}_1$ and $\mathbf{S}_0$, respectively.

Describing systems. Following [24], a useful way to formally describe the behavior of a system $\mathbf{S}$ is to associate with it a function $\mathsf{p}_{\mathbf{S}}$ mapping a possible transcript $\tau = ((u_1, v_1), \dots, (u_q, v_q))$ with a probability $\mathsf{p}_{\mathbf{S}}(\tau) \in [0,1]$. This is to be interpreted as the probability that if all queries $u_1, \dots, u_q$ in τ are asked to $\mathbf{S}$ in this order, the answers are $v_1, \dots, v_q$. Note that this is not a probability distribution (i.e., summing $\mathsf{p}_{\mathbf{S}}(\tau)$ over all possible τ's does not give one). Moreover, $\mathsf{p}_{\mathbf{S}}$ is independent of any possible distinguisher — it is a description of the system. (And in fact, this is precisely how [24] defines a system.)

Because our distinguishers are computationally unbounded, it is sufficient to assume them to be *deterministic* without loss of generality. A simple key observation is that for deterministic distinguisher A, given the transcript distribution X of the interaction with $\mathbf{S}$, we always have $\Pr[X = \tau] \in \{0, \mathsf{p}_{\mathbf{S}}(\tau)\}$. This is because, if $\tau = ((u_1, v_1), \dots, (u_q, v_q))$, then either A is such that it asks queries $u_1, \dots, u_q$ when fed answers $v_1, \dots, v_q$ (in which case $\Pr[X = \tau] = \mathsf{p}_{\mathbf{S}}(\tau)$), or it is not, in which case clearly $\Pr[X = \tau] = 0$.

Let $\mathcal{T}$ denote the set of transcripts τ such that $\Pr[X_1 = \tau] > 0$. We call such transcripts *valid*. Also, note that if $\tau \in \mathcal{T}$, then we also have $\Pr[X_0 = \tau] = \mathsf{p}_{\mathbf{S}_0}(\tau)$. Therefore, we can rewrite (8) as

$$\mathsf{SD}(X_0, X_1) = \sum_{\tau \in \mathcal{T}} \max\{0, \mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau)\} . \tag{9}$$

Note that which transcripts are valid depends on A, as well as on the system $\mathbf{S}_1$.

The H-coefficient method. Let us revisit the well-known H-coefficient technique [9,28] within this notational framework. (This is also very similar to alternative equivalent treatments, like the "interpolation method" presented in [5,26].) The key step is to partition valid transcripts $\mathcal{T}$ into two sets, the *good* transcripts Γ_{good} and the *bad* transcripts Γ_{bad}. Then, if we can establish the existence of a value ϵ such that for all $\tau \in \Gamma_{\text{good}}$, we have $1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)} \leq \epsilon$, then we can conclude that

$$\begin{aligned}
\mathsf{SD}(X_0, X_1) &= \sum_{\tau} \max\{0, \mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau)\} \\
&\leq \sum_{\tau \in \Gamma_{\text{good}}} \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \max\left\{0, 1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)}\right\} + \sum_{\tau \in \Gamma_{\text{bad}}} \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot 1 \\
&\leq \epsilon + \Pr[X_1 \in \Gamma_{\text{bad}}] .
\end{aligned}$$

We note that in the typical treatment of this method, many authors don't notationally differentiate explicitly between e.g. $\Pr[X_0 = \tau]$ and $\mathsf{p}_{\mathbf{S}_0}(\tau)$ (and likewise for X_1 and $\mathbf{S}_1$), even though this connection is implicitly made. (For example, for typical cryptographic systems, the order of queries is re-arranged to compute $\Pr[X_0 = \tau]$ without affecting the probability, which is a property of $\mathsf{p}_{\mathbf{S}_0}$, since queries may not appear in that order for the given A.) Treating these separately will however be very helpful in the following.

The expectation method. In the H-coefficient method, ϵ typically depends on some *global* properties of the distinguisher (e.g., the number of queries) and the system (key length, input length, etc). However, this can be generalized: Assume that we can give a non-negative function $g : \mathcal{T} \to [0, \infty)$ such that

$$1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)} \leq g(\tau) \tag{10}$$

for all $\tau \in \Gamma_{\text{good}}$, then we can easily conclude, similar to the above, that

$$\begin{aligned} \mathsf{SD}(X_0, X_1) &\leq \sum_{\tau \in \Gamma_{\text{good}}} \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot g(\tau) + \Pr[X_1 \in \Gamma_{\text{bad}}] \\ &\leq \mathbf{E}[g(X_1)] + \Pr[X_1 \in \Gamma_{\text{bad}}] \ . \end{aligned}$$

Note that we have used the fact that the function g is non-negative for the first term to be upper bounded by the expectation $\mathbf{E}[g(X_1)]$. We refer to this method as the *expectation method*, and we will see below that this idea is very useful.

The H-coefficient technique corresponds to the special case where g is "constant", whereas here the value may depend on further specifics of the transcript at hand. Obviously, good choices of g, Γ_{good}, and Γ_{bad} are specific to the problem at hand. We also note that one can set $g(\tau) = 1$ for bad transcripts, and then dispense with the separate calculation of the probability. (The way we present it above, however, makes it more amenable to the typical application.) Note that Chen and Steinberger [9] explain that in the H-coefficient method one may go beyond the simple partitioning in good and bad transcripts. In a sense, what we are doing here is going to the extreme, partitioning Γ_{good} into singleton sets.

3.2 Point-wise proximity

A core observation is that for some pairs of systems $\mathbf{S}_0$ and $\mathbf{S}_1$ (and this will be the case for those we consider), we are able to establish a stronger "point-wise" proximity property.

Definition 1 (Point-wise proximity). *We say that two systems* $\mathbf{S}_0$ *and* $\mathbf{S}_1$ *satisfy* ϵ-point-wise proximity *if,* for every possible transcript τ *with* q *queries,*

$$\Delta(\tau) = \mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) \leq \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \epsilon(q) \ . \tag{11}$$

Note that ϵ is a function of q, and often we will let it depend on more fine-grained partitions of the query complexity. (Also in some cases, the query complexity will be implicit.) In particular, for a certain q-query distinguisher A, by Equation (9), it is clear that ϵ-point-wise proximity implies that $\mathsf{SD}(X_0, X_1) \leq \epsilon$, which is also a bound on A's advantage. Observe that point-wise proximity is a *property of a pair of systems* $\mathbf{S}_0$ *and* $\mathbf{S}_1$, independent of the adversaries interacting with them. Also, it is *equivalent* to the fact that

$$1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)} \leq \epsilon$$

for all τ such that $\mathsf{p}_{\mathbf{S}_1}(\tau) > 0$.

In other words, establishing ϵ-proximity corresponds to applying the H-coefficient method without bad transcripts. This is exactly the special case considered by Bernstein [5], and a related notion (for the special case of permutations) was also considered by Patarin [27]. Of course, this method is not always applicable, but when it is, it will bring numerous advantages.

An example: The switching lemma. Let us look at the example where $\mathbf{S}_0$ is a random function, and $\mathbf{S}_1$ is a random permutation, both with domain and range some N-element set. The following uses ϵ-proximity to show what we believe to be the simplest proof of the Switching Lemma.

Simply observe that for every transcript $\tau = ((u_1, v_1), \ldots, (u_q, v_q))$ (for which we assume wlog that $u_1, \ldots, u_q$ are distinct), if the $v_1, \ldots, v_q$ are also distinct, we have $\mathsf{p}_{\mathbf{S}_1}(\tau) > 0$, and moreover

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau)}{\mathsf{p}_{\mathbf{S}_1}(\tau)} = \frac{\prod_{i=0}^{q-1}(N-i)}{N^q} = \prod_{i=0}^{q-1}\left(1 - \frac{i}{N}\right) = 1 - p_{\mathsf{coll}}(q, N) \ ,$$

where $p_{\mathsf{coll}}(q, N)$ is the probability of a collision among q independent uniform elements from an N-element set. We thus have $p_{\mathsf{coll}}(q, N)$ proximity, from which the bound follows.

The expectation method. We outline a general method to prove ϵ-point-wise proximity based on the above general expectation method.

As the starting point, we extend the system $\mathbf{S}_0$ to depend on some auxiliary random variable S (e.g., a secret key). In particular, we write $\mathsf{p}_{\mathbf{S}_0}(\tau, s)$ to be the probability that $\mathbf{S}_0$ answers queries according to τ *and* that $S = s$. Further, we define $\mathsf{p}_{\mathbf{S}_1}(\tau, s) = \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \Pr[S = s]$, i.e., we think of $\mathbf{S}_1$ as also additionally sampling an auxiliary variable S with the same marginal distribution as in $\mathbf{S}_0$, except that the behavior of $\mathbf{S}_1$ remains independent of S. Then, for every transcript τ,

$$\Delta(\tau) = \sum_s \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \sum_s \mathsf{p}_{\mathbf{S}_0}(\tau, s) = \sum_s \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \ .$$

Now, we establish the following lemma, that is based on the above expectation method.

Lemma 2 (The expectation method). Fix a transcript τ for which $\mathsf{p}_{\mathbf{S}_1}(\tau) > 0$. Assume that there exists a partition Γ_{good} and Γ_{bad} of the range $\mathcal{U}$ of S, as well as a function $g : \mathcal{U} \to [0, \infty)$ such that $\Pr[S \in \Gamma_{\text{bad}}] \leq \delta$ and for all $s \in \Gamma_{\text{good}}$,

$$1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} \leq g(s) \ .$$

Then,

$$\Delta(\tau) \leq (\delta + \mathbf{E}(g(S))) \cdot \mathsf{p}_{\mathbf{S}_1}(\tau) \ .$$

Proof. Note that $s \in \mathcal{U}$ implies $\Pr[S = s] > 0$, and thus $\mathsf{p}_{\mathbf{S}_1}(\tau, s) > 0$. We can easily compute

$$\begin{aligned}\Delta(\tau) &\leq \sum_{s \in \mathcal{U}} \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \\ &= \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \sum_{s \in \mathcal{U}} \Pr[S = s] \cdot \Big(1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)}\Big) \\ &\leq \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \Big(\sum_{s \in \Gamma_{\mathrm{bad}}} \Pr[S = s] + \sum_{s \in \Gamma_{\mathrm{good}}} \Pr[S = s] \cdot g(s)\Big) \\ &\leq (\delta + \mathbf{E}(g(S))) \cdot \mathsf{p}_{\mathbf{S}_1}(\tau) \ . \ \square\end{aligned}$$

We stress that the partitioning into Γ_{good} and Γ_{bad}, as well as the function g and the random variable S, are all allowed to depend on τ (and in fact will depend on them in applications).

Transcript reduction. Lemma 2 gives us one possible approach to prove ϵ-point-wise proximity. Another technique we will use is to simply reduce this property to ϵ-point-wise proximity for another pair of systems.

Typically, we will assume that we are in the above extended setting, where we have enhanced the systems $\mathbf{S}_0$ and $\mathbf{S}_1$ with some auxiliary random variable S. Here, in contrast to the above, we assume that S is not necessarily independent of the behavior of the system $\mathbf{S}_1$. Further, assume that we are given two other systems $\mathbf{S}'_0$ and $\mathbf{S}'_1$ for which ϵ-point-wise proximity holds. To this end, we are simply going to provide an explicit reduction $\mathscr{R}$ which is going to map every (τ, s) for $\mathbf{S}_0$ and $\mathbf{S}_1$ into a transcript $\mathscr{R}(\tau, s)$ for $\mathbf{S}'_0$ and $\mathbf{S}'_1$ such that

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s))} \ .$$

whenever $\mathsf{p}_{\mathbf{S}_1}(\tau, s) > 0$. This will be sufficient for our purposes, because (with $\mathcal{U}$ being the set of s such that $\mathsf{p}_{\mathbf{S}_1}(\tau, s) > 0$)

$$\begin{aligned}\Delta(\tau) &\leq \sum_{s \in \mathcal{U}} \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \Big(1 - \frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)}\Big) \\ &= \sum_{s \in \mathcal{U}} \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \Big(1 - \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s))}\Big) \leq \epsilon \cdot \mathsf{p}_{\mathbf{S}_1}(\tau) \ .\end{aligned}$$

Note that here $\epsilon = \epsilon(q')$, where q' is the number of queries in $\mathscr{R}(\tau, s)$.

3.3 From single-user to multi-user security

There is no generic way to derive a *tight* bound on the multi-user security of a construction given a bound on its single-user security — the naive approach uses a hybrid argument, but as we have no bounds on the per-user number of queries of the attacker (which may vary adaptively), this leads to a loss in the reduction. Here, we show how given point-wise proximity for the single-user case, a bound for multi-user security can generically be found via a hybrid argument.

We assume now we are in the above multi-user prp security setting presented in Section 2, and we let $\mathsf{p}_{\mathsf{real}}$ and $\mathsf{p}_{\mathsf{rand}}$ describe the oracles available in the real and random experiments (which we can see as systems in the framework above). Assume that we already established ϵ-point-wise proximity for the single-user case for transcripts with at most p primitive queries and q function queries (and we think of $\epsilon = \epsilon(p, q)$ as a function of p and q). That is, we have shown that for *every* transcript τ such that all function queries have form $\textsc{Enc}(i, x)$ and $\textsc{Dec}(i, y)$ for the same i (whereas $\textsc{Prim}(J, u)$ / $\textsc{PrimInv}(J, v)$ are unrestricted),

$$\mathsf{p}_{\mathsf{rand}}(\tau) - \mathsf{p}_{\mathsf{real}}(\tau) \leq \mathsf{p}_{\mathsf{rand}}(\tau) \cdot \epsilon(p, q) \ . \tag{12}$$

Let m be the number of calls to π/π^{-1} that a single call to Π/Π^{-1} makes. Also assume now that ϵ satisfies the following properties: (i) $\epsilon(x, y) + \epsilon(x, z) \leq \epsilon(x, y + z)$, for every $x, y, z \in \mathbb{N}$, and (ii) $\epsilon(\cdot, z)$ and $\epsilon(z, \cdot)$ are non-decreasing functions on $\mathbb{N}$, for every $z \in \mathbb{N}$. Property (ii) usually holds, because asking more queries should only increase the adversary's advantage. Property (i) is also usually satisfied by typical functions we use to bound distinguishing advantages. Then, we show the following.

Lemma 3 (From su to mu point-wise proximity). Assume all conditions above are met. Then for all transcripts τ with at most q function queries (for arbitrary users) and p primitive queries,

$$\mathsf{p}_{\mathsf{rand}}(\tau) - \mathsf{p}_{\mathsf{real}}(\tau) \leq \mathsf{p}_{\mathsf{rand}}(\tau) \cdot 2\epsilon(p + q \cdot m, q) \tag{13}$$

Proof. Fix an arbitrary transcript τ, and assume that in τ, function queries are made for r users $u_1, \ldots, u_r \in \mathbb{N}$. Wlog, assume that $\mathsf{p}_{\mathsf{rand}}(\tau) \geq \mathsf{p}_{\mathsf{real}}(\tau)$ and $\epsilon(p + qm, q) \leq 1/2$; otherwise the claimed result is vacuous. For each $i \in \{0, 1, \ldots, r\}$, consider the hybrid system $\mathbf{S}_i$ which provides a compatible interface with the real and random games, and answers primitives queries in the same way, but queries for user u_j for $j > i$ are answered with the actual construction Π and Π^{-1}, whereas queries for u_j with $j \leq i$ are answered by i independent random permutations. Then clearly $\mathsf{p}_{\mathbf{S}_0}(\tau) = \mathsf{p}_{\mathsf{real}}(\tau)$ and $p_{\mathbf{S}_r}(\tau) = \mathsf{p}_{\mathsf{rand}}(\tau)$. We can thus rewrite

$$\mathsf{p}_{\mathsf{rand}}(\tau) - \mathsf{p}_{\mathsf{real}}(\tau) = \sum_{i=1}^{r} \mathsf{p}_{\mathbf{S}_i}(\tau) - \mathsf{p}_{\mathbf{S}_{i-1}}(\tau) \ .$$

Suppose that τ contains q_i queries to $\textsc{Enc}(u_i, \cdot)/\textsc{Dec}(u_i, \cdot)$. We'll prove that for any $i \in \{1, \ldots, r\}$,

$$\mathsf{p}_{\mathbf{S}_i}(\tau) - \mathsf{p}_{\mathbf{S}_{i-1}}(\tau) \leq \mathsf{p}_{\mathbf{S}_i}(\tau) \cdot \epsilon(p + qm, q_i) \ . \tag{14}$$

This claim will be justified later. Now Equation (14) implies that

$$\mathsf{p}_{\mathbf{S}_{i-1}}(\tau) \geq (1 - \epsilon(p + qm, q_i)) \cdot \mathsf{p}_{\mathbf{S}_i}(\tau)$$

for every $i \in \{1, \ldots, r\}$. Thus for any $i \in \{1, \ldots, r\}$,

$$\begin{aligned}\mathsf{p}_{\mathbf{S}_0}(\tau) &\geq \mathsf{p}_{\mathbf{S}_i}(\tau) \prod_{j=1}^{i} (1 - \epsilon(p + qm, q_j)) \geq \mathsf{p}_{\mathbf{S}_i}(\tau)\Big(1 - \sum_{j=1}^{i} \epsilon(p + qm, q_j)\Big) \\ &\geq \mathsf{p}_{\mathbf{S}_i}(\tau)\Big(1 - \sum_{j=1}^{r} \epsilon(p + qm, q_j)\Big) \geq \mathsf{p}_{\mathbf{S}_i}(\tau)\Big(1 - \epsilon(p + qm, q)\Big) \geq \frac{1}{2}\mathsf{p}_{\mathbf{S}_i}(\tau) \ .\end{aligned}$$

The first inequality is due to the fact that $(1-x)(1-y) \geq 1-(x+y)$ for every $0 \leq x, y \leq 1$; the second last inequality is due to the property (i) of function ϵ; and the last one is due to the assumption that $\epsilon(p+qm, q) \leq 1/2$. Combining this with Equation (14),

$$\begin{aligned}\sum_{i=1}^{r} \mathsf{p}_{\mathbf{S}_i}(\tau) - \mathsf{p}_{\mathbf{S}_{i-1}}(\tau) &\leq \sum_{i=1}^{r} \mathsf{p}_{\mathbf{S}_i}(\tau) \cdot \epsilon(p+qm, q_i) \\ &\leq \sum_{i=1}^{r} 2\mathsf{p}_{\mathbf{S}_0}(\tau) \cdot \epsilon(p+qm, q_i) \leq 2\mathsf{p}_{\mathbf{S}_0}(\tau) \cdot \epsilon(p+qm, q),\end{aligned}$$

which leads to the claimed result, due to the assumption that $\mathsf{p}_{\mathsf{rand}}(\tau) \geq \mathsf{p}_{\mathsf{real}}(\tau) = \mathsf{p}_{\mathbf{S}_0}(\tau)$. What's left is to prove Equation (14). To this end, fix $i \in \{1, \ldots, r\}$, and we are going to use the transcript reduction technique presented above. First off, enhance $\mathbf{S}_{i-1}$ and $\mathbf{S}_i$ with an auxiliary variable S which contains (i) the transcript of all internal PRIM/PRIMINV caused by querying $\textsc{Enc}(u_j, \cdot)/\textsc{Dec}(u_j, \cdot)$, and (ii) the keys K_j of users u_j, for $j > i$. Now, given (τ, s), note that if we start by removing all queries from τ for users u_j for $j < i$ (which are answered by random permutations in both $\mathbf{S}_{i-1}$ and $\mathbf{S}_i$), obtaining a transcript τ', then we necessarily have

$$\frac{\mathsf{p}_{\mathbf{S}_{i-1}}(\tau, s)}{\mathsf{p}_{\mathbf{S}_i}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}_{i-1}}(\tau', s)}{\mathsf{p}_{\mathbf{S}_i}(\tau', s)} \;.$$

This is because the distribution of these answers is independent of what is in τ', s in both $\mathbf{S}_{i-1}$ and $\mathbf{S}_i$, and in both cases the distribution is identical. Then, given τ' and a value s for S (in either of the system), we can easily construct a transcript $\mathscr{R}(\tau', s)$ where all function queries for users u_j for $j > i$ are removed, all primitive queries in s are made directly to the PRIM and PRIMINV oracles in τ', and all keys K_j of users u_j for $j > i$ are removed. It is easy to verify that

$$\frac{\mathsf{p}_{\mathbf{S}_{i-1}}(\tau, s)}{\mathsf{p}_{\mathbf{S}_i}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}_{i-1}}(\mathscr{R}(\tau', s))}{\mathsf{p}_{\mathbf{S}_i}(\mathscr{R}(\tau', s))},$$

because (i) the function queries of users u_j can be derived from the primitive queries and K_j, and (ii) the keys K_j for $j > i$ are independent of what's used for user i. However, note $\mathscr{R}(\tau', s)$ contains q_i ENC/DEC queries, all for users u_i, and at most $p + q \cdot m$ queries to PRIM / PRIMINV. As for those transcripts we have already established ϵ-point-wise proximity, Equation (14) follows by the transcript reduction method. □

4 Exact Bounds for Key-Alternating Ciphers

4.1 Results and Discussion

This section provides a comprehensive single- and multi-user security analysis of key-alternating ciphers. After reviewing the construction, and the concrete bound proved by Chen and Steinberger [9], we state and discuss our main results, starting with the single-user security case.

KEY-ALTERNATING CIPHERS. Let us review the key-alternating cipher construction. Let t and n be positive integers, and let $\pi : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^n$ be a family of permutations on $\{0,1\}^n$. We write $\pi_i(\cdot)$ to denote $\pi(i, \cdot)$, and N for 2^n. The Key-Alternating Cipher (KAC) construction gives

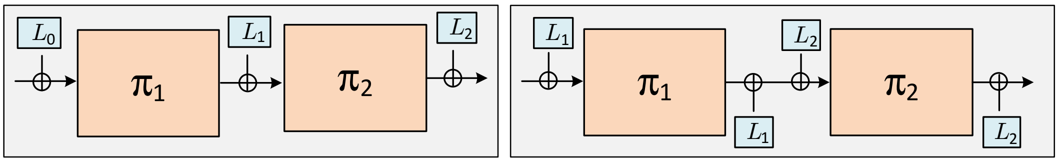

Fig. 2: **Left:** Illustration of KAC$[\pi, 2]$. **Right:** Illustration of KACX$[\pi, 2]$.

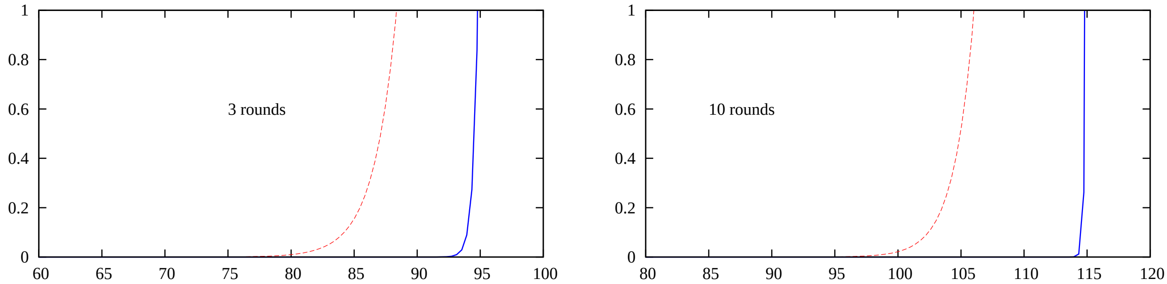

Fig. 3: **Su PRP security of** KAC **on** 3 **rounds (left) and** 10 **rounds (right) on 128-bit strings: our bounds versus CS's.** The solid lines depict our bounds, and the dashed ones depict CS's bounds. In both pictures, $p = q$, and the x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the PRP security of KAC.

a blockcipher $\mathrm{KAC}[\pi, t] : (\{0,1\}^n)^{t+1} \times \{0,1\}^n \to \{0,1\}^n$ as follows. On input x and keys $K = (L_0, \ldots, L_t) \in (\{0,1\}^n)^{t+1}$, $\mathrm{KAC}[\pi, t](K, x)$ returns y_t, where $y_0 = x \oplus L_0$, and $y_i = \pi_i(y_{i-1}) \oplus L_i$ for every $i \in \{1, \ldots, t\}$. It is a direct generalization of the classic Even-Mansour construction [14]. See Fig. 2 for an illustration of $\mathrm{KAC}[\pi, 2]$.

The CS bound. Chen and Steinberger (CS) [9] shows that if an adversary makes at most q queries to Enc/Dec, and at most $p \le N/3$ queries to $\textsc{Prim}(i, \cdot)$ and $\textsc{PrimInv}(i, \cdot)$ for every $i \in \{1, \ldots, t\}$, then

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \le \frac{qp^t}{N^t} \cdot Ct^2(6C)^t + \frac{(t+1)^2}{C} \tag{15}$$

for any $C \ge 1$. Since Equation (15) holds for *any* $C \ge 1$, to determine the best upper bound for $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A)$ according to this inequality, one needs to find the *minimum* of the right-hand side of Equation (15). For each fixed p, q and t, from the inequality of arithmetic and geometric means:

$$\begin{aligned}\frac{qp^t}{N^t} \cdot Ct^2(6C)^t + \frac{(t+1)^2}{C} &= \frac{qp^t}{N^t} \cdot Ct^2(6C)^t + \frac{(t+1)}{C} + \cdots + \frac{(t+1)}{C}\\ &\ge (t+2)\left(\frac{qp^tCt^2(6C)^t}{N^t} \cdot \frac{(t+1)}{C} \cdots \frac{(t+1)}{C}\right)^{1/(t+2)}\\ &= (t+2)\left(\frac{q(6p)^t}{N^t} \cdot t^2(t+1)^{t+1}\right)^{1/(t+2)} .\end{aligned}$$

The equality happens if $C = \left(\frac{N^t(t+1)}{qt^2(6p)^t}\right)^{(t+2)}$. Equation (15) thus can be rewritten as

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \le (t+2)\left(\frac{q(6p)^t}{N^t} \cdot t^2(t+1)^{t+1}\right)^{1/(t+2)} .$$

(This bound is slightly smaller than the claimed result in [9, Corollary 1].) While this bound is asymptotically optimal, meaning that the adversary needs to spend about $N^{t/(t+1)}$ queries for the bound to become vacuous, it's concretely much weaker than the best possible bound, which is roughly qp^t/N^t [16].

Single-user security of KACs. We establish the following theorem, which gives a near-exact bound on the PRP security of the KAC$[\pi, t]$ construction in the ideal-permutation model. Following the theorem, we first give some comments. The proof is found in Section 4.2, where we also give a high-level overview.

Theorem 1 (Su PRP security of KACs). Let t and n be positive integers, and let $\pi : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^n$ be a family of ideal permutations on $\{0,1\}^n$. Let KAC$[\pi, t]$ be as above. For an adversary A that makes at most q queries to Enc/Dec, and at most p_i queries to Prim$(i, \cdot)$ and PrimInv$(i, \cdot)$ for every $i \in \{1, \ldots, t\}$, it holds that

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \leq 4^t q p_1 \cdots p_t / N^t \ . \tag{16}$$

This bound constitutes a significant improvement over the CS bound. For example, consider $n = 128$ and $t = 3$. For $p = 2^{96}$ and $q = 2^{64}$, CS's result yields $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,3]}(A) \leq 0.71$, whereas according to Theorem 1, $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,3]}(A) \leq 2^{-26}$. See Fig. 3 for a graphical comparison of CS's bound and ours for the case $p = q$ and both $t = 3$ and $t = 10$ rounds. Note that the latter case is the one matching AES-128 the closest. In particular, here, we see that the advantage starts to become noticeable roughly at $q = p = 2^{100}$ for the CS bound, whereas this happens only at 2^{113} for our new bound. One of the issues in the CS bound is that the $1/(t + 2)$ exponent smoothes the actual bound considerably, and thus gives a much less sharp transition from small advantage to large as t increases.

Query regimes. Let us point out two important remarks on the bound. First off, it is important that our bound does *not* require any bound on q and $p_1, \ldots, p_t$. Any of these values can equal N, and the construction remains secure as long as $4^t q p_1 \cdots p_t / N^t$ remains small enough. Dealing with such $q = N$ and $p_i = N$ case requires in fact a completely novel approach, which we introduce and explain below in Section 4.2. This will be important when using our bounds in the proof for the analysis of XOR cascades, which we want to hold true *even* if N is small (e.g., in the case of format-preserving encryption (FPE) [3]) and the attacker distributes $q \gg N$ queries across multiple users, possibly exhausting all possible queries for some of these users.

On the other hand, one might worry that an adversary may *adaptively* distribute the number of queries among the permutations $\pi_1, \ldots, \pi_t$, and want a bound in terms of p, the total number of queries to π. Naively, the bound in Theorem 1 is only $q(4p)^t/N^t$. However, we can exploit our point-wise proximity based approach to get a sharper bound: In each transcript τ, the number of queries $p_i[\tau]$ to π_i is completely determined, and thus Equation (17) in the proof of Theorem 1 can be rewritten as

$$\begin{aligned} \mathsf{ps}_1(\tau) - \mathsf{ps}_0(\tau) &\leq \mathsf{ps}_1(\tau) \cdot \frac{4^t q p_1[\tau] \cdots p_t[\tau]}{N^t} \\ &\leq \mathsf{ps}_1(\tau) \cdot \frac{4^t q (p_1[\tau] + \cdots + p_t[\tau])^t}{N^t t^t} \leq \mathsf{ps}_1(\tau) \cdot \frac{q(4p)^t}{N^t t^t} \ . \end{aligned}$$

Then $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{KAC}[\pi,t]}(A) \le q(4p)^t/(Nt)^t$.

Variants. Consider the following natural variant $\mathrm{KACX}[\pi, t]$ of $\mathrm{KAC}[\pi, t]$. It uses only t subkeys $(L_1, \ldots, L_t) \in (\{0,1\}^n)^t$. On input x, it returns returns y_t, where $y_0 = x$, and $y_i = \pi_i(y_{i-1} \oplus L_i) \oplus L_i$ for every $i \in \{1, \ldots, t\}$. See Fig. 2 for an illustration of KACX. Note that KACX is KAC with effective key $(L_1, L_1 \oplus L_2, L_2 \oplus L_3, \ldots, L_{t-1} \oplus L_t, L_t)$, or in other words, we have chosen random keys *under the constraint that their checksum equals* 0^n.

While we do not give the concrete proof, we note that the same security bound and proof will continue to work: in the proof, whenever we need to use the independence of the subkeys, we consider only t subkeys at a time. We note that for $t = 1$ this is exactly the statement that the security of Even-Mansour is not affected when one sets both keys to be equal.

We note that Gilboa, Gueron, and Nandi [20] also establish pointwise proximity for both single-user and multi-user security of $\mathrm{KACX}[\pi, 1]$. They however don't realize that single-user pointwise proximity implies multi-user security, as a direct proof for multi-user security of $\mathrm{KACX}[\pi, 1]$ is easy. Moreover, their proofs still follow the "good versus bad" paradigm of the H-coefficient technique, as it is enough to give tight bounds for $\mathrm{KACX}[\pi, 1]$.

4.2 Proof of Theorem 1

This section is devoted to the proof of Theorem 1. We begin with a high-level overview of the proof structure. Following the notational framework of Section 3.1, let $\mathbf{S}_0$ and $\mathbf{S}_1$ be the systems associated by the real and ideal game in the prp security definition. In particular, transcripts τ for these systems contain two different types of entries:

- Enc/Dec *queries.* Queries to $\textsc{Enc}(1, x)$ returning y and $\textsc{Dec}(1, y)$ returning x are associated with an entry (enc, x, y).
- Prim/PrimInv *queries.* Queries to $\textsc{Prim}(j, x)$, returning y, and those to $\textsc{PrimInv}(j, y)$, returning x, are associated with an entry $(\mathtt{prim}, j, x, y)$

Note that a further distinction between entries corresponding to forward and backward queries is not necessary, as this will not influence the probabilities $\mathsf{p}_{\mathbf{S}_0}(\tau)$ and $\mathsf{p}_{\mathbf{S}_1}(\tau)$ that a certain transcript occurs. Similarly, these probabilities are invariant under permuting the entries of τ. We also assume without loss of generality that no repeated entries exist in τ (this corresponds to the fact that an attacker asks no redundant queries).

Overview. Our goal is to establish the point-wise proximity for $\mathbf{S}_0$ and $\mathbf{S}_1$, i.e., for any transcript τ containing q entries $(\mathsf{enc}, \cdot, \cdot)$, and at most p_i entries of form $(\mathtt{prim}, i, \cdot, \cdot)$ for $i = 1, \ldots, t$, we show

$$\mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) \le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{4^t q p_1 \cdots p_t}{N^t} \quad . \tag{17}$$

In particular, the proof of (17) is made by two parts:

- **Case 1.** $q, p_1, \ldots, p_t \le N/4$. Then, we give a direct proof of (17) using the expectation method from Lemma 2, where the auxiliary variable S will consist of the secret keys $L_0, L_1, \ldots, L_t$ (in $\mathbf{S}_0$). Our proof will resemble in some aspects that of Chen and Steinberger [9], but it will be much simpler due to the fact that the queries are fixed by τ, and we will only argue over the probability of S. We will still resort to the involved and elegant "path-counting" lemma of [9], but it will only be used to define a function g for which computing the expectation of $g(S)$ will be fairly easy.

- **Case 2.** At least one of $q, p_1, \ldots, p_t$ is bigger than $N/4$. We'll use the transcript reduction method, where the other two systems $\mathbf{S}'_0$ and $\mathbf{S}'_1$ on which we assume we have established point-wise proximity provide the real and ideal games for a $(t-1)$-round KAC.

Therefore, our proof for Equation (17) uses induction on the number of rounds of the KAC. If all queries are smaller than $N/4$ then we can give a direct proof, otherwise the transcript reduction lands us back to the induction hypothesis. To this end, note that although KAC is defined for $t \geq 1$ rounds, we can also define $\mathrm{KAC}[\pi, 0](K, x) = x \oplus K$ for every $x \in \{0,1\}^n$, and the bound degenerates to 1. This is our base case in which Equation (17) vacuously holds.

Now suppose that Equation (17) holds for KAC of $0, \ldots, t-1$ rounds. We now prove that it also holds for KAC of t rounds as well. We'll consider the following two cases.[3]

Case 1: $q, p_1, \ldots, p_t \leq N/4$. Fix a transcript τ. We use the expectation method. Let S be the random variable for the key of $\mathrm{KAC}[\pi, t]$ in $\mathbf{S}_0$, and let $\mathcal{K} = (\{0,1\}^n)^{t+1})$ be the key space. Then S is uniformly distributed over $\mathcal{K}$. For each key $s = (L_0, \ldots, L_t) \in \mathcal{K}$, define the graph $G(s)$ as follows:

- Its set of vertices are partitioned into $t+1$ sets $V_0, \ldots, V_t$, each of 2^n elements. For each $j \in \{0, \ldots, t\}$, use the elements of $\{j\} \times \{0,1\}^n$ to uniquely label the elements of V_j.
- For each entry $(\mathtt{prim}, j, x, y)$ in τ, connect the vertices $(j-1, x \oplus L_{j-1})$ and (j, y).

For a path P in $G(s)$, let $|P|$ denote the number of edges in this path. (A vertex is a also a path that has no edge.) We define the following notion of good and bad keys.

Definition 2 (Bad and good keys). *We say that a key $s = (L_0, \ldots, L_t)$ is* bad *if τ contains an entry (enc, x, y) such that in the graph $G(s)$, there's a path P_0 starting from $(0, x)$ and a path P_1 starting from $(t, y \oplus L_t)$ such that $|P_0| + |P_1| \geq t$. If a key is not bad then we'll say that it's* good. *Let Γ_{bad} be the set of bad keys, and let $\Gamma_{\mathrm{good}} = \mathcal{K} \backslash \Gamma_{\mathrm{bad}}$.*

Let $Z_s(i,j)$ be the number of paths from vertices in V_i to vertices in V_j of $G(s)$. For $0 \leq a < b \leq t$, let $\mathcal{B}(a,b)$ be the collection of sets $\sigma = \{(i_0, i_1), (i_1, i_2), \ldots, (i_{\ell-1}, i_\ell)\}$, with $a = i_0 < \cdots < i_\ell = b$. Let the Enc entries of τ be $(\mathsf{enc}, x_1, y_1), \ldots, (\mathsf{enc}, x_q, y_q)$. For $k \in \{1, \ldots, q\}$, let $\alpha_k[s]$ be the length of the longest path starting from $(0, x_k)$, and $t - \beta_k[s]$ be the length of the longest path ending at (t, y_k). For $0 \leq a < b \leq t$, let $R_{a,b,k}[s] = 1$ if $\alpha_k[s] \geq a$ and $\beta_k[s] \leq b$, and let $R_{a,b,k}[s] = 0$ otherwise. Note that if s is good then $\alpha_k[s] < \beta_k[s]$ for every $k \in \{1, \ldots, q\}$.
Recall that in the expectation method, one needs to find a non-negative function $g : \mathcal{K} \to [0, \infty)$ such that $g(s)$ bounds $1 - \mathsf{ps}_{\mathbf{S}_0}(\tau, s)/\mathsf{ps}_{\mathbf{S}_1}(\tau, s)$ for all $s \in \Gamma_{\mathrm{good}}$. The function g is directly given in the following technical lemma. The proof, which is based on the main combinatorial lemma of [9], is in Appendix A.

Lemma 4. For any $s \in \Gamma_{\mathrm{good}}$, it holds that

$$1 - \frac{\mathsf{ps}_{\mathbf{S}_0}(\tau, s)}{\mathsf{ps}_{\mathbf{S}_1}(\tau, s)} \leq \sum_{k=1}^{q} \sum_{0 \leq a < b \leq t} R_{a,b,k}[s] \cdot \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j) \in \sigma} \frac{Z_s(i,j)}{N - p_j - q} .$$

[3] Note that here the unusual thing is that Case 1 is handled via a direct proof.

Before we continue the proof, a few remarks are needed. First, note that Lemma 4 only needs $q+p_i < N$ for every $i \in \{1, \dots, t\}$. Therefore, one in fact can consider Case 1 for $q, p_1, \dots, p_t \le N/\lambda$, for an arbitrary constant $\lambda > 2$, and Case 2 for $\max\{q, p_1, \dots, p_t\} > N/\lambda$. This will lead to the bound around $q(cp/N)^t$, where $c = \max\{\lambda, 2(\lambda-1)/(\lambda-2)\}$. To minimize this, the best choice of λ is $2+\sqrt{2}$, but we use $\lambda = 4$ for simplicity.

We finally have everything in place to apply the expectation method. Note that

$$\begin{aligned}\mathbf{E}[g(S)] &= \mathbf{E}\left(\sum_{k=1}^{q} \sum_{0 \le a < b \le t} R_{a,b,k}[S] \cdot \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j) \in \sigma} \frac{Z_S(i,j)}{N - p_j - q}\right) \\ &\le \sum_{k=1}^{q} \mathbf{E}\left(\sum_{0 \le a < b \le t} R_{a,b,k}[S] \cdot \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j) \in \sigma} \frac{2Z_S(i,j)}{N}\right),\end{aligned}$$

where the last inequality is due to the hypothesis that $p_1, \dots, p_t, q \le N/4$. We will need the following technical lemma below; the proof is in Appendix B.

Lemma 5. For $k \in \{1, \dots, q\}$,

$$\mathbf{E}\Big(\sum_{0 \le a < b \le t} R_{a,b,k}[S] \cdot \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j) \in \sigma} \frac{2Z_S(i,j)}{N}\Big) \le \frac{(4^t - t - 1)p_1 \cdots p_t}{N^t} \quad .$$

Note that expectation in Lemma 5 is over the *uniform* choices of the key vector $S = (S_0, S_1, \dots, S_t)$, and the proof of Lemma 5 can actually compute the *exact* value of this expectation. Hence, from Lemmas 2, 4, and Lemma 5, to get our bound for Case 1, it suffices to prove that

$$\Pr[S \in \Gamma_{\text{bad}}] \le (t+1)qp_1 \cdots p_t / N^t \quad . \tag{18}$$

To justify Equation (18), let $S = (S_0, \dots, S_t)$. If $S \in \Gamma_{\text{bad}}$ then τ must contain entries $(\texttt{enc}, x, y)$, $(\texttt{prim}, 1, u_1, v_1), (\texttt{prim}, 2, u_2, v_2), \dots, (\texttt{prim}, t, u_t, v_t)$ such that one of the following happens:

- $u_1 = x \oplus S_0$, and $u_i = v_{i-1} \oplus S_i$ for every $i \in \{2, \dots, t\}$, or
- $v_t = y \oplus S_t$, and $u_i = v_{i-1} \oplus S_i$ for every $i \in \{2, \dots, t\}$, or
- $u_1 = x \oplus S_0$, $v_t = y \oplus S_t$, and there is some $\ell \in \{2, \dots, t\}$ such that $u_i = v_{i-1} \oplus S_i$ for every $i \in \{2, \dots, t\} \backslash \{\ell\}$.

Since $S_0, \dots, S_t$ are uniformly and independently random in $\{0,1\}^n$, the chance that S is bad is at most $(t+1)qp_1 \dots p_t/N^t$.

Case 2: $N/4 < \max\{q, p_1, \dots, p_t\} \le N$. Fix a transcript τ. We have three sub-cases below, each needs a different way to define S and uses a different transcript reduction.

We now give an intuition for the proof. We want to derive from (τ, s) a transcript $\mathscr{R}(\tau, s)$ for a system $\mathbf{S}_0'$ that implement the real game for a $(t-1)$-round KAC. In most cases (Cases 2.1 and 2.2), this KAC construction is $\mathrm{KAC}[\pi, t-1]$, and S consists of the last subkey L_t and some additional query-answer pairs. In this case $\mathsf{p}_{\mathbf{S}_1}(\tau, s)$ means the probability that $\mathbf{S}_1$ behaves according to the entries in (τ, s), and that $L_t \leftarrow\!\!\$\ \{0,1\}^n$ independent of $\mathbf{S}_1$ agrees with the subkey in s.

The target transcript $\mathscr{R}(\tau, s)$ consists of the PRIM entries to $\pi_1, \dots, \pi_{t-1}$ in (τ, s), and the query-answer pairs to $\mathrm{KAC}[\pi, t-1]$ that one can infer from the entries $(\texttt{enc}, \cdot, \cdot)$, the entries $(\texttt{prim}, t, \cdot, \cdot)$, and the last subkey as specified in (τ, s). The random variable S and the system $\mathbf{S}_1'$ that implements the ideal game for $\mathrm{KAC}[\pi, t-1]$ will be constructed so that for every $b \in \{0,1\}$, the event that $\mathbf{S}_b$ behaves according to (τ, s) consists of two independent events: (i) $\mathbf{S}_b'$ behaves according to $\mathscr{R}(\tau, s)$, and (ii) π_t behaves according to the entries in (τ, s), and L_t agrees with what's specified in s. Since (ii) doesn't use Enc and Dec oracles, the reduction preserves the ratio $\mathsf{p}_{\mathbf{S}_0}(\tau, s)/\mathsf{p}_{\mathbf{S}_1}(\tau, s)$.

Case 2.1: $p_1, \ldots, p_t \le N/4$ but $N/4 < q \le N$. We'll in fact give an even stronger bound

$$\mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) \le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{4^{t-1} p_1 \ldots p_t}{N^{t-1}} \enspace .$$

Let S be the random variable for the last subkey L_t in $\mathbf{S}_0$ and the $(N-q)$ Enc queries/answers that τ lacks. (We stress that here S has only a *single* subkey, so a value s for S will have the form $\langle L_t, (\mathsf{enc}, x_1, y_1), \ldots, (\mathsf{enc}, x_{N-q}, y_{N-q})\rangle$.) It suffices to show that for any s such that $\mathsf{p}_{\mathbf{S}_1}(\tau, s) > 0$,

$$\mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \le \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \frac{4^{t-1} p_1 \ldots p_t}{N^{t-1}} \enspace . \tag{19}$$

Let $\mathbf{S}_0'$ be the system that implements the real game on $\mathrm{KAC}[\pi, t-1]$. Let f be the ideal permutation that $\mathbf{S}_1$ uses for answering Enc/Dec queries. Let f' be the permutation such that $f'(x) = \pi_t^{-1}(f(x))$ for every $x \in \{0,1\}^n$, and thus f' is also an ideal permutation. The permutation f can be viewed as the cascade of f' and π_t (meaning that $f(x) = \pi_t(f'(x))$ for every $x \in \{0,1\}^n$). Let $\mathbf{S}_1'$ be a system that provides the ideal game on $\mathrm{KAC}[\pi, t-1]$ and uses f' to answer Enc/Dec queries.

For any $b \in \{0,1\}$, although there are N Enc entries in (τ, s) for $\mathbf{S}_b$, since there are only p_t query-answer pairs to π_t, one can only "backtrack" p_t Enc query-answer pairs for $\mathbf{S}_b'$. Let $\mathscr{R}(\tau, s)$ be the transcript consisting of these p_t backtracked pairs and the query-answer pairs to $\pi_1, \ldots, \pi_{t-1}$. Formally, for any entry $(\mathtt{prim}, i, u, v)$ in (τ, s), add this to $\mathscr{R}(\tau, s)$ if $i \le t-1$. Next, for any entry $(\mathtt{prim}, t, u, v)$ in τ, there is exactly one entry (enc, x, y) in (τ, s) such that $v \oplus L_t = y$, so add (enc, x, u) to $\mathscr{R}(\tau, s)$ as the corresponding backtracked query-answer pair. Then $\mathscr{R}(\tau, s)$ has p_t Enc entries and p_i query-answer pairs for π_i, for every $i \le t-1$. Now, for $\mathbf{S}_b$ to behave according to (τ, s), it means that (i) $\mathbf{S}_b'$ must behave according to $\mathscr{R}(\tau, s)$, (ii) the subkey in S—recall that S contains only the last subkey L_t—must agree with what is specified in s, and (iii) π_t must be completely determined from $\mathbf{S}_b'$, the last subkey L_t, and the N Enc entries of (τ, s). Since π_t is independent of $\mathbf{S}_b'$ and L_t,

$$\mathsf{p}_{\mathbf{S}_b}(\tau, s) = \frac{1}{N \cdot N!} \cdot \mathsf{p}_{\mathbf{S}_b'}(\mathscr{R}(\tau, s)) \enspace .$$

Hence

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}_0'}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}_1'}(\mathscr{R}(\tau, s))} \enspace .$$

But from the induction hypothesis,

$$1 - \frac{\mathsf{p}_{\mathbf{S}_0'}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}_1'}(\mathscr{R}(\tau, s))} \le \frac{4^{t-1} p_1 \ldots p_t}{N^{t-1}} \enspace .$$

Case 2.2: $p_1, \ldots, p_{t-1} \le N/4$ but $p_t > N/4$. We'll in fact give an even stronger bound

$$\mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) \le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{4^{t-1} q p_1 \ldots p_{t-1}}{N^{t-1}} \ .$$

Let S be the random variable for the last subkey L_t in $\mathbf{S}_0$ and the $(N - p_t)$ queries/answers to π_t that τ lacks. From now on, this case is exactly the same as Case 2.1, except that since there are now N queries to π_t but only q Enc queries in (τ, s), we can only backtrack q Enc queries in $\mathbf{S}'_b$.

Case 2.3: There is some index $i \in \{1, \ldots, t-1\}$ such that $N/4 < p_i \le N$. We'll give an even stronger bound

$$\mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) \le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{4^{t-1} q}{N^{t-1}} \prod_{j \in \{1,\ldots,t\} \backslash \{i\}} p_j \ .$$

Let S be the random variable for the subkey L_i in $\mathbf{S}_0$ and the other $(N - p_i)$ query-answer pairs to π_i that τ lacks. Fix s such that $\mathsf{p}_{\mathbf{S}_1}(\tau, s) > 0$. It suffices to prove that

$$\mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \le \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \frac{4^{t-1} q}{N^{t-1}} \prod_{j \in \{1,\ldots,t\} \backslash \{i\}} p_j \ .$$

In this case, we'll need to build another $(t-1)$-round KAC. Intuitively, we "collapse" the ith and $(i+1)$th round of KAC$[\pi, t]$ into a single round. Formally, construct $\pi' : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^n$ from π and the subkey L_i in s as follows. For every $j < i$, we have $\pi'(j, \cdot) = \pi(j, \cdot)$. For every $j > i$, let $\pi'(j, \cdot) = \pi(j+1, \cdot)$. Finally, let $\pi'(i, x) = \pi(i+1, \pi(i, x) \oplus L_i)$ for every $x \in \{0,1\}^n$. Thus π' is also a family of independent, ideal permutations on $\{0,1\}^n$. Let $\mathbf{S}'_0$ be a system that provides the real game on KAC$[\pi', t-1]$. Let f be the ideal permutation that $\mathbf{S}_1$ uses for answering Enc/Dec queries and let $\mathbf{S}'_1$ be a system that provides the ideal game on KAC$[\pi', t-1]$ and uses f to answer Enc/Dec queries.

Now, in (τ, s), we have N query-answer pairs for π_i and p_{i+1} query-answer pairs for π_{i+1}. One thus can "connect" those pairs to obtain p_{i+1} query-answer pairs for π'_i, which is the cascade of π_i and π_{i+1}. Formally, for any entry $(\texttt{prim}, j, a, b)$ in (τ, s), if $j < i$ then add this entry to $\mathscr{R}(\tau, s)$ as a query for π'_j, and if $j > i+1$ then add $(\texttt{prim}, j-1, a, b)$ to $\mathscr{R}(\tau, s)$ as a query for π'_{j-1}. Next, for every entry $(\texttt{prim}, i+1, u, v)$ in τ, there is exactly one entry $(\texttt{prim}, i, x, y)$ in (τ, s) such that $y \oplus L_i = u$, so add $(\texttt{prim}, i, x, v)$ to $\mathscr{R}(\tau, s)$ as the corresponding connecting query. Hence $\mathscr{R}(\tau, s)$ has q Enc queries and p_j queries to π'_j if $j < i$, and p_{j+1} queries to π'_j if $j \ge i$.

For each $b \in \{0,1\}$, for $\mathbf{S}_b$ to behave according to (τ, s), it means that (i) $\mathbf{S}'_b$ must behave according to $\mathscr{R}(\tau, s)$, (ii) the subkey in S must agree with what's specified in s, and (iii) π_t must behave according to the N entries specified by (τ, s). Note that π'_i is the cascade of π_i and π_{i+1}, and since π_{i+1} is independent of π_i, so is π'_i. Hence

$$\mathsf{p}_{\mathbf{S}_b}(\tau, s) = \frac{1}{N \cdot N!} \cdot \mathsf{p}_{\mathbf{S}'_b}(\mathscr{R}(\tau, s)) \ .$$

Hence

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s))} \ .$$

But from the induction hypothesis,

$$1 - \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s))} \le \frac{4^{t-1} q}{N^{t-1}} \prod_{j \in \{1,\ldots,t\} \backslash \{i\}} p_j \ .$$

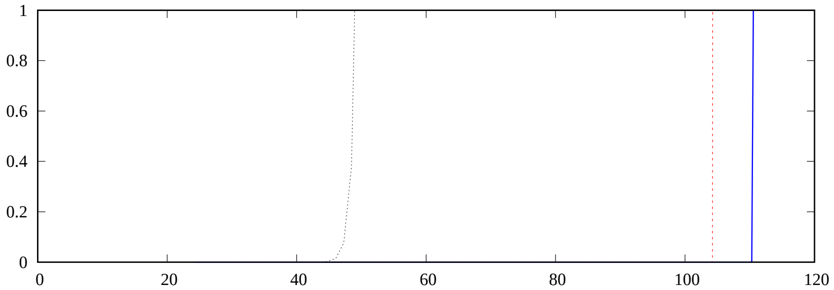

Fig. 4: **Mu PRP security of 10-round KAC on 128-bit strings.** From left to right: the naive bound by using the hybrid argument with CS's result, the naive bound by using the hybrid argument with the su PRP result in Theorem 1, and the bound in Theorem 2. We set $p = q = u$, where u is the number of users. The x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of KAC.

4.3 Multi-user security of KAC

In this section, we consider the multi-user security of KAC. The bounds are immediate, and rely on the fact that the actual *proof* of Theorem 1 established point-wise proximity. Indeed, from Equation (17) in the proof of Theorem 1 and Lemma 3, we obtain Theorem 2. The analogous claims also hold for the variant KACX we discussed above.

Theorem 2 (Mu PRP security of KACs). Let t and n be positive integers, and let $\pi : \mathbb{N} \times \{0,1\}^n \to \{0,1\}^n$ be a family of ideal permutations on $\{0,1\}^n$. Let A be an adversary that makes at most q queries to $\textsc{Enc}/\textsc{Dec}$, and at most p_i queries to $\textsc{Prim}(i,\cdot)/\textsc{PrimInv}(i,\cdot)$ for every $i \in \{1,\ldots,t\}$. Then

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KAC}[\pi,t]}(A) \le \frac{2 \cdot 4^t q(p_1 + qt) \cdots (p_t + qt)}{N^t} \enspace .$$

We note that this bound is essentially the same as the one from Theorem 1, with an additional factor two and the additive term qt. This additive term plays a significant role when t is small, but its role decreases as q grows. Concretely, for $t = 1$, we recover the Even-Mansour multi-user bound of Mouha and Luykx [25], i.e., $\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KAC}[\pi,1]}(A) \le \frac{8(qp+q^2)}{N}$. The $O(q^2/N)$ term takes into account collisions on the keys across multiple users, which allows to easily distinguish and is therefore tight. Note that for $t = 1$, the distinction between single-key or two-key Even-Mansour is exactly the distinction between KAC and KACX, and our bounds are identical.

Beating the hybrid argument. We would like to stress once more the importance of giving direct bounds for mu security, as opposed to using a naive hybrid argument. Indeed, if we used the hybrid argument on our su PRP result in Theorem 1 then we would obtain an inferior bound with form

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KAC}[\pi,t]}(A) \le \frac{u \cdot 4^t q(p_1 + qt) \cdots (p_t + qt)}{N^t}$$

where u is the number of users. If one used the hybrid argument on CS's original bound, then the bound becomes

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KAC}[\pi,t]}(A) \le u(t+2)\left(\frac{q(6p + 6qt)^t}{N^t} \cdot t^2(t+1)^{t+1}\right)^{1/(t+2)} \enspace .$$

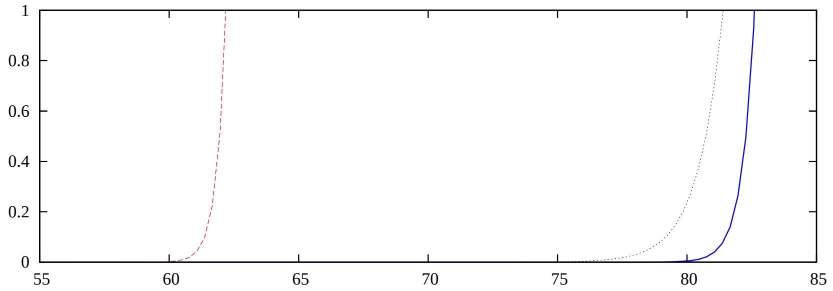

Fig. 5: **Mu PRP security of 2-round KACX on 128-bit strings.** From left to right: the naive bound by using the hybrid argument with our su PRP result for KACX, the bound from CLS's result, and our mu PRP for KACX. We set $p = q = u$, where u is the number of users. The x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of KACX.

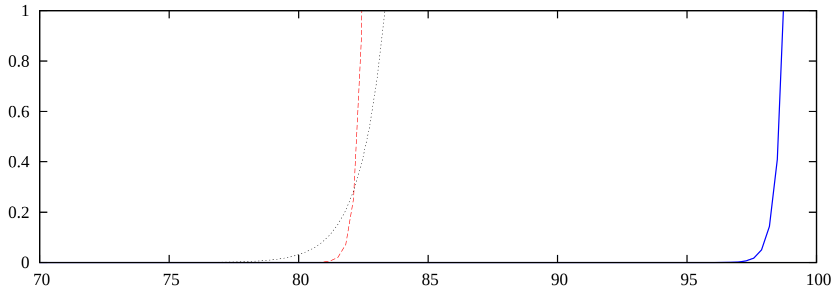

Fig. 6: **Mu PRP security of 4-round KACX on 128-bit strings.** From left to right: the bound from CLS's result (the black dotted line), the naive bound by using the hybrid argument with our su PRP result for KACX (the red dashed line), and our mu PRP for KACX (the blue solid line). We set $p = q = u$, where u is the number of users. The x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of KACX.

This makes one important point apparent: While the exponent $1/(t + 2)$ in CS's bound is already undesirable in the su PRP setting, in the mu PRP case, it's much worse, as illustrated in Fig. 4. If one models AES as a 10-round KAC on 128-bit strings then our mu PRP result suggests that AES has about 110-bit security. Using the hybrid argument with our su PRP result decreases it to 100-bit security, whereas using the hybrid argument on CS's result plummets to 45-bit security.

Mu security for KACX. Again, the bound in Theorem 2 also applies for the variant KACX of KAC. Cogliati, Lampe, and Seurin (CLS) [10] realize the same bound for $t = 1$, and obtain the following bound for $t = 2$:

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KACX}[\pi,2]}(A) \le \frac{30\sqrt{q}(p+q)}{N} \enspace .$$

This bound is much better than the naive one by using the hybrid argument on our su PRP result for KACX, but it's still considerably weaker than our mu PRP result. See Fig. 5 for an illustration. For $t > 2$, if t is even, CLS obtain the following bound:

$$\mathsf{Adv}^{\pm\text{mu-prp}}_{\text{KACX}[\pi,t]}(A) \le \frac{4 \cdot 2^{t/4}\sqrt{q}(p+q)^{t/4}}{N^{t/4}} \enspace .$$

In this case, CLS's result is just comparable with the naive bound. An illustration is shown in Fig. 6.

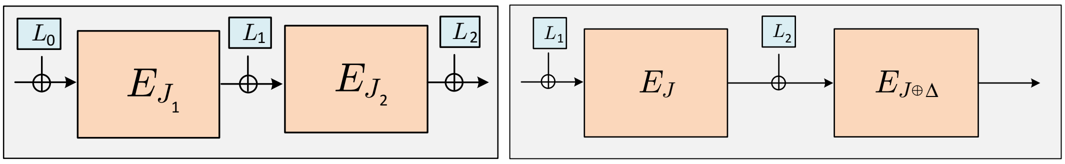

Fig. 7: **Left:** The XC$[E, 2]$ construction. **Right:** The 2XOR$[E]$ construction.

5 XOR Cascades

In this section, we apply the above results to study XOR cascades for blockcipher key-length extension. Variants of XOR cascades have been studied in the literature [16, 17, 19, 21, 23] and the connection with KACs was already observed. However, we improve these results along two different axes: Tightness (we give a much better reduction to the security of KACs than the one of [17], using point-wise proximity), and multi-user security. In particular, to the best of our knowledge, this is the first work studying multi-user key-length extension, a problem we consider to be extremely important, given the considerable security loss in the multi-user regime.

The XOR-Cascade construction. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Let $t \geq 1$ be an integer, and let $\mathcal{K} = (\{0,1\}^k)^t \times (\{0,1\}^n)^{t+1}$. Let Sample be a sampling algorithm that samples $L_0, \ldots, L_t \leftarrow\$ \{0,1\}^n$, and samples without replacement $J_1, \ldots, J_t$ from $\{0,1\}^k$, and outputs $(J_1, \ldots, J_1, L_0, \ldots, L_t)$. The XOR-Cascade construction XC$[E, t]$, on a key $K = (J_1, \ldots, J_t, L_0, \ldots, L_t) \in \mathcal{K}$, describes a permutation on $\{0,1\}^n$ as follows. On input x, XC$[E, t](x)$ returns y_t, where $y_0 = x \oplus L_0$, and $y_i = E_{J_i}(y_{i-1}) \oplus L_i$ for every $i \in \{1, \ldots, t\}$. See Fig. 7 for an illustration of XC$[E, 2]$.

We also define – in analogy with KACX above – a version of XC with t sub-keys $L_1, \ldots, L_t$ (rather than $t+1$), which xor's L_i to the input and the output of E_{J_i} in the i-th round. We refer to this as XCX$[E, t]$, and note that it is simply the t-fold sequential composition of DESX [21].

Single-user security of XC$[E, t]$. The following theorem establishes the single-user security for XC$[E, t]$ in the ideal-cipher model, and, in contrast to previous analyses [16, 17, 23], the resulting bound is essentially exact. We require the keys $J_1, \ldots, J_t$ to be sampled by Sample as random yet distinct. This is no big loss – an additional $t^2/2^k$ term can added to take this into account, but this term is going to be large when moving to the multi-user case. Below, we'll develop a better bound for the independent-key case, and for now, stick with distinct keys.

Theorem 3 (Su PRP security of XC, distinct subkeys). Let t be a positive integer. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let XC$[E, t]$ and Sample be as above. Then in the ideal-cipher model, for any adversary A that makes at most q Enc/Dec queries, and at most p Prim/PrimInv queries,

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t],\mathrm{Sample}}(A) \leq \frac{4^t q p^t}{2^{t(k+n)}} \ . \tag{20}$$

The proof is in Appendix C. Here we point out a few remarks. First off, we note the bound above (and its proof) can easily adapted to analyze XCX$[E, t]$. Moreover, the proof itself is a direct application of point-wise proximity combined with the transcript reduction technique to reduce XC

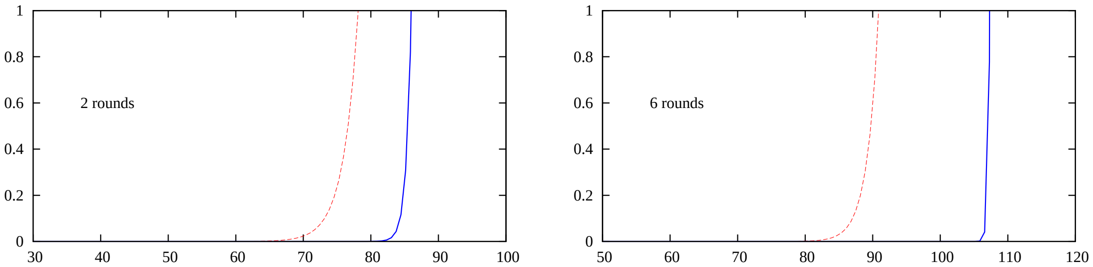

Fig. 8: **Su PRP security (distinct subkeys) of XC on 2 iterations (left) and 6 iterations (right) on $k = 56$ and $n = 64$: our bound versus the results in [16, 17].** The solid lines depict the bound in Theorem 3, and the dashed ones depict the bound obtained by combining the reduction in [16, 17] and our result in Theorem 1. In both pictures, $q = 2^n$, and the x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the su PRP security of XC.

case to the KAC case. This will give a tight relationship, substantially improving on the previous results by Gaži [16] and its generalization by Gaži et al. [17], which actually used an *adversarial* reduction, and needed to resort to Markov-like arguments which, once again, we avoid. Concretely, if we combine the reduction in [16, 17] with our KAC result in Theorem 1, we'll obtain the following weak bound

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t],\mathrm{Sample}}(A) \le 4^t \cdot (2t+2)\left(\frac{qp^t}{2^{t(k+n)}}\right)^{1/(t+1)} .$$

As illustrated in Fig. 8, the gap between the bound above and ours is substantial.

Multi-user security of XC. We now consider the multi-user security of XC. Since the *proof* of Theorem 3 actually establishes pointwise proximity, from Lemma 3, we obtain Theorem 4 below. If we instead use the hybrid argument on the su PRP security then we obtain an inferior bound

$$\mathsf{Adv}^{\pm\mathrm{mu\text{-}prp}}_{\mathrm{XC}[E,t],\mathrm{Sample}}(A) \le u \cdot 4^t q(p+qt)^t/2^{t(k+n)}$$

where u is the number of users. If we use the hybrid argument on the bound obtained by combining the reduction in [16, 17] with our KAC result in Theorem 1, we'll obtain an even weaker bound

$$\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t],\mathrm{Sample}}(A) \le u \cdot 4^t(2t+2)\left(\frac{q(p+qt)^t}{2^{t(k+n)}}\right)^{1/(t+1)} .$$

The three bounds are illustrated in Fig. 9.

Theorem 4 (Mu PRP security of XC, distinct subkeys). Let t be a positive integer. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let XC$[E,t]$ and Sample be as above. Then in the ideal-cipher model, for any adversary A that makes at most q Enc/Dec queries, and at most p Prim/PrimInv queries,

$$\mathsf{Adv}^{\pm\mathrm{mu\text{-}prp}}_{\mathrm{XC}[E,t],\mathrm{Sample}}(A) \le 2 \cdot 4^t q(p+qt)^t/2^{t(k+n)} .$$

We stress here that q is allowed to be larger than $N = 2^n$ — nothing in the theorem limits this, and security is obtained as long $2 \cdot 4^t q(p+qt)^t/2^{t(k+n)}$ is sufficiently small. This is conceptually

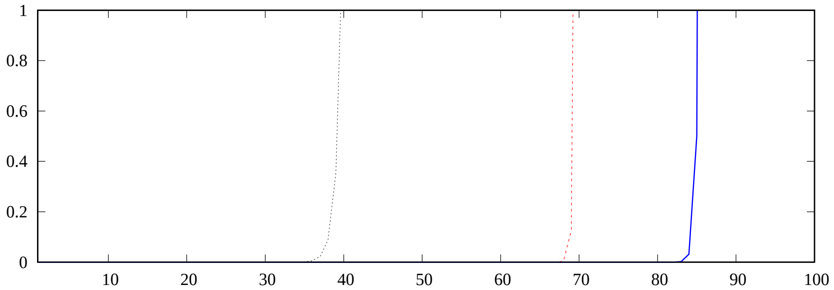

Fig. 9: **Mu PRP security (distinct subkeys) of 3-round** XC **on** $k = 56$ **and** $n = 64$**: our bound versus naive ones from the hybrid argument.** From left to right: the naive bound by using the hybrid argument with the bound obtained by combining the reduction in [16,17] with our KAC result in Theorem 1, the naive bound by using the hybrid argument with the su PRP result in Theorem 3, and the bound in Theorem 4. We set $p = q = u$, where u is the number of users. The x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of XC.

very important. Indeed, we may want to apply our result even to ciphers for which N is very small (these arise in the setting of FPE [3], where one could have $N \approx 2^{30}$, or even less), and a multi-user attacker can exhaust the domain for multiple keys. In passing, we note that the reason such a strong result is possible is inherited directly from the fact that Theorem 1 does not make any restrictions on q.

There are some variants of XC in the literature. For example, Gaži and Tessaro (GT) [19] gave a variant of XC$[E, 2]$ that they call 2XOR. This construction, as illustrated in Fig. 7, uses a shorter key and saves one additional xor, compared to XC$[E, 2]$. While its su PRP security appears to be the same as XC$[E, 2]$, as GT's result suggests, in Appendix E, we show that it has much weaker mu PRP security by giving an attack.

On uniform subkeys. So far we have considered security of the XC construction when each key $K = (J_1, \dots, J_t, L_0, \dots, L_t)$ is chosen so that the subkeys $J_1, \dots, J_t$ are distinct. A natural question is to bound the degradation when $J_1, \dots, J_t \leftarrow_{\$} \{0,1\}^k$. First consider the su setting. A simple solution is to add a term $t^2/2^k$ to account for the probability that there are some $i \neq j$ such that $J_i = J_j$. This is fine for the su setting, but when one moves to the mu setting, this term blows up to $ut^2/2^k$, where u is the number of users. This happens even in the ideal case where the adversary distributes the queries evenly among users. To avoid this undesirable term, in Proposition 1 below, we take a different approach. Intuitively, even if there are only $\ell \leq t$ distinct subkeys, then at least our construction should achieve security level $\epsilon(\ell)$ similar to the bound in Theorem 3 for XC$[E, \ell]$. Let L be the random variable for the number of distinct subkeys in XC$[E, t]$, for example, $\Pr[L = t] \geq 1 - t^2/2^k$. Then our bound would be the expectation $\mathbf{E}(\epsilon(L))$. The gap between this bound and the naive one with the term $t^2/2^k$ may not be large on practical values of n and k, but it allows us to use Lemma 3 to obtain a good mu PRP bound.

Proposition 1 (Su PRP security of XC**, uniform subkeys).** Let $t \geq 2$ be an integer. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let XC$[E, t]$ be as above. Then in the ideal-cipher model, for any adversary A that makes at most q Enc/Dec queries, and at most p Prim/PrimInv queries,

(a) If $t \geq 3$ then $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t]}(A) \leq \frac{4^t q p^t}{2^{(n+k)t}} + \frac{qt^2}{2^k}\left(\frac{t}{2^k} + \frac{4p}{2^{k+n}}\right)^{t-2}$.

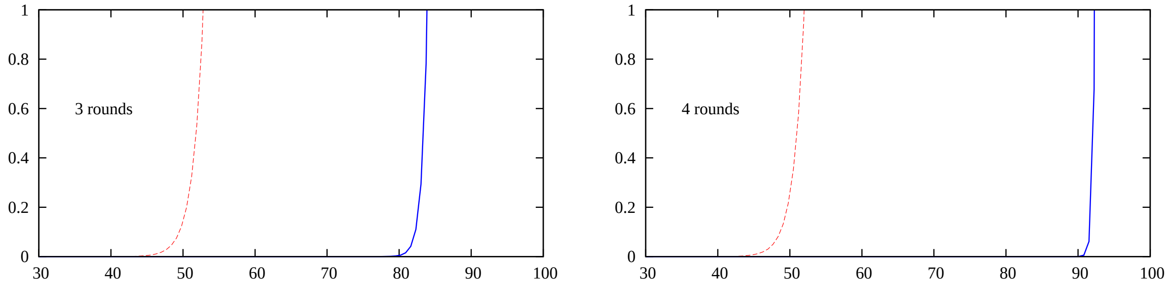

Fig. 10: **Mu PRP security of** XC **(uniform subkeys) on** 3 **iterations (left) and** 4 **iterations (right) on** $k = 56$ **and** $n = 64$**: our bound versus naive one.** The dashed lines depict the bound obtained by adding a term $ut^2/2^k$ to the bound in Theorem 4, and the solid ones depict the bound in Theorem 5, where u is the number of users. In both pictures, $p = q = u$, and the x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of XC.

(b) If $t = 2$ then $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{XC}[E,t]}(A) \le \frac{q(4p)^2}{2^{2(n+k)}} + \frac{4qp}{2^{2k+n}} + \frac{2q}{2^{k+n/2}}$.

The proof of Proposition 1 is in Appendix D, and it also establishes pointwise proximity. From Lemma 3, we obtain Theorem 5 below. As illustrated in Fig. 10, this bound is much better than the naive one obtained via adding a term $ut^2/2^k$ to the bound in Theorem 4 (to account for the probability that there is a user whose subkeys are not distinct), where u is the number of users. When one increases the number of rounds then our bound shows that the security substantially improves (from 80-bit to 90-bit security), but the naive bound still stays at 50-bit security, since the bound $ut^2/2^k$ is the bottleneck, and it gets *worse* when t increases.

Theorem 5 (Mu PRP security of XC**, uniform subkeys).** Let $t \ge 2$ be an integer. Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let $\mathrm{XC}[E,t]$ be as above. Then in the ideal-cipher model, for any adversary A that makes at most q Enc/Dec queries, and at most p Prim/PrimInv queries,

(a) If $t \ge 3$ then $\mathsf{Adv}^{\pm\mathrm{mu\text{-}prp}}_{\mathrm{XC}[E,t]}(A) \le \frac{2 \cdot 4^t q(p+qt)^t}{2^{(n+k)t}} + \frac{2qt^2}{2^k}\left(\frac{t}{2^k} + \frac{4p+4qt}{2^{k+n}}\right)^{t-2}$.

(b) If $t = 2$ then $\mathsf{Adv}^{\pm\mathrm{mu\text{-}prp}}_{\mathrm{XC}[E,t]}(A) \le \frac{2q(4p+8q)^2}{2^{2(n+k)}} + \frac{8q(p+2q)}{2^{2k+n}} + \frac{4q}{2^{k+n/2}}$.

Interpreting the bounds in Theorem 5. For the case $t = 3$, there's a considerable gap compared to the matching attack. See Fig. 11 for an illustration of the degradation of the bound in Theorem 5 compared to that in Theorem 4. This gap is probably an artifact of the proof technique rather than reflecting a true security loss when using uniform subkeys: for example, in the su case, if $J_1 = \cdots = J_t$ then we give up, but of course even in this extreme case, the construction should still retain some reasonable security. For $t \ge 4$ and all practical choices of n and k, the bounds in Theorem 5 and Theorem 4 are close: the former is just about $t^2 + 1$ times worse than the latter. To justify this, note that we can assume that $4(p + qt)/2^n > 2^{k/2}$, otherwise both bounds are tiny. Then

$$\frac{qt^2}{2^k}\left(\frac{t}{2^k} + \frac{4p+4qt}{2^{k+n}}\right)^{t-2} \approx \frac{qt^2}{2^k}\left(\frac{4p+4qt}{2^{k+n}}\right)^{t-2} < t^2 \cdot \frac{4^t q(p+qt)^t}{2^{(n+k)t}} .$$

Pictorially, as shown in Fig. 11, the two bounds are too close, and we have to choose very small n and k so that the gap between the two lines is still visible to the naked eye. Likewise, for $t = 2$

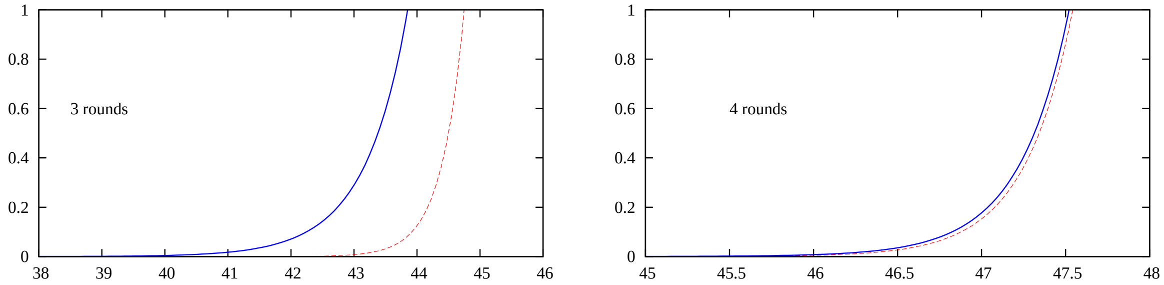

Fig. 11: **Mu PRP security of** XC **on** 3 **iterations (left) and** 4 **iterations (right) on** $k = n = 32$**: uniform versus distinct subkeys.** The dashed lines depict the bound in Theorem 4, and the solid ones depict the bound in Theorem 5. In both pictures, $p = q$, and the x-axis gives the log (base 2) of p, and the y-axis gives upper bounds on the mu PRP security of XC. The parameters n and k are chosen to be small so that in the right picture, the gap between the two lines is still visible to the naked eye.

and and all practical choices of n and k, the bound in Theorem 5 is about twice worse than that of Theorem 4. (In Proposition 1, for $t = 2$, if $J_1 = J_2$ then we don't give up, but show that the construction still retains security bound up to $\frac{4qp}{2^{k+n}} + \frac{2q}{2^{n/2}}$. However, this method fails to work for $t = 3$. It's why the bound in Theorem 5 is still sharp for $t = 2$, but deteriorates for $t = 3$.)

Acknowledgments. We thank Mihir Bellare for insightful feedback, and Daniel J. Bernstein and Samuel Neves for providing relevant pointers. We also wish to thank Atul Luykx and Bart Mennink for pointing out a glitch in a previous version of this write up. Finally, we thank the CRYPTO 2016 reviewers for many insightful comments.

This research was partially supported by NSF grants CNS-1423566 and CNS-1553758 (CAREER).

References

1. E. Andreeva, A. Bogdanov, Y. Dodis, B. Mennink, and J. P. Steinberger. On the indifferentiability of key-alternating ciphers. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 531–550. Springer, Heidelberg, Aug. 2013.
2. M. Bellare, A. Boldyreva, and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In B. Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.
3. M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format-preserving encryption. In M. J. Jacobson Jr., V. Rijmen, and R. Safavi-Naini, editors, *SAC 2009*, volume 5867 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2009.
4. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.
5. D. J. Bernstein. How to stretch random functions: The security of protected counter sums. *Journal of Cryptology*, 12(3):185–192, 1999.
6. D. J. Bernstein. Break a dozen secret keys, get a million more for free. http://blog.cr.yp.to/20151120-batchattacks.html, 2015.
7. 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 D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 45–62. Springer, Heidelberg, Apr. 2012.

8. S. Chen, R. Lampe, J. Lee, Y. Seurin, and J. P. Steinberger. Minimizing the two-round Even-Mansour cipher. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 39–56. Springer, Heidelberg, Aug. 2014.
9. S. Chen and J. P. Steinberger. Tight security bounds for key-alternating ciphers. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350. Springer, Heidelberg, May 2014.
10. B. Cogliati, R. Lampe, and Y. Seurin. Tweaking Even-Mansour ciphers. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 189–208. Springer, Heidelberg, Aug. 2015.
11. Y. Dai, J. Lee, B. Mennink, and J. P. Steinberger. The security of multiple encryption in the ideal cipher model. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 20–38. Springer, Heidelberg, Aug. 2014.
12. Y. Dai and J. Steinberger. Tight security bounds for multiple encryption. Cryptology ePrint Archive, Report 2014/096, 2014. http://eprint.iacr.org/2014/096.
13. O. Dunkelman, N. Keller, and A. Shamir. Minimalism in cryptography: The Even-Mansour scheme revisited. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 336–354. Springer, Heidelberg, Apr. 2012.
14. S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In H. Imai, R. L. Rivest, and T. Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 210–224. Springer, Heidelberg, Nov. 1993.
15. S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, 1997.
16. P. Gaži. Plain versus randomized cascading-based key-length extension for block ciphers. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 551–570. Springer, Heidelberg, Aug. 2013.
17. P. Gazi, J. Lee, Y. Seurin, J. P. Steinberger, and S. Tessaro. Relaxing full-codebook security: A refined analysis of key-length extension schemes. In G. Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 319–341. Springer, Heidelberg, Mar. 2015.
18. P. Gaži and U. M. Maurer. Cascade encryption revisited. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 37–51. Springer, Heidelberg, Dec. 2009.
19. P. Gaži and S. Tessaro. Efficient and optimally secure key-length extension for block ciphers via randomized cascading. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 63–80. Springer, Heidelberg, Apr. 2012.
20. S. Gilboa, S. Gueron, and M. Nandi. Balanced permutations Even-Mansour ciphers. *Cryptography*, 1(1):2, 2016.
21. J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 252–267. Springer, Heidelberg, Aug. 1996.
22. R. Lampe, J. Patarin, and Y. Seurin. An asymptotically tight security analysis of the iterated Even-Mansour cipher. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 278–295. Springer, Heidelberg, Dec. 2012.
23. J. Lee. Towards key-length extension with optimal security: Cascade encryption and xor-cascade encryption. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 405–425. Springer, Heidelberg, May 2013.
24. U. M. Maurer. Indistinguishability of random systems. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 110–132. Springer, Heidelberg, Apr. / May 2002.
25. N. Mouha and A. Luykx. Multi-key security: The Even-Mansour construction revisited. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 209–223. Springer, Heidelberg, Aug. 2015.
26. M. Nandi. A simple and unified method of proving indistinguishability. In R. Barua and T. Lange, editors, *INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 317–334. Springer, Heidelberg, Dec. 2006.
27. J. Patarin. About Feistel schemes with six (or more) rounds. In S. Vaudenay, editor, *FSE'98*, volume 1372 of *LNCS*, pages 103–121. Springer, Heidelberg, Mar. 1998.
28. J. Patarin. The "coefficients H" technique (invited talk). In R. M. Avanzi, L. Keliher, and F. Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345. Springer, Heidelberg, Aug. 2009.
29. 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.
30. S. Tessaro. Optimally secure block ciphers from ideal primitives. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 437–462. Springer, Heidelberg, Nov. / Dec. 2015.

A Proof of Lemma 4

Let $s = (L_0, \ldots, L_t)$. Recall that the Enc entries of τ are $(\mathsf{enc}, x_1, y_1), \ldots, (\mathsf{enc}, x_q, y_q)$. Let G_0 be $G(s)$. For each $k \in \{1, \ldots, q\}$, let G_k be the graph obtained by G_{k-1} as follows. Let $z_0 \leftarrow x_k \oplus L_0$, and for each $i \in \{1, \ldots, t\}$, let $z_i \leftarrow \pi_i(z_{i-1}) \oplus L_i$ and connect vertices $(i-1, z_{i-1})$ and $(i, z_i \oplus L_i)$, if this edge is not yet in the graph G_{k-1}.

Fix some $k \in \{1, \ldots, q\}$. Let G be a graph in the support of the random variable G_{k-1}. We say that G is *well-formed* if there is a path in G connecting $(0, x_j)$ and $(t, y_j \oplus L_t)$ for every $j \in \{1, \ldots, k-1\}$. If G is well-formed then let $\Pr_G[x_k \to y_k]$ be the probability that in $\mathbf{S}_0$, if S agrees with s, and the Prim/PrimInv oracles behave according to the constraints specified in G for every $j \in \{1, \ldots, k-1\}$, then querying x_k to Enc results in y_k. Let G^* be the graph obtained from G by deleting the path connecting $(0, x_i)$ and $(t, y_i \oplus L_t)$ for every $i \in \{1, \ldots, k-1\}$. Let $U_G(i,j)$ be the number of paths P from vertices in V_i to vertices in V_j of G^*, such that there is no vertex in V_{i-1} that is connected to the first vertex of P. Let $F_G(j)$ be the number of edges in G^* connecting vertices in V_{j-1} and those in V_j. Then $U_G(i,j) \le Z_s(i,j)$ and $F_G(j) \le p_j$. We'll need the following result of CS.[4]

Lemma 6. [9, Lemma 1] Fix $k \in \{1, \ldots, q\}$ and let $\ell = k - 1$. Let G be a well-formed graph in the support of G_{k-1} and let $U_G(i,j)$, and $E_G(j)$ be as above. Then

$$\Pr_G[x_k \to y_k] = \frac{1}{N-\ell}\Big(1 - \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} (-1)^{|\sigma|} \prod_{(i,j)\in\sigma} \frac{U_G(i,j)}{N - \ell - F_G(j)}\Big) \enspace . \qquad \square$$

From Lemma 6, for a well-formed G in the support of G_{k-1},

$$\begin{aligned} &\Pr_G[x_k \to y_k] \\ &= \frac{1}{N-k+1}\Big(1 - \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} (-1)^{|\sigma|} \prod_{(i,j)\in\sigma} \frac{U_G(i,j)}{N-k+1-F_G(j)}\Big) \\ &\ge \frac{1}{N-k+1}\Big(1 - \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} \prod_{(i,j)\in\sigma} \frac{Z_s(i,j)}{N-q-p_j}\Big) \enspace . \end{aligned} \tag{21}$$

Let Bad_b be the event that $\mathbf{S}_b$ behaves according to the Prim entries in (τ, s). Then $\Pr[\mathrm{Bad}_0] = \Pr[\mathrm{Bad}_1]$. On the one hand,

$$\mathsf{ps}_1(\tau, s) = \frac{\Pr[\mathrm{Bad}_1]}{N^{t+1} \cdot N(N-1)\cdots(N-q+1)} \enspace .$$

[4] [9, Lemma 1] looks like just the case $k = 1$ of our Lemma 6, but CS used it for a general k and said that

> "Thus the shores of G will have size $N - \ell$, not N. Indeed, we committed a white lie when we stated in Lemma 1 that the shores of G would be copies of $\{0,1\}^n$. Of course, all that mattered was the size of those shores, and we can apply Lemma 1 by replacing N with $N - \ell$ throughout the main bound."

Indeed, when Dai and Steinberger [12] used this result, they rewrote it for a general k. Lemma 6 here follows the statement in [12, Lemma 3].

On the other hand, for $\mathbf{S}_0$ to behave according to (τ, s), it means that (i) S must agree with s, and the system must behave according to the $\textsc{Prim}$ entries in (τ, s), and (ii) for every $k = 1, \ldots, q-1$, if condition (i) holds and querying $x_1, \ldots, x_{k-1}$ to the $\textsc{Enc}$ oracle in $\mathbf{S}_0$ results in $y_1, \ldots, y_{k-1}$ respectively, then querying x_k to $\textsc{Enc}$ will result in y_k. Then from Equation (21),

$$\mathsf{p}_{\mathbf{S}_0}(\tau, s) \geq \frac{\Pr[\mathrm{Bad}_0]}{N^{(t+1)}} \cdot \prod_{k=1}^{q} \frac{1}{N-k+1}\Big(1 - \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} \prod_{(i,j)\in\sigma} \frac{Z_s(i,j)}{N-q-p_j}\Big),$$

and thus

$$\begin{aligned}
\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} &\geq \prod_{k=1}^{q}\Big(1 - \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} \prod_{(i,j)\in\sigma} \frac{Z_s(i,j)}{N-q-p_j}\Big) \\
&\geq 1 - \sum_{k=1}^{q} \sum_{\sigma \in \mathcal{B}(\alpha_k[s], \beta_k[s])} \prod_{(i,j)\in\sigma} \frac{Z_s(i,j)}{N-q-p_j} \\
&\geq 1 - \sum_{k=1}^{q} \sum_{0\leq a<b\leq t} R_{a,b,k}[s] \cdot \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j)\in\sigma} \frac{Z_s(i,j)}{N-p_j-q}
\end{aligned}$$

where the second last inequality is due to the fact that $(1-x)(1-y) \geq 1-x-y$ for every $0 \leq x, y \leq 1$, and the last inequality is due to the fact that $R_{a,b,k}[s] = 1$ if $a = \alpha_k[s]$ and $b = \beta_k[s]$, and $R_{a,b,k}[s] \geq 0$ otherwise.

B Proof of Lemma 5

Fix $k \in \{1, \ldots, q\}$. Then

$$\mathbf{E}(R_{a,b,k}[S]) = \frac{p_1 \cdots p_a p_{b+1} \cdots p_t}{N^{t-(b-a)}}, \tag{22}$$

where $p_1 \cdots p_a$ is interpreted as 1 if $a = 0$, and likewise, $p_{b+1} \cdots p_t$ is interpreted as 1 if $b = t$. To justify Equation (22), note that $R_{a,b,k}[s] = 1$ if and only if (1) there are entries $(\texttt{prim}, 1, u_1, v_1), \ldots,$ $(\texttt{prim}, a, u_a, v_a)$ such that $u_1 = x_k \oplus L_0$ and $u_{r+1} = v_r \oplus L_r$ for every $r \in \{1, \ldots, a-1\}$, and (2) there are entries $(\texttt{prim}, b+1, u_{b+1}, v_{b+1}), \ldots, (\texttt{prim}, t, u_t, v_t)$ such that $u_{r+1} = v_r \oplus L_r$ for every $r \in \{b+1, \ldots, t-1\}$ and $v_t \oplus L_t = y_k$.

Now, for $\sigma \in \mathcal{B}(a,b)$ and each $(i,j) \in \sigma$, note that the random variable $Z_S(i,j)$ depends only on the subkeys $S_a, S_{a+1}, \ldots, S_{b-1}$ of S, whereas the random variable $R_{a,b,k}[S]$ depends only on the other subkeys of S. Hence $Z_S(i,j)$ and $R_{a,b,k}[S]$ are independent, and thus

$$\begin{aligned}
&\mathbf{E}\Big(\sum_{0\leq a<b\leq t} R_{a,b,k}[S] \sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j)\in\sigma} \frac{2Z_S(i,j)}{N}\Big) \\
&= \sum_{0\leq a<b\leq t} \mathbf{E}(R_{a,b,k}[S]) \cdot \mathbf{E}\Big(\sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j)\in\sigma} \frac{2Z_S(i,j)}{N}\Big) \\
&= \sum_{0\leq a<b\leq t} \frac{p_1 \cdots p_a p_{b+1} \cdots p_t}{N^{t-(b-a)}} \cdot \mathbf{E}\Big(\sum_{\sigma \in \mathcal{B}(a,b)} \prod_{(i,j)\in\sigma} \frac{2Z_S(i,j)}{N}\Big) .
\end{aligned}$$

We now claim that for any $0 \le a < b \le t$,

$$\mathbf{E}\Big(\sum_{\sigma\in\mathcal{B}(a,b)}\prod_{(i,j)\in\sigma}\frac{2Z_S(i,j)}{N}\Big) = \frac{2\cdot 3^{b-a-1}p_{a+1}\cdots p_b}{N^{b-a}} \tag{23}$$

We'll justify this claim later. Then

$$\begin{aligned}&\sum_{0\le a<b\le t}\frac{p_1\cdots p_a p_{b+1}\cdots p_t}{N^{t-(b-a)}}\cdot\mathbf{E}\Big(\sum_{\sigma\in\mathcal{B}(a,b)}\prod_{(i,j)\in\sigma}\frac{2Z_S(i,j)}{N}\Big)\\ &= \sum_{0\le a<b\le t}\frac{2\cdot 3^{b-a-1}p_1\cdots p_t}{N^t} = \sum_{\ell=0}^{t-1}\frac{2(t-\ell)3^\ell p_1\cdots p_t}{N^t}\ .\end{aligned}$$

To obtain the claimed result, what's left is to prove that

$$H(t) = \sum_{\ell=0}^{t-1}2(t-\ell)3^\ell \le 4^t - t - 1\ .$$

To justify this, note that

$$\begin{aligned}H(t) = \frac{3H(t)}{2} - \frac{H(t)}{2} &= \sum_{\ell=1}^{t}(t-\ell+1)3^\ell - \sum_{\ell=0}^{t-1}(t-\ell)3^\ell\\ &= 3^t - t - 1 + \sum_{\ell=0}^{t-1}3^\ell = \frac{3^{t+1}-1}{2} - t - 1 \le 4^t - t - 1\ .\end{aligned}$$

We now justify Equation (23). Fix $0 \le a < b \le t$. For each $\ell \in \{1, \ldots, b-a\}$, let B_ℓ be the subset of $\mathcal{B}(a,b)$ such that $|\sigma| = \ell$ for every $\sigma \in B_\ell$. Then

$$|B_\ell| = \binom{b-a-1}{\ell-1},$$

because there's a one-to-one correspondence between each $\sigma = \{(i_0,i_1),(i_1,i_2),\ldots,(i_{\ell-1},i_\ell)\} \in B_\ell$ and $\{i_1,\ldots,i_{\ell-1}\} \subseteq \{a,\ldots,b-1\}$. To justify Equation (23), note that its left-hand side can be rewritten as

$$\mathbf{E}\Big(\sum_{\ell=1}^{b-a}\frac{2^\ell}{N^\ell}\sum_{\sigma\in B_\ell}\prod_{(i,j)\in\sigma}Z_S(i,j)\Big) = \sum_{\ell=1}^{b-a}\frac{2^\ell}{N^\ell}\sum_{\sigma\in B_\ell}\mathbf{E}\Big(\prod_{(i,j)\in\sigma}Z_S(i,j)\Big)\ .$$

Moreover,

$$\mathbf{E}\Big(\prod_{(i,j)\in\sigma}Z_S(i,j)\Big) = \prod_{(i,j)\in\sigma}\mathbf{E}\Big(Z_S(i,j)\Big), \tag{24}$$

because for each $\ell \in \{1,\ldots,b-a\}$ and each $\sigma \in B_\ell$, the random variables $Z_S(i,j)$, with $(i,j)\in\sigma$, are independent. (Indeed, the randomness of $Z_S(i,j)$ is solely from the subkeys $S_{a+i},\ldots,S_{a+j-1}$ of S.) On the other hand, for each $\ell \in \{1,\ldots,b-a\}$ and each $a \le i < j \le b$, the expected value of each $Z_S(i,j)$ is exactly $p_{i+1}\cdots p_j/N^{j-i-1}$: entries $(\texttt{prim}, i+1, u_{i+1}, v_{i+1}), \ldots, (\texttt{prim}, j, u_j, v_j)$ in

τ will form a path between a vertex in V_i and another in V_j of the graph $G(s)$ if and only if $u_{r+1} = v_r \oplus L_r$ for every $r \in \{i, \ldots, j-1\}$, where $s = (L_0, \ldots, L_t)$. Then

$$\begin{aligned}
\sum_{\ell=1}^{b-a} \frac{2^\ell}{N^\ell} \sum_{\sigma \in B_\ell} \prod_{(i,j)\in\sigma} \mathbf{E}\Big(Z_S(i,j)\Big) &= \sum_{\ell=1}^{b-a} \frac{2^\ell}{N^\ell} \sum_{\sigma \in B_\ell} \prod_{(i,j)\in\sigma} \frac{p_{i+1}\cdots p_j}{N^{j-i-1}} \\
&= \sum_{\ell=1}^{b-a} 2^\ell \sum_{\sigma \in B_\ell} \frac{p_{a+1}\cdots p_b}{N^t} \\
&= \frac{2p_{a+1}\cdots p_b}{N^t} \sum_{\ell=1}^{b-a} |B_\ell| \cdot 2^{\ell-1} \\
&= \frac{2p_{a+1}\cdots p_b}{N^t} \sum_{\ell=1}^{b-a} \binom{b-a-1}{\ell-1} 2^{\ell-1} \\
&= \frac{2 \cdot 3^{b-a-1} p_{a+1}\cdots p_b}{N^t},
\end{aligned}$$

as claimed.

C Proof of Theorem 3

Let $\mathbf{S}_0$ be a system that provides the real game and $\mathbf{S}_1$ be a system that provides the ideal game. Let S be the random variable for the subkeys $(J_1, \ldots, J_t)$ in $\mathbf{S}_0$. Fix a transcript τ. Let $\mathcal{U}$ be the support of S. Our goal is to show that

$$\begin{aligned}
\mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) &= \sum_{s\in\mathcal{U}} \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \\
&\le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{4^t q p^t}{2^{t(k+n)}} \quad . \qquad (25)
\end{aligned}$$

In τ, there are two different types of entries:

- Enc/Dec *queries.* Queries to Enc$(1, x)$ returning y and Dec$(1, y)$ returning x are associated with an entry (enc, x, y).
- Prim/PrimInv *queries.* Queries to Prim(J, x), returning y, and to PrimInv(J, y), returning x, are associated with an entry $(\mathtt{prim}, J, x, y)$

Fix an $s = (J_1, \ldots, J_t)$ in the support of S. Let $p_i[s]$ be the number of entries $(\mathtt{prim}, J_i, \cdot, \cdot)$ in τ. We'll employ a transcript reduction, and our systems $\mathbf{S}'_0$ and $\mathbf{S}'_1$ provide the real and random games for KAC$[\pi, t]$, where $\pi(i, \cdot) = E(J_i, \cdot)$. Moreover, $\mathbf{S}_b$ and $\mathbf{S}'_b$ also share the same subkeys $L_0, \ldots, L_t$. In the transcript $\mathscr{R}(\tau, s)$, the Enc entries remain intact, and we only keep $(\mathtt{prim}, J, x, y)$ in τ if there is some $i \in \{1, \ldots, t\}$ such that $J = J_i$. We'll change that Prim entry to $(\mathtt{prim}, i, x, y)$, indicating that we're querying π_i. Since the Prim entries that we delete from (τ, s) are independent of π,

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s)} = \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s))} \quad . \qquad (26)$$

From Equation (17) in the proof of Theorem 1, the right-hand side of Equation (26) is at least

$$1 - \frac{4^t q}{2^{nt}} \cdot p_1[s] \cdots p_t[s] \enspace .$$

Hence

$$\begin{aligned}
&\sum_{s \in \mathcal{U}} \mathsf{ps}_{\mathbf{S}_1}(\tau, s) - \mathsf{ps}_{\mathbf{S}_0}(\tau, s) \\
&\le \sum_{s \in \mathcal{U}} \mathsf{ps}_{\mathbf{S}_1}(\tau, s) \cdot \frac{4^t q}{2^{nt}} \cdot p_1[s] \cdots p_t[s] \\
&= \frac{\mathsf{ps}_{\mathbf{S}_1}(\tau)}{2^k(2^k-1)\cdots(2^k-t+1)} \sum_{s \in \mathcal{U}} \frac{4^t q}{2^{nt}} \cdot p_1[s] \cdots p_t[s] \enspace ,
\end{aligned}$$

where we have used the fact that S is independent of everything else in $\mathbf{S}_1$. What's left is to prove that

$$\sum_{s \in \mathcal{U}} p_1[s] \cdots p_t[s] \le \frac{2^k(2^k-1)\cdots(2^k-t+1)}{2^{kt}} p^t \enspace . \tag{27}$$

For each finite set B and each integer $1 \le i \le |B|$, let $\mathcal{D}(B, i)$ be collection of tuples $(\ell_1, \ldots, \ell_i)$ such that $\ell_1, \ldots, \ell_i \in B$ and $\ell_1, \ldots, \ell_i$ are distinct. Let $V_1, V_2, \ldots$ be an enumeration of $\{0,1\}^k$. Now suppose that in τ, for each $i \in \{1, \ldots, 2^k\}$, there are δ_i entries $(\mathtt{prim}, V_i, \cdot, \cdot)$. Then

$$\begin{aligned}
\sum_{s \in \mathcal{U}} p_1[s] \cdots p_t[s] &= \sum_{(\ell_1, \ldots, \ell_t) \in \mathcal{D}(\{1, \ldots, 2^k\}, t)} \delta_{\ell_1} \cdots \delta_{\ell_t} \\
&\le \frac{2^k(2^k-1)\cdots(2^k-t+1)}{2^{kt}} \Big(\sum_{i=1}^{2^k} \delta_i\Big)^t \\
&\le \frac{2^k(2^k-1)\cdots(2^k-t+1)}{2^{kt}} p^t,
\end{aligned}$$

where the first inequality is due to Maclaurin's inequality.

D Proof of Proposition 1

Let $\mathbf{S}_0$ be a system that provides the real game and $\mathbf{S}_1$ be a system that provides the ideal game. Let S be the random variable for the subkeys $(J_1, \ldots, J_t)$ in $\mathbf{S}_0$. Fix a transcript τ. In τ, there are two different types of entries:

- Enc/Dec *queries.* Queries to $\textsc{Enc}(1, x)$ returning y and $\textsc{Dec}(1, y)$ returning x are associated with an entry (enc, x, y).
- Prim/PrimInv *queries.* Queries to $\textsc{Prim}(J, x)$, returning y, and to $\textsc{PrimInv}(J, y)$, returning x, are associated with an entry $(\mathtt{prim}, J, x, y)$

Let $\mathcal{U}$ be the support of S. For a key $s = (J_1, \ldots, J_t)$, we say that the subkey J_i of K is *duplicate* if there's some $j \ne i$ such that $J_j = J_i$; otherwise J_i is *distinct*. Let B_ℓ be the subset of $(\{0,1\}^k)^t$ such that any key $s \in B_\ell$ has exactly ℓ distinct subkeys.

Case 1: $t \geq 3$. Our goal is to show that

$$\begin{aligned}\mathsf{ps}_1(\tau) - \mathsf{ps}_0(\tau) &= \sum_{s \in \mathcal{U}} \mathsf{ps}_1(\tau, s) - \mathsf{ps}_0(\tau, s) \\ &\leq \mathsf{ps}_1(\tau) \cdot \left(\frac{4^t q p^t}{2^{(n+k)t}} + \frac{qt}{2^k} \left(\frac{t}{2^k} + \frac{4p}{2^{k+n}} \right)^{t-2} \right) \quad . \end{aligned} \tag{28}$$

For a key $s \in B_\ell$, let $p_i[s]$ be the number of entries $(\texttt{prim}, R_i, \cdot, \cdot)$ in τ for every $i \leq \ell$, where $R_1, \ldots, R_\ell$ are the distinct subkeys of s. We claim that for any $s \in B_\ell$,

$$\mathsf{ps}_1(\tau, s) - \mathsf{ps}_0(\tau, s) \leq \mathsf{ps}_1(\tau, s) \cdot \frac{4^\ell q p_1[s] \cdots p_\ell[s]}{2^{n\ell}} \quad . \tag{29}$$

We'll justify this claim later. When we sum both sides of Equation (29) over all $\ell \in \{1, \ldots, t\}$ and all $s \in B_\ell$, the left-hand side of Equation (28) is bounded by

$$\mathsf{ps}_1(\tau) \sum_{\ell=1}^{t} \frac{4^\ell q}{2^{n\ell}} \sum_{s \in B_\ell} \Pr[S = s] \cdot p_1[s] \cdots p_\ell[s] \quad . \tag{30}$$

Note that $B_{t-1} = \emptyset$. Moreover, from Equation (27) in the proof of Theorem 3,

$$\sum_{s \in B_t} \Pr[S = s] \cdot p_1[s] \cdots p_\ell[s] \leq \frac{p^t}{2^{kt}} \quad . \tag{31}$$

We then claim that for each $\ell \in \{1, \ldots, t-2\}$,

$$\sum_{s \in S_\ell} \Pr[S = s] \cdot p_1[s] \cdots p_\ell[s] \leq \binom{t-2}{\ell} \left(\frac{t}{2^k} \right)^{t-\ell-1} \left(\frac{p}{2^k} \right)^\ell \quad . \tag{32}$$

We postpone justifying this claim. From Equations (30), (31), and (32), the left-hand side of Equation (28) is bounded by

$$\begin{aligned}&\mathsf{ps}_1(\tau) \cdot \left(\frac{4^t q p^t}{2^{(n+k)t}} + \frac{qt^2}{2^{(t-1)k}} \sum_{\ell=0}^{t-2} \binom{t-2}{\ell} t^{t-2-\ell} (4p/2^n)^\ell \right) \\ &= \mathsf{ps}_1(\tau) \cdot \Big(\frac{4^t q p^t}{2^{(n+k)t}} + \frac{qt^2}{2^{(t-1)k}} (t + 4p/2^n)^{t-2} \Big),\end{aligned}$$

justifying the bound in Equation (28).

Justifying Equation (32). Let $V_1, V_2, \ldots$ be an enumeration of $\{0,1\}^k$. Now suppose that in τ, for each $i \in \{1, \ldots, 2^k\}$, there are δ_i entries $(\texttt{prim}, V_i, \cdot, \cdot)$. Let $\mathcal{D}(\ell)$ be collection of tuples $(r_1, \ldots, r_\ell)$ such that $r_1, \ldots, r_\ell \in \{1, \ldots, 2^k\}$ and $r_1, \ldots, r_\ell$ are distinct. Then the left-hand side of Equation (32) is a linear combination of $\delta_{r_1} \cdots \delta_{r_\ell}$ for $(r_1, \ldots, r_\ell) \in \mathcal{D}(\ell)$. In this linear combination, the coefficient of each term $\delta_{r_1} \cdots \delta_{r_\ell}$ is at most the probability that S contains exactly ℓ distinct subkeys, and those subkeys have indices $r_1, \ldots, r_\ell$. This is bounded by

$$\begin{aligned}
&\frac{1}{2^k(2^k-1)\cdots(2^k-\ell+1)} \cdot \binom{t}{\ell}\left(\frac{t-\ell-1}{2^k}\right)^{t-\ell-1} \\
&\leq \frac{1}{2^k(2^k-1)\cdots(2^k-\ell+1)} \cdot \binom{t}{\ell}\frac{(t-\ell-1)}{t}\left(\frac{t}{2^k}\right)^{t-\ell-1} \\
&= \frac{1}{2^k(2^k-1)\cdots(2^k-\ell+1)} \cdot \binom{t-1}{\ell}\left(\frac{t}{2^k}\right)^{t-\ell-1} \\
&\leq \frac{1}{2^k(2^k-1)\cdots(2^k-\ell+1)} \cdot t \cdot \binom{t-2}{\ell}\left(\frac{t}{2^k}\right)^{t-\ell-1} \quad .
\end{aligned}$$

Thus, to prove Equation (32), it suffices to show that

$$\sum_{(r_1,\ldots,r_\ell)\in\mathcal{D}(\ell)} \delta_{r_1}\cdots\delta_{r_\ell} \leq \frac{2^k(2^k-1)\cdots(2^k-\ell+1)}{2^{k\ell}}\left(\sum_{i=1}^{2^k}\delta_i\right)^{\ell} ,$$

but this is implied by Maclaurin's inequality.

Justifying Equation (29). Pick arbitrary $\ell \in \{0,\ldots,t\}$ and $s \in B_\ell$. Let $s = (J_1,\ldots,J_t)$, and let $1 \leq v_1 < \cdots < v_\ell \leq t$ be the indices such that each J_{v_i} is a distinct subkey of s. Let Z be the random variable for the subkey L_r and all N query-answer pairs of E_{J_r} of $\mathbf{S}_0$, for all $r \in \{1,\ldots,t\}\backslash\{v_1,\ldots,v_\ell\}$. We write $\mathsf{p}_{\mathbf{S}_0}(\tau,s,z)$ to be the probability that $\mathbf{S}_0$ answers queries according to τ, and that $S = s$ and $Z = z$. In this case $\mathsf{p}_{\mathbf{S}_1}(\tau,s,z)$ means the probability that $\mathbf{S}_1$ behaves according to the entries in (τ,s,z), and that $S \leftarrow\!\!\$\ (\{0,1\}^k)^t$ and $L_r \leftarrow\!\!\$\ \{0,1\}^n$ (for all $r \in \{1,\ldots,t\}\backslash\{v_1,\ldots,v_\ell\}$) independent of $\mathbf{S}_1$ agree with what's specified in (s,z). It suffices to prove that

$$\mathsf{p}_{\mathbf{S}_1}(\tau,s,z) - \mathsf{p}_{\mathbf{S}_0}(\tau,s,z) \leq \mathsf{p}_{\mathbf{S}_1}(\tau,s,z) \cdot \frac{4^\ell q p_1[s]\cdots p_\ell[s]}{2^{n\ell}},$$

since summing both sides for all z leads to Equation (29). Wlog, we only need to consider z such that $\mathsf{p}_{\mathbf{S}_1}(\tau,s,z) > 0$. We'll use the transcript-reduction method, in which the other systems $\mathbf{S}'_0$ and $\mathbf{S}'_1$ will provide the real and ideal games for an KAC$[\pi,\ell]$ respectively.

The family π is constructed as follows. Recall that z specifies subkey L_i for every index $i \in \{1,\ldots,t\}\backslash\{v_1,\ldots,v_\ell\}$. Let $\rho(i,\cdot) = E(J_i,\cdot) \oplus L_i$ for all $i \in \{1,\ldots,t\}\backslash\{v_1,\ldots,v_\ell\}$. For functions $f, g : \{0,1\}^n \to \{0,1\}^n$, let $f \circ g$ denote the cascade of f and g, meaning that $(f \circ g)(x) = g(f(x))$ for every $x \in \{0,1\}^n$. For every $j \in \{1,\ldots,\ell\}$, let $\pi(j,\cdot) = F_j(\cdot) \circ E(J_{v_j},\cdot)$, where $F_j(\cdot) = \rho(v_{j-1}+1,\cdot) \circ \cdots \circ \rho(v_j - 1,\cdot)$ and $v_0 = 0$. Then π is a family of independent, ideal permutations on $\{0,1\}^n$.

Next, the system $\mathbf{S}'_0$ uses the subkeys $(L_{v_1},\ldots,L_{v_\ell})$ of $\mathbf{S}_0$ as its key. Let $F_{\ell+1}(\cdot) = \rho(v_{\ell+1},\cdot) \circ \cdots \circ \rho(t,\cdot)$ if $v_\ell < t$, and let $F_{\ell+1}(\cdot)$ be the identity function on $\{0,1\}^n$ otherwise. Let f be the ideal permutation that $\mathbf{S}_1$ uses to answer Enc/Dec queries, and let f' be the permutation such that $f' \circ F_{\ell+1} = f$. The system $\mathbf{S}'_1$ uses f' to answer Enc/Dec queries. Hence on each Enc query $(1,x)$ and each $b \in \{0,1\}$, the answer of $\mathbf{S}_b$ is exactly $F_{\ell+1}(y)$, where y is the answer of $\mathbf{S}'_b$ for Enc query $(1,x)$.

For each $b \in \{0,1\}$, note that from (τ, s, z) in $\mathbf{S}_b$, since we know all queries/answers for $F_{\ell+1}$, we can "backtrack" to obtain q Enc queries/answers for $\mathbf{S}'_b$. Moreover, for each $i \in \{1, \ldots, \ell\}$, since we know all queries/answers for F_j, we can "connect" that with the $p_j[s]$ queries/answers for $E(J_{v_j}, \cdot)$ to obtain $p_j[s]$ queries/answers for π_j. Let $\mathscr{R}(\tau, s, z)$ be the corresponding transcript under this reduction. For system $\mathbf{S}_b$ to behave according to (τ, s, z), it means that (i) $\mathbf{S}'_b$ must behave according to $\mathscr{R}(\tau, s, z)$, and (ii) for all $r \in \{1, \ldots, t\} \backslash \{v_1, \ldots, v_\ell\}$, E_{J_r} and L_r must be as specified in z, and S must be s. Since $\mathbf{S}'_b$ is independent of (ii),

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s, z)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s, z)} = \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s, z))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z))} \enspace .$$

Hence, what's left is to prove that

$$\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z)) - \mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s, z)) \le \mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z)) \cdot \frac{4^\ell q p_1[s] \cdots p_\ell[s]}{2^{n\ell}},$$

but this follows from Equation (17) in the proof of Theorem 1.

Case 2: $t = 2$. Our goal is to show that

$$\begin{aligned} \mathsf{p}_{\mathbf{S}_1}(\tau) - \mathsf{p}_{\mathbf{S}_0}(\tau) &= \sum_{s \in \mathcal{U}} \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \\ &\le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \left(\frac{q(4p)^2}{2^{2(n+k)}} + \frac{4qp}{2^{2k+n}} + \frac{2q}{2^{k+n/2}} \right) \enspace . \end{aligned}$$

Note that $B_1 = \emptyset$ and as in Case 1,

$$\sum_{s \in B_2} \mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \le \mathsf{p}_{\mathbf{S}_1}(\tau) \cdot \frac{q(4p)^2}{2^{2(n+k)}} \enspace .$$

Since $|B_0| = |\mathcal{U}|/2^k$, what's left is to prove that, for any $s = (J, J) \in B_0$,

$$\mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \le \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \Big(\frac{4q \cdot p[s]}{2^n} + \frac{2q}{2^{n/2}} \Big), \tag{33}$$

where $p[s]$ is the number of entries $(\texttt{prim}, J, \cdot, \cdot)$ in τ. Fix $s = (J, J) \in B_0$. If $2q > 2^{n/2}$ then Equation (33) vacuously holds. Assume that $2q \le 2^{n/2}$, and thus $\frac{2q}{2^{n/2}} \ge \frac{4q^2}{2^n}$. Hence it suffices to prove that

$$\mathsf{p}_{\mathbf{S}_1}(\tau, s) - \mathsf{p}_{\mathbf{S}_0}(\tau, s) \le \mathsf{p}_{\mathbf{S}_1}(\tau, s) \cdot \frac{4q(q + p[s])}{2^n} \enspace .$$

Wlog, assume that $q + p[s] \le 2^n$, otherwise the bound is vacuous. Let us first explain the proof idea. We now can't directly employ the transcript-reduction technique to go to KAC, because both of our rounds use *the same* permutation. To resolve this issue, we handicap the following information: (i) the subkey L_0, and (ii) the round-1 outputs $E_J(x \oplus L_0)$ for all entries (enc, x, y). With this information, the first round can be ignored, and we can go to KAC$[\pi, 1]$, with $\pi = E_J$ and key (L_1, L_2). But now (ii) gives the adversary additional q Prim entries, and thus the adversary will have totally $q + p[s]$ Prim entries and q Enc entries.

Formally, let Z be the random variable for the subkey L_0 of $\mathbf{S}_0$ and $(\mathtt{prim}, J, x \oplus L_0, E_J(x \oplus L_0))$, for all entries (enc, x, y) in τ. We write $\mathsf{p}_{\mathbf{S}_0}(\tau, s, z)$ to be the probability that $\mathbf{S}_0$ answers queries according to τ, and that $S = s$ and $Z = z$. In this case $\mathsf{p}_{\mathbf{S}_1}(\tau, s, z)$ means the probability that $\mathbf{S}_1$ behaves according to the entries in (τ, s, z), and that $S \leftarrow\!\!\$\ (\{0,1\}^k)^2$ and $L_0 \leftarrow\!\!\$\ \{0,1\}^n$ independent of $\mathbf{S}_1$ agree with what's specified in (s, z). It suffices to prove that

$$\mathsf{p}_{\mathbf{S}_1}(\tau, s, z) - \mathsf{p}_{\mathbf{S}_0}(\tau, s, z) \le \mathsf{p}_{\mathbf{S}_1}(\tau, s, z) \cdot \frac{q(q + p[s])}{2^n} \enspace .$$

Wlog, we only need to consider z such that $\mathsf{p}_{\mathbf{S}_1}(\tau, s, z) > 0$. Let $\pi = E_J$. We'll use the transcript-reduction method, in which the systems $\mathbf{S}'_0$ and $\mathbf{S}'_1$ will provide the real and ideal games for $\mathrm{KAC}[\pi, 1]$. The system $\mathbf{S}'_0$ uses the subkeys (L_1, L_2) of $\mathbf{S}_0$ as its key. Let f be the ideal permutation that $\mathbf{S}_1$ uses to answer its $\textsc{Enc}/\textsc{Dec}$ queries. Let L be the subkey specified in z. The system $\mathbf{S}'_1$ uses the permutation f' such that $f'(E_J(x \oplus L)) = f(x)$ for every $x \in \{0,1\}^n$ to answer its $\textsc{Enc}/\textsc{Dec}$ queries; this f' is also an ideal permutation on $\{0,1\}^n$.

Construct $\mathscr{R}(\tau, s, z)$ as follows. For any entry $(\mathtt{prim}, J, u, v)$ in (τ, z), change it to $(\mathtt{prim}, 1, u, v)$. For each entry (enc, x, y) in τ, look for the unique entry $(\mathtt{prim}, 1, u, v)$ such that $x \oplus u$ is the subkey specified in z, and then add (enc, v, y) to $\mathscr{R}(\tau, s, z)$. For each $b \in \{0,1\}$, for system $\mathbf{S}_b$ to behave according to (τ, s, z), it means that (i) $\mathbf{S}'_b$ must behave according to $\mathscr{R}(\tau, s, z)$, and (ii) L_0 must be as specified in z, and S must be s. Since $\mathbf{S}'_b$ is independent of (ii),

$$\frac{\mathsf{p}_{\mathbf{S}_0}(\tau, s, z)}{\mathsf{p}_{\mathbf{S}_1}(\tau, s, z)} = \frac{\mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s, z))}{\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z))} \enspace .$$

Hence, what's left is to prove that

$$\mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z)) - \mathsf{p}_{\mathbf{S}'_0}(\mathscr{R}(\tau, s, z)) \le \mathsf{p}_{\mathbf{S}'_1}(\mathscr{R}(\tau, s, z)) \cdot \frac{4q(q + p[s])}{2^n},$$

but this follows from Equation (17) in the proof of Theorem 1.

E XC's relation with Gaži and Tessaro's 2XOR

Gaži and Tessaro (GT) [19] consider the following variant $\mathrm{2XOR}[E] : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ of $\mathrm{XC}[2, E]$, for $\mathcal{K} = \{0,1\}^k \times \{0,1\}^{2n}$. Let $\Delta \in \{0,1\}^k$ be a nonzero constant. Then for any $K = (J, L_1, L_2) \in \mathcal{K}$ and any $x \in \{0,1\}^n$, $\mathrm{2XOR}[E](K, x) = E_{J \oplus \Delta}(E_J(x \oplus L_1) \oplus L_2)$. See Fig. 7 for an illustration of the 2XOR construction. GT show that in the ideal-cipher model, for any adversary A that makes 2^n queries to $\textsc{Enc}/\textsc{Dec}$, and p queries to $\textsc{Prim}/\textsc{PrimInv}$, $\mathsf{Adv}^{\pm\mathrm{prp}}_{\mathrm{2XOR}[E]}(A) \le 4(p^2/2^{2k+n})^{1/3}$. If we ignore the exponent $1/3$, which is an artifact of GT's proof technique, then at the first glance, the construction $\mathrm{2XOR}[E]$ may appear as strong as $\mathrm{XC}[E, 2]$. However, we'll show an attack to demonstrate that in the multi-user setting, the former is much weaker than the latter. We'll make $O(2^k)$ queries to $\textsc{Enc}/\textsc{Dec}$, and $O(2^n)$ queries to $\textsc{Prim}/\textsc{PrimInv}$, and get advantage at least $1/2$ over $\mathrm{2XOR}[E]$. On the other hand, from Theorem 4, such an attack can only hope to get advantage about $2^{k - 2\min\{k,n\}}$ over $\mathrm{XC}[E, 2]$. For practical choices of E such as AES ($k = n = 128$) or DES ($k = 56$ and $n = 64$), the gap between $\mathrm{2XOR}[E]$ and $\mathrm{XC}[E, 2]$ is large.

The attack is as follows. Let $\ell = 1 + \lceil (2k+3)/(n-1) \rceil$. Assume that $\ell = O(1)$ and $\ell \le 2^{n-1}$, which holds for all practical values of n and k. Pick arbitrary distinct n-bit strings $y_1, \ldots, y_\ell$. For each $i = 1, \ldots, 2^k$, query $\textsc{Dec}(i, y_1), \ldots, \textsc{Dec}(i, y_\ell)$ to get answers $x_1[i], \ldots, x_\ell[i]$ respectively, and store $(x_1[i] \oplus x_2[i]) \,\|\, \cdots \,\|\, (x_1[i] \oplus x_\ell[i])$ in a hash table H. Pick an arbitrary key $J \in \{0,1\}^k$ and query $\textsc{PrimInv}(J, \cdot)$ on the entire domain, and maintain the queries/answers in an array R, meaning that $R[x] = \textsc{PrimInv}(J, x)$ for every $x \in \{0,1\}^n$. Then, query $\textsc{PrimInv}(J \oplus \varDelta, y_1), \ldots, \textsc{PrimInv}(J \oplus \varDelta, y_\ell)$ to get answers $u_1, \ldots, u_\ell$ respectively. Next, for each $s \in \{0,1\}^n$, process as follows. Let $v_j = s \oplus (u_1 \oplus u_j)$ for every $j \le \ell$. If there is an entry in H matching $(R[v_1] \oplus R[v_2]) \,\|\, \cdots \,\|\, (R[v_1] \oplus R[v_\ell])$ then terminate and output 1. Finally, at the end of the loop, output 0. The cost of the attack is $O(n(2^k + 2^n))$, in both time and space.

For analysis, first consider the real game. For each individual user, the chance that its key J matches the corresponding key in the attack is only $1/2^k$. However, since we have 2^k users, the chance that there's *some* user whose key J matches the corresponding key in the attack is $1 - (1 - 1/2^k)^{2^k} \ge 1 - 1/e$, where e is the base of the natural logarithm. Moreover, if there's such a match then the adversary will always output 1. Hence in the real game, the chance that the adversary outputs 1 is at least $1 - 1/e$. On the other hand, in the ideal game, the chance that the adversary outputs 1 is at most

$$\frac{2^{2k}}{(2^n - 1) \cdots (2^n - \ell + 1)} \le \frac{2^{2k}}{2^{(n-1)(\ell-1)}} \le \frac{1}{8} \enspace .$$

Hence the adversary wins with advantage at least $1 - 1/e - 1/8 \ge 1/2$.

In $2\mathrm{XOR}[E]$, one uses two related keys J and $J \oplus \varDelta$ for E. If one instead uses two keys J and J', say $J' \leftarrow_{\$} \{0,1\}^k \backslash \{J\}$, then the new scheme is still vulnerable to a variant of the attack above. Recall that in the original attack, we only query query $\textsc{PrimInv}(J' \oplus \varDelta, y_1), \ldots, \textsc{PrimInv}(J' \oplus \varDelta, y_\ell)$ and then search the hash table for $J' = J \oplus \varDelta$. Now we'll do that for every $J' \in \{0,1\}^k \backslash \{J\}$. We'll need $O(2^k)$ queries for $\textsc{Enc}/\textsc{Dec}$ and $O(2^k + 2^n)$ queries for $\textsc{Prim}/\textsc{PrimInv}$. The storage cost is still $O(n(2^k + 2^n))$ but the running time is now $O(n(2^{2k} + 2^n))$.