Security of Symmetric Primitives against Key-Correlated Attacks

Aisling Connolly[1,2], Pooya Farshim[2,3], and Georg Fuchsbauer[2,3]

[1] Ingenico Group, Paris, France
[2] Département d'informatique École normale supérieure (DIENS), Centre national de la recherche scientifique (CNRS), Université Paris Sciences et Lettres (PSL), Paris, France
[3] Inria, Paris, France
{aisling.connolly,pooya.farshim,georg.fuchsbauer}@ens.fr

Abstract. We study the security of symmetric primitives against *key-correlated attacks* (KCA), whereby an adversary can arbitrarily correlate keys, messages, and ciphertexts. Security against KCA is required whenever a primitive should securely encrypt key-dependent data, even when it is used under related keys. KCA is a strengthening of the previously considered notions of related-key attack (RKA) and key-dependent message (KDM) security. This strengthening is strict, as we show that 2-round Even–Mansour fails to be KCA secure even though it is *both* RKA and KDM secure. We provide feasibility results in the ideal-cipher model for KCAs and show that 3-round Even–Mansour is KCA secure under key offsets in the random-permutation model. We also give a natural transformation that converts any authenticated encryption scheme to a KCA-secure one in the random-oracle model. Conceptually, our results allow for a unified treatment of RKA and KDM security in idealized models of computation.

Keywords: Key-correlated attack · related-key attack · key-dependent-message attack · ideal-cipher model · random-oracle model · authenticated encryption · xkcd.

1 Introduction

Cryptographic algorithms are subject to a multitude of threats. Many of these threats are accounted for in the theoretical security analysis carried out by cryptographers, but not all. For example, early on, the seminal paper of Goldwasser and Micali [GM84] pointed out that guarantees of semantic security may break down if the adversary sees encryptions of the secret key. Formal analyses of protocols can also become moot [Bih94b, Bih94a] when the assumption that cryptosystems are run on independently generated keys no longer holds. A number of works have analyzed the security of cryptosystems in the presence of key-dependent messages or when different keys are generated in dependent ways (see the related work section below). We continue this line of work and ask to what extent basic cryptosystems (such as blockciphers and symmetric encryption) can resist attacks that exploit correlated inputs.

1.1 Motivation

Our motivation for studying correlated-input security is twofold. We are interested in settings where a cryptosystem may be run on related keys—either by design or due to attacks—to securely encrypt messages that depend on the encryption key. Suppose a user stores a secret key K on its hard drive. An adversary may be able to tamper with this key, for example flip some of its bits and change it to $K \oplus \Delta$ for some bit string Δ. It may

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2019, No. 3, pp. 193–230
DOI:10.13154/tosc.v2019.i3.193-230

then obtain a full-disk encryption under this key. It is not clear what security assertions can be made, as this setting falls outside both the related-key attack (RKA) and the key-dependent message (KDM) models. Indeed, the RKA model only allows the adversary to obtain encryptions of the form $\mathsf{Enc}(\phi(K), M)$, for functions ϕ mapping keys to keys, but for key-independent messages, while KDM accounts for key-dependent encryptions of type $\mathsf{Enc}(K, \psi(K))$, for functions ψ mapping keys to messages, but under untampered keys. In the described attack, the adversary obtains $\mathsf{Enc}(K \oplus \Delta, K \oplus \Delta)$. This is not covered by either of these models since both the key and the message are correlated with the original key. Other applications of KCAs include efficient garbling of XORs [App16], where KCA security (called RK-KDM there) with respect to linear functions or the form $\alpha \cdot K \oplus \Delta$ for a bit α are used.

These settings require a stronger security notion, which is what we introduce here. The Key-Correlated Attacks (KCAs) model lets the adversary obtain encryptions of key-dependent messages under related keys. Generally, wherever there is a possibility of both RKAs and KDM attacks, i.e., key-correlated encryptions of the form $\mathsf{Enc}(\phi(K), \psi(K))$, there is good chance that the actual security needed is KCA security. A typical use case is when the round functions of a block cipher are keyed via related keys, and the construction is used to encrypt key-dependent data.

In our model, for generality, simplicity and strength, we symmetrically allow for key-dependent *ciphertexts*, that is, the adversary can see $\mathsf{Dec}(\phi(K), \psi(K))$. Such settings arise when the decryption algorithm of a blockcipher is run during encryption, which is for example the case in the triple DES construction [BR06], the ElmD construction [BDMN16], or in amplification theorems for blockciphers [MP04, CPS14].

Our second motivation is conceptual in that KCA provides a unified approach to RKA and KDM security analyses of symmetric primitives. More concretely, our goal is to prove KCA feasibility theorems and then derive RKA and KDM security as simple corollaries. This allows for reuse of security proofs and identifies classes of permitted attacks more generally, while leading to stronger security results.

1.2 Related work

RKA security. Knudsen and Biham [Knu93, Bih94b, Bih94a] initiated the study of RKAs and Bellare and Kohno [BK03] gave a theoretical treatment. High-profile RKAs on AES were discovered by Biryukov et al. [BKN09, BK09]. The RKA model was extended by Albrecht et al. [AFPW11] to account for attacks that depend on an ideal primitive [Har09, Ber10]. The RKA security of Feistel networks [BF15] and Even–Mansour ciphers [FP15, CS15] have been studied. Bellare, Cash, and Miller [BCM11] present a comprehensive treatment of RKA security for various cryptographic primitives.

KDM security. Goldwasser and Micali [GM84] already hinted at the need for KDM security. The first use of KDM security appears in the work of Camenisch and Lyskanskaya [CL01] for anonymous credentials. Black, Rogaway, and Shrimpton [BRS03] formulated KDM security for symmetric encryption and proved its feasibility in the random-oracle model. Halevi and Krawczyk [HK07] later gave feasibility results in the standard model. Bellare and Keelveedhi [BK11] studied KDM in the context of authenticated encryption. Bellare, Cash, and Keelveedhi [BCK11] give a generic construction of a tweakable blockcipher from a blockcipher which is KDM secure. More recently, Farshim, Khati, and Vergnaud [FKV17] studied KDM security for the ideal cipher and the iterated Even–Mansour constructions. In the asymmetric setting the first feasibility result in the standard model for rich classes of functions was by Boneh et al. [BHHO08]. Camenisch, Chandran and Shoup [CCS09] gave a KDM-CCA secure public-key encryption scheme.

Correlated inputs. Study of security under correlated inputs goes back to the work of Ishai et al. [IKNP03] as correlation-robustness in the design of oblivious transfer protocols.

Correlated-input security was made explicit for hash functions by Goyal, O'Neill, and Rao [GOR11], who show relations with related-key attacks. The work of Böhl, Davies and Hofheinz [BDH14] considers related-key attacks in the presence of key-dependent messages. Their RKA-KDM security could be considered a natural analogue of our model for public-key encryption. They construct schemes that achieve their notion based on number-theoretic assumptions such as DDH, LWE, QR, or DCR. Applebaum [App16] gives an RKA-KDM symmetric encryption scheme based on the LPN assumption.

1.3 Contributions

Building on the above line of works, we formulate a new security model incorporating and strengthening both the RKA and KDM models. We speak of key-correlated attack (KCA) in this context, a name that is loosely inspired by the introduction of correlated-input attacks against hash functions [GOR11] (note that the notion of *key-dependent input attacks* has already been used by Halevi and Krawczyk [HK07]). We give appropriate definitions of security under key-correlated attacks that relate well to the standard RKA and KDM security notions. Our definition extends that in [App16] for randomized symmetric encryption under chosen-plaintext attacks to the setting of authenticated encryption with associated data (AEAD).

We start with comparing our notion to existing ones. After proving that KCA implies RKA and KDM security, we show that KCA security is strictly stronger than even simultaneously having RKA and KDM security. We give a natural separation by demonstrating a KCA attack on the 2-round Even–Mansour cipher, which was shown to satisfy both RKA and KDM security in two previous works [FP15, FKV17].

After defining KCA and showing a separation result, we study feasibility of KCAs. Our starting point is the ideal-cipher model, in which all parties have oracle access to a keyed random permutation in both directions. We cannot allow arbitrary dependencies of keys and messages as otherwise "trivial" attacks, which work against any scheme, arise. To exclude these and thus obtain a meaningful notion, we restrict the classes of allowed dependencies. We show that if they satisfy appropriate notions of *key-unpredictability* and *claw-freeness* then the ideal cipher satisfies KCA security. Roughly speaking, key-unpredictability requires that the adversary does not obtain encryptions or decryption under predictable keys. Claw-freeness, on the other hand, requires that the inputs are distinct, and so repetition pattern of the outputs cannot be exploited. Analogues of these notions were previously considered in the RKA and KDM settings.

In our setting we require a third condition, which we call *cross-key-claw detectability*, that allows us to deal with claws across encryption and decryption queries. This notion is sufficiently weak so that in the RKA and KDM setting it automatically follows from claw-freeness. In the KCA setting, however, it does not necessarily, as it restricts claws on keys.[1] Two results on the RKA and KDM security of the ideal cipher by Bellare and Kohno [BK03] and Farshim, Khati, Vergnaud [FKV17] respectively, fall out as natural corollaries of our theorem.

Turning to concrete constructions, we analyze the KCA security of the iterated Even–Mansour cipher with *three* rounds in the random-permutation model. We show that with reuse of keys (which is known to be necessary for RKA security [FP15]) and using different permutations (which is necessary for KDM security whenever keys are reused [BW99, FKV17]) we can provably achieve security against key-correlated attacks that concurrently encrypt messages M or offsets of key of the form $K \oplus \Delta_2$ under other offsets of the key of the form $K \oplus \Delta_1$. This strengthens two feasibility results due to Farshim, Khati, Vergnaud [FKV17] and Farshim and Procter [FP15].

[1] Claw-freeness can be modified to key-claw-freeness across encryption and decryption so that cross-key-claw-freeness is automatic. But in this case the reach of our feasibility results do not extend to the KDM setting since under KDM keys always collide.

From a technical point of view, the novelty of our KCA proof for 3-round Even–Mansour is that we keep the outer permutations partially consistent with the replaced forgetful oracles as well as the permutation oracles. For legal queries we show this can be done with overwhelming probability, while a detection algorithm will allow us to identify illegal queries and reject them. This proof thus deviates from previous works in which oracles are fully decoupled. As a result we also obtain a different (albeit somewhat more complex) way to prove the RKA security of 3-round Even–Mansour against key offsets by replacing the outer (rather the inner) permutation [FP15].

We end the paper by showing how to generically transform any AE-secure AEAD scheme to one which is a KCA-secure in the random-oracle model by hashing the key with nonces. For this result we only require the set of allowed functions to be unpredictable, as nonces automatically prevent repetitions due to claws in the functions. In contrast to previous work by Bellare and Keelveedhi [BK11] on similar transforms for achieving KDM-security, our scheme is secure with *key-dependent nonces and headers*. Although key-dependent headers are briefly discussed in [BK11], security with respect to key-dependent nonces is not considered at all. Arguably, however, there is a stronger case for the key-dependency of nonces than of headers: when nonces are randomly chosen they might become correlated with the key due to, e.g., bad generation of random numbers. For key-dependent headers, Bellare and Keelveedhi give a negative result, by having the adversary exploit the pattern of decryption errors (either $\perp$ for an illegal query or 0 for failure in authenticity) to recover the key. In our setting, however, the decryption oracle only returns a single error symbol, which enables security under key-dependent inputs. If our model were modified to also have distinct error symbols, an attack similar to that in [BK11] would arise. We note that in these settings one might be able to obtain non-trivial feasibility results by requiring a form claw-freeness.[2]

Our work leaves a number of open questions to be addressed in this area: How many rounds are needed for other ciphers (such as Feistel networks) in order to achieve provable KCA security? Do modes of operation (such as CBC) provide KCA security assuming that their underlying blockciphers are KCA secure? Does KCA help in the cryptanalysis of concrete blockciphers?

Organization. We start with basic notation and preliminaries in Sec. 2. In Sec. 3, we define KCA security for blockciphers and study its relation to RKA and KDM security before showing two separation results in Sec. 4. In Sec. 5 we study KCA in the ideal-cipher model and in Sec. 6 we prove 3-round Even–Mansour KC-CCA secure for offsets. Sec. 7 contains our generic construction of a KCA-secure authenticated encryption scheme from any AE-secure one.

2 Preliminaries

Notation. We let $\{0,1\}^n$ denote the set of bit strings of length n and $\{0,1\}^*$ the set of all finite-length bit strings. For two bit strings X and Y, $X|Y$ denotes their concatenation and (X,Y) denotes a uniquely (and efficiently) decodable encoding of X and Y. The length of X is denoted by $|X|$. By $x \twoheadleftarrow S$ we mean sampling x uniformly from set S, whereas by $y \twoheadleftarrow \mathcal{A}(x)$ we mean the action of assigning the output of the randomized algorithm $\mathcal{A}$ on input x to y. We denote appending element X (resp., a list L') to a list L by $L : X$ (resp., $L : L'$).

We adopt the code-based game-playing language of Bellare and Rogaway [BR06], for which all lists are initialized to empty and all bad flags to false. Following the concrete security approach, we will primarily content ourselves with defining advantages and providing concrete reductions, without dwelling too much on the question when a

[2] Consistently, the attack in [BK11] exploits claws in the key-dependent headers.

scheme is actually deemed secure or not (e.g., for a sufficiently large class of adversaries, the advantages are sufficiently small). One can of course easily recast our work in an asymptotic framework (where for all probabilistic polynomial-time adversaries advantages should be negligible in the security parameter λ, i.e., in $\lambda^{-\omega(1)}$).

Blockciphers. Given a non-empty finite set $\mathcal{K}$ and a non-empty set $\mathcal{M}$, called the key space and the message space respectively, we let $\mathrm{Block}(\mathcal{K}, \mathcal{M})$ denote the set of all functions $\mathsf{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ such that for each $K \in \mathcal{K}$ the map $\mathsf{E}(K, \cdot)$ is (1) a permutation on $\mathcal{M}$ and (2) length-preserving in the sense that for all $M \in \mathcal{M}$ we have that $|\mathsf{E}(K, M)| = |M|$. Such an E uniquely defines its inverse $\mathsf{D}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$. A blockcipher for key space $\mathcal{K}$ and message space $\mathcal{M}$ is a tuple of efficient algorithms $\mathrm{BC} := (\mathsf{E}, \mathsf{D})$ such that $\mathsf{E} \in \mathrm{Block}(\mathcal{K}, \mathcal{M})$ and D is its inverse. We assume that the keys of a blockcipher are chosen uniformly from the key space $\mathcal{K}$, which is typically $\{0,1\}^k$ for some $k \in \mathbb{N}$ called the key length. Algorithm E is the deterministic encryption algorithm $\mathsf{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$. Typically $\mathcal{M} = \{0,1\}^n$ for some $n \in \mathbb{N}$ called the block length. Algorithm D is the deterministic decryption algorithm $\mathsf{D}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$. A blockcipher is correct if for all $K \in \mathcal{K}$ and $M \in \mathcal{M}$ we have $\mathsf{D}(K, \mathsf{E}(K, M)) = M$ and also $\mathsf{E}(K, \cdot)$ is length-preserving (which follows from decryptability if $\mathcal{M} = \{0,1\}^n$). A (public) permutation on $\mathcal{M}$ is a blockcipher with a singleton key space $\mathcal{K} = \{\varepsilon\}$. For variable input length blockciphers, we require that for each n, $\{0,1\}^n$ is either contained in $\mathcal{M}$, or $\{0,1\}^n \cap \mathcal{M} = \emptyset$. We denote a permutation with P and its inverse with P^{-}.

Ideal ciphers. The ideal cipher for key space $\mathcal{K}$ and message space $\mathcal{M}$ is the uniform distribution over $\mathrm{Block}(\mathcal{K}, \mathcal{M})$. The ideal-cipher model (ICM) for given key and message spaces $\mathcal{K}$, $\mathcal{M}$ is a model of computation where all parties, honest or otherwise, have oracle access to a uniformly random element in $\mathrm{Block}(\mathcal{K}, \mathcal{M})$ and its inverse. The ideal-cipher model when restricted to $\mathcal{K} = \{\varepsilon\}$ gives rise to the random-permutation model (RPM). We abbreviate $\mathrm{Block}(\{0,1\}^k, \{0,1\}^n)$ by $\mathrm{Block}(k, n)$ and $\mathrm{Block}(\{\varepsilon\}, \{0,1\}^n)$ by $\mathrm{Perm}(n)$.

3 The Model

3.1 Concept and definitions

Correlation-derivation function (CDF). A correlation-derivation function (CDF) is a circuit of the form

$$\xi\colon \mathcal{K} \longrightarrow \mathcal{K} \times \mathcal{M} \ .$$

A set of such functions is called a CDF set. Throughout the paper we denote CDF sets by Ξ and require that membership of a CDF set can be efficiently decided. We will define key-correlated security primitives relative to two CDF sets Ξ^e and Ξ^d that describe allowed encryption and decryption queries, respectively.

KCA-secure blockciphers. Let BC be a blockcipher with key space $\mathcal{K} = \{0,1\}^k$ and message (and ciphertext) space $\mathcal{M} = \{0,1\}^n$. Let Ξ^e and Ξ^d be CDF sets for the input space $\mathcal{K} \times \mathcal{M}$. The KC-CCA advantage of an adversary $\mathcal{A}$ against BC is defined by

$$\mathbf{Adv}^{\text{kc-cca}}_{\mathrm{BC}}(\Xi^e, \Xi^d, \mathcal{A}) := 2 \cdot \Pr\left[\text{KC-CCA}^{\Xi^e, \Xi^d, \mathcal{A}}_{\mathrm{BC}}\right] - 1 \ , \tag{1}$$

where game $\text{KC-CCA}^{\Xi^e, \Xi^d, \mathcal{A}}_{\mathrm{BC}}$ is given in Fig. 1. In this game, the adversary's goal is to decide whether (case $b = 1$) its oracles are using the blockcipher $\mathrm{BC} = (\mathsf{E}, \mathsf{D})$ or (case $b = 0$, the *ideal* case) a random permutation iE and its inverse iD. Lists ML and CL prevent the adversary from trivially winning the game. Otherwise the adversary could, for instance, recover the challenge key K^* by querying an encryption (or decryption) of K^* and subsequently ask for decryption (or encryption) of the oracle's reply.

Game $\text{KC-CCA}_{\text{BC}}^{\Xi^e,\Xi^d,\mathcal{A}}$:	Proc. $\text{KCE}_{\text{NC}}(\xi^e)$:	Proc. $\text{KCD}_{\text{EC}}(\xi^d)$:
$b \twoheadleftarrow \{0,1\}$	$(K, M) \leftarrow \xi^e(K^*)$	$(K, C) \leftarrow \xi^d(K^*)$
$(\mathsf{iE}, \mathsf{iD}) \twoheadleftarrow \text{Block}(\mathcal{K}, \mathcal{M})$	if $(K, M) \in \mathsf{ML}$: return $\perp$	if $(K, C) \in \mathsf{CL}$: return $\perp$
$K^* \twoheadleftarrow \mathcal{K}$	$C \leftarrow \mathsf{E}(K, M)$	$M \leftarrow \mathsf{D}(K, M)$
$b' \twoheadleftarrow \mathcal{A}^{\text{KCE}_{\text{NC}},\text{KCD}_{\text{EC}}}$	if $b = 0$: $C \leftarrow \mathsf{iE}(K, M)$	if $b = 0$: $M \leftarrow \mathsf{iD}(K, M)$
return $(b' = b)$	$\mathsf{CL} \leftarrow \mathsf{CL} : (K, C)$	$\mathsf{ML} \leftarrow \mathsf{ML} : (K, M)$
	return C	return M

Figure 1: Game defining the (Ξ^e, Ξ^d)-KC-CCA security of a blockcipher $\text{BC} = (\mathsf{E}, \mathsf{D})$ with key space $\mathcal{K}$ and message space $\mathcal{M}$. We require that $\xi^e \in \Xi^e$ and $\xi^d \in \Xi^d$ for all queries.

We note that for two sets $\Xi_1^e \subseteq \Xi_2^e$ and $\Xi_1^d \subseteq \Xi_2^d$ security against (Ξ_2^e, Ξ_2^d)-KC-CCA implies security against (Ξ_1^e, Ξ_1^d)-KC-CCA. Whenever $\Xi^d = \emptyset$ we obtain a chosen-plaintext attack model.

Note. Analogues of KCA security can be formulated for hash functions and pseudorandom generators, which become equivalent to correlated-input security for hash functions [GOR11] and RKA security for PRGs [BCM11].

3.2 Examples

As examples of KC functions, suppose that related keys, described by functions from some set Φ, are used within the specification of an encryption scheme (an example are the 3GPP protocols [IK04]). Suppose further that the scheme is used to encrypt messages that depend on the key in ways represented by functions from a set Ψ. The overall effect is that an adversary has access to ciphertexts corresponding to *key-dependent messages under related keys*. In other words, the adversary can see key-correlated ciphertexts for functions:

$$\xi\colon K \mapsto (\phi(K), \psi(K)) \text{ with } \phi \in \Phi \text{ and } \psi \in \Psi \ , \quad \text{and}$$
$$\xi\colon K \mapsto (\phi(K), M) \text{ with } \phi \in \Phi \text{ and } M \in \mathcal{M} \ .$$

Alternatively, suppose an adversary has access to encryptions under related keys with respect to Φ through, say, injection of faults that change bits of a hardware-stored encryption key [BDL97, BS97].[3] If the scheme is used to encrypt key-dependent messages with respect to Ψ, the adversary would be able to launch a KCA for the functions:

$$\xi\colon K \mapsto \big(\phi(K), \psi(\phi(K))\big) \text{ with } \phi \in \Phi \text{ and } \psi \in \Psi \ , \quad \text{and}$$
$$\xi\colon K \mapsto (\phi(K), M) \text{ with } \phi \in \Phi \text{ and } M \in \mathcal{M} \ .$$

The KCA model thus captures, among other things, a variety of *joint* RKA and KDM attacks on a blockcipher.

3.3 Relation with RKA and KDM

Let id denote the identity function on a key space $\mathcal{K}$ and let Γ denote the set of all constant functions $K \mapsto M$ for $M \in \mathcal{M}$. We identify a pair of functions (ϕ, ψ), with ranges $\mathcal{K}$ and $\mathcal{M}$ respectively, with the correlation-derivation function $K \mapsto (\phi(K), \psi(K))$.

CPA/CCA-only sets. We call a pair (Ξ^e, Ξ^d) CCA-only if $\Xi^e = \Xi^d = \{id\} \times \Gamma$. (The adversary can only make regular encryption and decryption queries.) We call (Ξ^e, Ξ^d) CPA-only if $\Xi^e = \{id\} \times \Gamma$ and $\Xi^d = \emptyset$.

RKA/KDM-only sets. We call a pair (Ξ^e, Ξ^d) RKA-only if $\Xi^e = \Xi^d = \Phi \times \Gamma$ for an RKA set Φ of functions mapping keys to keys. We call (Ξ^e, Ξ^d) KDM-only if $\Xi^e = \{id\} \times \Psi$

[3] KCA security does not imply security against fault attacks in general.

and $\Xi^d = \{id\} \times \Gamma$ (i.e., no key-dependent ciphertexts allowed [FKV17]) for a KDM set Ψ of functions mapping keys to messages. (The CPA versions are defined analogously by demanding that $\Xi^d = \emptyset$.) We assume that any KDM set Ψ contains Γ, as it is natural to always allow for chosen plaintexts.

We show that our definition of KC security extends the standard RKA and KDM definitions for blockciphers, which we recall in Appendix A. In particular, for KDM-only sets we have that KCA and KDM security are equivalent and similarly for RKA-only sets KCA and RKA security are equivalent. For two of the four implications, we assume *claw-freeness* of the correlation-derivation functions. This requires that it is hard to find two different functions that have the same output given a random input and is defined formally in Sec. 5 (p. 202). Claw-freeness is required because the KCA game returns $\perp$ whenever claws are detected whereas the RKA game does not. The proof of the following proposition can be found in Appendix A.

Proposition 1. *Let* BC *be a blockcipher. Let* Ξ^e *and* Ξ^d *be two CDF sets.*

RKA: *Suppose* (Ξ^e, Ξ^d) *are RKA-only with* $\Xi^e = \Xi^d = \Phi \times \Gamma$. *If* BC *is* (Ξ^e, Ξ^d)-KC-CCA-*secure then it is* Φ-RK-CCA *secure. If* BC *is* Φ-RK-CCA-*secure and* $\Xi^e = \Xi^d$ *is claw-free then* BC *is* (Ξ^e, Ξ^d)-KC-CCA-*secure.*

In particular, let n *be the block length,* $\mathcal{A}$ *be an adversary and* q *the maximum number of its queries. Then there exists an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\text{rk-cca}}_{\text{BC}}(\Phi, \mathcal{A}) \leq \mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{B}) + q^2/2^n \ .$$

Moreover, there exist adversaries $\mathcal{B}$ *and* $\mathcal{B}^{\text{cf}}$ *such that*

$$\mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{A}) \leq q^2 \cdot \left(\mathbf{Adv}^{\text{cf}}_{\Phi}(\mathcal{B}^{\text{cf}}) + 1/2^n\right) + \mathbf{Adv}^{\text{rk-cca}}_{\text{BC}}(\Phi, \mathcal{B}) \ .$$

KDM: *Suppose* (Ξ^e, Ξ^d) *are KDM-only with* $\Xi^e = \{id\} \times \Psi$ *and* $\Xi^d = \{id\} \times \Gamma$. *If* BC *is* Ψ-KDM-CCA *secure then it is* (Ξ^e, Ξ^d)-KC-CCA-*secure. If* BC *is* (Ξ^e, Ξ^d)-KC-CCA-*secure and* Ξ^e *is claw-free then* BC *is* Ψ-KDM-CCA-*secure.*

In particular, for every adversary $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{A}) = \mathbf{Adv}^{\text{kdm-cca}}_{\text{BC}}(\Psi, \mathcal{B}) \ ,$$

and there exist adversaries $\mathcal{B}$ *and* $\mathcal{B}^{\text{cf}}$ *such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\text{BC}}(\Psi, \mathcal{A}) \leq q^2 \cdot \mathbf{Adv}^{\text{cf}}_{\Psi}(\mathcal{B}^{\text{cf}}) + \mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{B}) \ ,$$

where q *is the maximum number of queries adversary* $\mathcal{A}$ *makes.*

Analogous implications hold in the CPA setting.

4 Key-Correlated Attacks

In this section we show that even if a blockcipher simultaneously achieves security against RK and KDM attacks, it may still fail to achieve security against concurrent RK/KDM attacks, and hence also fail to achieve KCA security. We provide two separations, one artificial and one natural, to demonstrate this.

4.1 A generic separation result

Let $\mathcal{K} = \mathcal{M} = \{0,1\}^n$ and define the following sets of functions:

$$\begin{aligned} \Phi^{\oplus} &:= \left\{K \mapsto K \oplus M : M \in \mathcal{M}\right\} \\ \Psi^{\oplus} &:= \left\{K \mapsto \alpha \cdot K \oplus M : M \in \mathcal{M}, \alpha \in \{0,1\}\right\} \\ \Xi^{\oplus} &:= \left\{K \mapsto (K \oplus M_1, \alpha \cdot K \oplus M_2) : M_1, M_2 \in \mathcal{M}, \alpha \in \{0,1\}\right\} \ . \end{aligned} \tag{2}$$

$\underline{\mathsf{E}'(K,M)}$:	$\underline{\mathsf{D}'(K,C)}$:
$C^* \leftarrow \mathsf{E}(\min(K,\overline{K}),0^n)$	$C^* \leftarrow \mathsf{E}(\min(K,\overline{K}),0^n)$
if $M = K$: return C^*	if $C = C^*$: return K
if $M = \mathsf{D}(K,C^*)$: return $\mathsf{E}(K,K)$	if $C = \mathsf{E}(K,K)$: return $\mathsf{D}(K,C^*)$
return $\mathsf{E}(K,M)$	return $\mathsf{D}(K,C)$

Figure 2: Blockcipher $(\mathsf{E}',\mathsf{D}')$ that is both KDM and RKA secure but not KCA secure. Note that $\min(K,\overline{K}) = \min(\overline{K},\overline{\overline{K}})$, thus $\mathsf{E}'(K,K) = \mathsf{E}'(\overline{K},\overline{K})$.

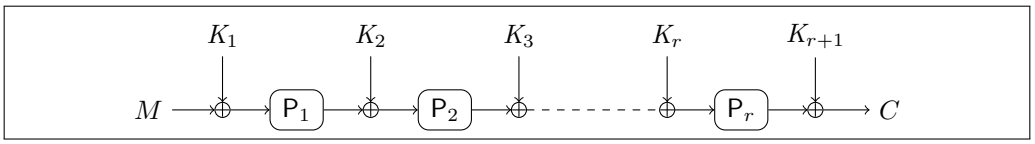


Figure 3: The r-round iterated Even–Mansour cipher.

Given a $\Phi^{\oplus}$-RKA secure and $\Psi^{\oplus}$-KDM secure blockcipher (E,D), consider the modified cipher $(\mathsf{E}',\mathsf{D}')$ shown in Fig. 2, where min is the lexicographic minimum and $\overline{K} := K \oplus 1^k$. Note that $(\mathsf{E}',\mathsf{D}')$ is again a blockcipher, i.e., a permutation for each value of the key (we simply swapped $\mathsf{E}(K,K)$ and C^* in the image space).

To see that $(\mathsf{E}',\mathsf{D}')$ is not $(\Xi^{\oplus},\emptyset)$-KC-CCA secure, consider an adversary that queries $K \mapsto (K,K)$ and $K \mapsto (\overline{K},\overline{K}) = (K \oplus 1^n, K \oplus 1^n)$ to its KCEnc oracle. For the modified cipher both queries yield the same result:

$$\mathsf{E}'(\overline{K},\overline{K}) = \mathsf{E}\big(\min(\overline{K},\overline{\overline{K}}),0^n\big) = \mathsf{E}\big(\min(K,\overline{K}),0^n\big) = \mathsf{E}'(K,K) \ ,$$

while for the ideal cipher this would only happen with probability $1/2^n$. On the other hand, the modified cipher $(\mathsf{E}',\mathsf{D}')$ remains both RKA and KDM secure as it essentially behaves like (E,D) when no joint RKA and KDM attacks as above can be mounted. We kept this discussion informal as our second separating example below is more natural and practically relevant.

4.2 Attack on 2-round Even–Mansour

The r-round Even–Mansour cipher [EM97, DR01, BKL+12] is defined by r permutations $\mathsf{P}_1,\ldots,\mathsf{P}_r$, has key space $\mathcal{K} = \{0,1\}^{(r+1)n}$, domain $\mathcal{M} = \{0,1\}^n$ and encryption and decryption algorithms (cf. Fig. 3)

$$\mathsf{E}\big((K_1,\ldots,K_{r+1}),M\big) := \mathsf{P}_r\big(\ldots\mathsf{P}_2(\mathsf{P}_1(M \oplus K_1) \oplus K_2)\ldots\big) \oplus K_{r+1} \ ,$$
$$\mathsf{D}\big((K_1,\ldots,K_{r+1}),C\big) := \mathsf{P}_1^-\big(\ldots\mathsf{P}_{r-1}^-(\mathsf{P}_r^-(C \oplus K_{r+1}) \oplus K_r)\ldots\big) \oplus K_1 \ .$$

The EM ciphers can also be considered in configurations where (some of the) keys and/or (some of the) permutations are reused in different rounds. We denote the EM cipher where P_i and K_{i+1} are used in round i by $\mathsf{EM}^{\mathsf{P}_1,\ldots,\mathsf{P}_r}[K_1,K_2,\ldots,K_{r+1}]$.

Recall the function sets $\Phi^{\oplus}$, $\Psi^{\oplus}$, and $\Xi^{\oplus}$ from (2). The 2-round Even–Mansour cipher with key reuse and independent permutations $\mathsf{E}(K,M) := \mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2}[K,K,K]$ was shown to be $\Phi^{\oplus}$-RK-CPA secure in [FP15] and $\Psi^{\oplus}$-KDM-CCA secure in [FKV17]. We now show that this construction fails to be $(\Xi^{\oplus},\emptyset)$-KC-CCA secure. Consider the KDM encryption

$$\mathsf{E}(K,K) = \mathsf{P}_2(\mathsf{P}_1(0^n) \oplus K) \oplus K \ . \tag{3}$$

Now let $\Delta^* := \mathsf{P}_1(0^n) \oplus \mathsf{P}_1(1^n)$ and consider the key-correlated encryption

$$\begin{aligned} C_1 := \mathsf{E}(K \oplus \Delta^*, K \oplus \Delta^* \oplus 1^n) &= \mathsf{P}_2(\mathsf{P}_1(1^n) \oplus K \oplus \Delta^*) \oplus K \oplus \Delta^* \\ &= \mathsf{P}_2(\mathsf{P}_1(0^n) \oplus K) \oplus K \oplus \Delta^* \\ &\overset{(3)}{=} \mathsf{E}(K,K) \oplus \Delta^* \ . \end{aligned}$$

Game $\text{KC-CCA}^{\Xi^e,\Xi^d,\mathcal{A}}_{\text{Block}(k,n)}$:	Proc. $\text{KCE}_{\text{NC}}(\xi^e)$:	Proc. $\text{KCD}_{\text{EC}}(\xi^d)$:
$b \twoheadleftarrow \{0,1\}$	$(K,M) \leftarrow \xi^e(K^*)$	$(K,C) \leftarrow \xi^d(K^*)$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \text{Block}(k,n)$	if $(K,M) \in \mathsf{ML}$: return $\perp$	if $(K,C) \in \mathsf{CL}$: return $\perp$
$(\mathsf{iE}',\mathsf{iD}') \twoheadleftarrow \text{Block}(k,n)$	$C \leftarrow \mathsf{iE}(K,M)$	$M \leftarrow \mathsf{iD}(K,C)$
$K^* \twoheadleftarrow \{0,1\}^k$	if $b=0$: $C \leftarrow \mathsf{iE}'(K,M)$	if $b=0$: $M \leftarrow \mathsf{iD}'(K,C)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\text{KCE}_{\text{NC}},\text{KCD}_{\text{EC}}}$	$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$	$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
return $(b'=b)$	return C	return M

Figure 4: Game defining (Ξ^e,Ξ^d)-KC-CCA security of the ideal cipher with key length k and block length n. We require that $\xi^e \in \Xi^e$ and $\xi^d \in \Xi^d$ for all queries ξ^e, ξ^d.

Thus the two key-correlated ciphertexts C_1 and $\mathsf{E}(K,K)$ differ by a known value Δ^*. For the ideal cipher this event only occurs with probability $1/2^n$ since $\Delta^* \neq 0^n$ and the two ciphertexts would be randomly and independently distributed among 2^n values. An attacker thus merely needs to make two queries $K \mapsto (K,K)$ and $K \mapsto (K \oplus \Delta^*, K \oplus \Delta^* \oplus 1^n)$ and check whether the answers differ by Δ^*. The advantage of this attacker is $1 - 1/2^n$.

5 KCA Security of the Ideal Cipher

In this section we study the feasibility of achieving security against key-correlated attacks in the ideal-cipher model. In this setting the adversary has oracle access to the ideal cipher in both the forward and backward direction.

KCA SECURITY OF THE IDEAL CIPHER. We begin by extending the standard notion of CCA security to the KCA setting, where key-dependent messages may be enciphered under related keys. Consider game $\text{KC-CCA}^{\Xi^e,\Xi^d,\mathcal{A}}_{\text{Block}(k,n)}$ shown in Fig. 4. The KC-CCA advantage of an adversary $\mathcal{A}$ against the ideal cipher with key length k and block length n is defined by

$$\mathbf{Adv}^{\text{kc-cca}}_{\text{Block}(k,n)}(\Xi^e,\Xi^d,\mathcal{A}) := 2 \cdot \Pr\left[\text{KC-CCA}^{\Xi^e,\Xi^d,\mathcal{A}}_{\text{Block}(k,n)}\right] - 1 \ .$$

We now prove a feasibility theorem for the ideal cipher that generalizes and extends those of both Bellare and Kohno [BK03] and of Farshim et al. [FKV17]. We define four conditions on CDF sets, which exclude trivial attacks.

UNPREDICTABILITY. The unpredictability advantage of an adversary $\mathcal{A}$ against a CDF set Ξ is defined as

$$\mathbf{Adv}^{\text{up}}_{\Xi}(\mathcal{A}) := \Pr\left[\xi(K) = (K_0,M_0) : (\xi,(K_0,M_0)) \twoheadleftarrow \mathcal{A}; K \twoheadleftarrow \mathcal{K}\right] \ ,$$

where we require that $\xi \in \Xi$. Informally, we say Ξ is unpredictable if the above advantage is "small" for every "reasonable" $\mathcal{A}$.

We also define a multi-shot version of this game where $\mathcal{A}$ outputs a list L_1 of candidates for ξ and a list L_2 of predicted values (K_0,M_0). Adversary $\mathcal{A}$ wins if some ξ in L_1 evaluates to some (K_0,M_0) in L_2 when run on a random key K. Denoting the advantage in this game by $\mathbf{Adv}^{\text{m-up}}_{\Xi}(\mathcal{A})$, a simple guessing argument shows that for any $\mathcal{A}$ there exists $\mathcal{B}$ with

$$\mathbf{Adv}^{\text{m-up}}_{\Xi}(\mathcal{A}) \ \leq \ \ell_1 \cdot \ell_2 \cdot \mathbf{Adv}^{\text{up}}_{\Xi}(\mathcal{B}) \ ,$$

where ℓ_1 and ℓ_2 are upper bounds on the sizes of L_1 and L_2 respectively.[4]

[4]Single-shot adversary $\mathcal{B}$ runs any given multi-shot $\mathcal{A}$ to obtain two lists, picks one entry at random from each list and outputs the pair. Then the success probability of $\mathcal{B}$ is at least $1/(\ell_1\ell_2)$ times that of $\mathcal{A}$.

KEY-UNPREDICTABILITY. The key-unpredictability advantage of an adversary $\mathcal{A}$ against a CDF set Ξ is defined as

$$\mathbf{Adv}^{\text{kup}}_{\Xi}(\mathcal{A}) := \Pr\left[\xi|_1(K) = K_0 : (\xi, K_0) \twoheadleftarrow \mathcal{A}; K \twoheadleftarrow \mathcal{K}\right] ,$$

where we require that $\xi \in \Xi$. Here $\xi|_1(K)$ denotes the projection to the first coordinate of $\xi(K)$. Informally we say Ξ is key-unpredictable if the above advantage is "small" for every "reasonable" $\mathcal{A}$.

We also define a multi-shot version of this game where $\mathcal{A}$ outputs a list L_1 of candidates ξ and a list L_2 of predicted values K_0. Adversary $\mathcal{A}$ wins if for a random key K, for some ξ in L_1 we have $\xi|_1(K) = K_0$ for some K_0 in L_2. If we denote the advantage in this game by $\mathbf{Adv}^{\text{m-kup}}_{\Xi}(\mathcal{A})$, again we have that there exists a single-shot $\mathcal{B}$ such that

$$\mathbf{Adv}^{\text{m-kup}}_{\Xi}(\mathcal{A}) \;\leq\; \ell_1 \cdot \ell_2 \cdot \mathbf{Adv}^{\text{kup}}_{\Xi}(\mathcal{B}) ,$$

where ℓ_1 and ℓ_2 are upper bounds on the sizes of L_1 and L_2 respectively. Note that key-unpredictability implies unpredictability. The converse does not hold as $\xi\colon K \mapsto (0, K)$ is unpredictable but not key-unpredictable.

CLAW-FREENESS. This requires that it is hard to find two distinct functions that output the same on a random input. Formally, the claw-free advantage of an adversary $\mathcal{A}$ against a CDF set Ξ is defined as

$$\mathbf{Adv}^{\text{cf}}_{\Xi}(\mathcal{A}) := \Pr\left[\xi \neq \xi' \wedge \xi(K) = \xi'(K) : (\xi, \xi') \twoheadleftarrow \mathcal{A}; K \twoheadleftarrow \mathcal{K}\right] ,$$

where we require that $\xi, \xi' \in \Xi$. We say Ξ is claw-free if the above advantage is "small" for every "reasonable" $\mathcal{A}$. Once again, a multi-shot version of the game with lists of candidates L_1 and L_2 for ξ and ξ' can be defined, and as before there exists $\mathcal{B}$ with

$$\mathbf{Adv}^{\text{m-cf}}_{\Xi}(\mathcal{A}) \;\leq\; \ell_1 \cdot \ell_2 \cdot \mathbf{Adv}^{\text{cf}}_{\Xi}(\mathcal{B}) .$$

CROSS-KEY-CLAW-DETECTABILITY. We introduce a new notion and call a pair of CDF sets (Ξ^e, Ξ^d) cross-key-claw-detectable (xkcd)[5] if there is an efficient detection algorithm Det such that for any $\mathcal{A}$ the advantage below is small.

$$\mathbf{Adv}^{\text{xkcd}}_{\Xi^e,\Xi^d,\mathsf{Det}}(\mathcal{A}) := \Pr\left[\begin{array}{l} \mathsf{Det}(\xi^e, C, \xi^d, M, 1) \neq \left[(\xi^e|_1(K), C) = \xi^d(K)\right] \;\vee \\ \mathsf{Det}(\xi^e, C, \xi^d, M, 2) \neq \left[(\xi^d|_1(K), M) = \xi^e(K)\right] \end{array} : \begin{array}{l} (\xi^e, C, \xi^d, M) \twoheadleftarrow \mathcal{A}; \\ K \twoheadleftarrow \mathcal{K} \end{array}\right] .$$

We require that $\xi^e \in \Xi^e$ and $\xi^d \in \Xi^d$. Roughly speaking, xkcd provides a way to treat legality of queries across encryption and decryption queries in the KCA game. For example, in an illegal decryption query the output of an earlier encryption query (stored by the game in list CL, cf. Fig. 1) is sent to the decryption oracle through a CDF function whose key component matches that used in encryption. Det allows us to decide when such cases occur and deal with them appropriately in the security analysis.

A generalization of this notion considers lists L_1 for pairs (ξ^e, C) and L_2 for (ξ^d, M) and $\mathcal{A}$ wins if Det fails for any given tuple (ξ^e, C, ξ^d, M) in the product of the lists. Again, a guessing argument shows that for every $\mathcal{A}$ there exists $\mathcal{B}$ with

$$\mathbf{Adv}^{\text{m-xkcd}}_{\Xi^e,\Xi^d,\mathsf{Det}}(\mathcal{A}) \;\leq\; \ell_1 \cdot \ell_2 \cdot \mathbf{Adv}^{\text{xkcd}}_{\Xi^e,\Xi^d,\mathsf{Det}}(\mathcal{B}) .$$

REMARK. We can modify claw-freeness to claw-freeness of keys over the union $\Xi^e \cup \Xi^d$. This would immediately imply xkcd, as Det can always return "No". But in this case the

[5] Please do not confuse this notion with the (popular :) web-comic xkcd.com.

reach of our feasibility result below will not extend to the KDM setting, since under KDM a single key is used in encryption and decryption and hence they always collide.

We will now state and prove our first feasibility result, which identifies a set of sufficient conditions on the pair (Ξ^e, Ξ^d) so that the ideal cipher is KCA secure with respect to it.

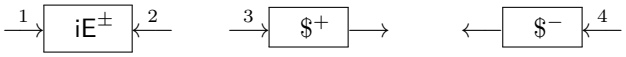
Theorem 1 (KCA security of the ideal cipher)**.** *Fix a key length k and block length n and let Ξ^e and Ξ^d be two sets of CDFs with signature $\xi\colon \{0,1\}^k \to \{0,1\}^k \times \{0,1\}^n$. Then for any (Ξ^e, Ξ^d)-KC-CCA adversary $\mathcal{A}$ against the ideal cipher making at most q direct ideal cipher queries, q_e encryption and q_d decryption queries, and for any detection algorithm* Det*, there are (multi-shot) adversaries $\mathcal{B}_1, \ldots, \mathcal{B}_5$ such that*

$$\mathbf{Adv}^{\text{kc-cca}}_{\text{Block}(k,n)}(\Xi^e, \Xi^d, \mathcal{A}) \le \mathbf{Adv}^{\text{m-kup}}_{\Xi^e}(\mathcal{B}_1) + \mathbf{Adv}^{\text{m-kup}}_{\Xi^d}(\mathcal{B}_2) + \mathbf{Adv}^{\text{m-cf}}_{\Xi^e}(\mathcal{B}_3) + \mathbf{Adv}^{\text{m-cf}}_{\Xi^d}(\mathcal{B}_4) + \mathbf{Adv}^{\text{m-xkcd}}_{\Xi^e,\Xi^d,\mathsf{Det}}(\mathcal{B}_5) + \big(qq_e + qq_d + q_e^2 + q_d^2 + q_e q_d\big)/2^n \ .$$

Algorithm $\mathcal{B}_1$ outputs two lists of sizes at most $\ell_1 \le q_e$ and $\ell_2 \le q$; $\mathcal{B}_2$ outputs lists of sizes at most $\ell_1 \le q_d$ and $\ell_2 \le q$; $\mathcal{B}_3$'s lists are of sizes at most $\ell_1, \ell_2 \le q_e$; $\mathcal{B}_4$'s lists are of sizes at most $\ell_1, \ell_2 \le q_d$, and $\mathcal{B}_5$'s lists of sizes at most $\ell_1 \le q_e$ and $\ell_2 \le q_d$.

Proof. The idea of the proof is that we will gradually modify KC-CCA of Fig. 4 so that KCEnc and KCDec oracles run with two *independent and forgetful random oracles*, denoted by $\$^+$ and $\$^-$ respectively, and further the validity checks are performed by the detection algorithm Det. These modifications ensure that we arrive at a game which does not depend on the explicit key-correlated keys, messages, or ciphertexts.

Pictorially, in this intermediate game with forgetful oracles the adversary can make queries that correspond to 1 (corresponding to iE), 2 (corresponding to iD), 3 and 4:



We now describe the transitional steps; the details can be found in Figs. 5 and 6.

Game_0 : This is the KC-CCA game with respect to $b = 1$ and where the four oracles lazily sample the ideal cipher. The same ideal cipher is used in iE and iD as those used in KCEnc and KCDec oracles. Consistency is ensured via shared lists T^+, T^-, Im^+, and Im^- used in lazy sampling.

Game_1 : This game differs from Game_0 in that different lists are used in iE and iD and in KCEnc and KCDec. The game still ensures consistency across the lists and hence there are no functional changes. This game also sets a flag $\mathsf{Bad}_{\text{kup}}$ preparing us to decouple the two lists. More precisely, whenever a call to iE or iD needs to use the lists of KCEnc and KCDec, or vice versa, $\mathsf{Bad}_{\text{kup}}$ is set. This game also sets a bad flag $\mathsf{Bad}_{\text{col}_1}$ whenever it samples a ciphertext (resp. plaintext) that was sampled before and hence appears on Im_1^+ or Im_2^+ (resp. Im_1^- or Im_2^-).

Game_2 : This game omits the code after $\mathsf{Bad}_{\text{kup}}$ or $\mathsf{Bad}_{\text{col}_1}$ is set. Thus Game_1 and Game_2 are identical until $\mathsf{Bad}_{\text{kup}}$ or $\mathsf{Bad}_{\text{col}_1}$ is set. We defer the analysis of this event to a later game. As a result the two ideal ciphers used in iE and iD and KCEnc and KCDec are decoupled.

Game_3 : This game checks for repeat queries to the ideal cipher used within the KCEnc and KCDec oracles. (Note that the queries to these oracles are assumed, without loss of generality, to be distinct.) The game sets a flag Bad_{cf} when this takes place. This game also sets a bad flag $\mathsf{Bad}_{\text{col}_2}$ whenever it samples a ciphertext (resp. plaintext) that was sampled before in a previous KCEnc or KCDec query (resp., also in a previous KCEnc and KCDec query). By dropping the code after Bad_{cf} and $\mathsf{Bad}_{\text{col}_2}$, we transition to a game where the ideal cipher within KCEnc and KCDec act as forgetful random oracles.

Game_4 : This game uses the detection algorithm Det to check whether or not queries to KCEnc or those to KCDec are valid. More explicitly:

(1) For each decryption query ξ^d, oracle KCDec runs $\mathsf{Det}(\xi^e, C, \xi^d, 0^n, 1)$ for all previous queries ξ^e to KCEnc, which were answered with C. (Intuitively, Det checks whether $\xi^d(K) = (\xi^e|_1(K), C)$; the input 0^n is arbitrary.)

(2) Analogously, for each query ξ^e, oracle KCEnc runs $\mathsf{Det}(\xi^e, 0^n, \xi^d, M, 2)$ for all previous queries ξ^d to KCDec which resulted in output M. If Det (which intuitively checks if $\xi^e(K) = (\xi^d|_1(K), M)$) returns "Yes" for any of these, KCEnc returns $\perp$, else it answers randomly.

Note that this game introduces no functional changes since it still computes the validity of a query using the lists. However it sets a bad flag $\mathsf{Bad}_{\text{xkcd}}$ when the validity check determined by Det is different from the (true) value of validity determined using the lists ML and CL. This then prepares us to drop the explicit check using the lists and only use Det in the next game.

Game_5 : This game drops the code after $\mathsf{Bad}_{\text{xkcd}}$, and so is identical to Game_4 unless $\mathsf{Bad}_{\text{xkcd}}$ has been set.

Note that in Game_5 the iE and iD oracles implement an ideal cipher whereas the KCEnc and KCDec oracles implement two forgetful random oracles, except that sometimes they might return $\perp$ if the Det says that a query is invalid. To signify these, in what follows we refer to KCEnc by \$ and KCDec by $\$^-$. We also use iE^+ for iE and iE^- for iD.

Note also that the above games are all identical unless one of the bad flags is set. Thus we analyze the probability of setting any of the bad flags in the final game Game_5, where the behavior of the oracles are independent of the secret key K^*.

We start by analyzing the probability of setting $\mathsf{Bad}_{\text{kup}}$ and $\mathsf{Bad}_{\text{col}_1}$. Flag $\mathsf{Bad}_{\text{kup}}$ can be set under iE if a query (K, M) to iE appears on T_2^+. Since only KCEnc and KCDec write to T_2^+, this would be the case if (a) $(K, M) = \xi^e(K^*)$ for some query ξ^e to KCEnc or (b) $(K, M) = (\xi^d|_1(K^*), M')$ for M' a randomly chosen output of KCDec.

In case (a) we can build an adversary that wins the multi-shot *unpredictability* of Ξ^e by simulating the oracles as in Game_5 (without the need for K^*, as all dependencies on K^* are eliminated by Game_5) and outputting two lists L_1 and L_2 consisting of all tuples (K, M) that were submitted to iE and ξ^e that were submitted to KCEnc. These lists are of sizes q and q_e respectively.

In case (b), however, we build an adversary that wins the multi-shot *key unpredictability* of Ξ^d (note the exponent d). The four oracles are simulated as in Game_5 (once again without the need for K^*), but now the adversary outputs two lists L_1 and L_2 consisting of all keys K for tuples (K, M) submitted to iE and ξ^e that were submitted to KCDec. These lists are of sizes q and q_d respectively.

We now look at the probability of setting $\mathsf{Bad}_{\text{col}_1}$. A straightforward analysis shows that this probability is upper bounded by $q(q_e + q_d)/2^n$.

The probability of setting $\mathsf{Bad}_{\text{kup}}$ under iD is analyzed similarly to cases (a) and (b) above via reductions to the multi-shot unpredictability of Ξ^d and the multi-shot key unpredictability of Ξ^e. The probability of setting $\mathsf{Bad}_{\text{col}_1}$ under iD is at most $q(q_e+q_d)/2^n$.[6]

Next we consider the probability of setting Bad_{cf} or $\mathsf{Bad}_{\text{col}_2}$. Flag Bad_{cf} can be set under KCEnc as a result of a query ξ_0^e if either (a) a previous distinct query ξ_1^e to KCEnc resulted in $\xi_0^e(K^*) = \xi_1^e(K^*)$, or (b), a previous query ξ_1^d to KCDec resulted in $\xi_0^e(K*) = (\xi_1^d|_1(K*), M)$, where M is a random value chosen by KCDec in a previous query.

[6] The above cases take care of collisions between wires $(1/2, 3)$, $(1/2, 4)$ in the picture above.

$\underline{\mathsf{Game}_0^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\mathrm{KCE}_{\mathrm{NC}},\mathrm{KCD}_{\mathrm{EC}}}$
return $(b'=1)$

$\underline{\text{Proc. } \mathsf{iE}(K,M)}$:
if $\mathsf{T}^+[K,M]$: return $\mathsf{T}^+[K,M]$
else $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}^+[K]$
$\mathsf{T}^+[K,M] := C$
$\mathsf{T}^-[K,C] := M$
$\mathsf{Im}^+[K] \leftarrow \mathsf{Im}^+[K] : C$
$\mathsf{Im}^-[K] \leftarrow \mathsf{Im}^-[K] : M$
return C

$\underline{\text{Proc. } \mathsf{iD}(K,C)}$:
if $\mathsf{T}^-[K,C]$: return $\mathsf{T}^-[K,C]$
else $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}^-[K]$
$\mathsf{T}^-[K,C] := M$
$\mathsf{T}^+[K,M] := C$
$\mathsf{Im}^-[K] \leftarrow \mathsf{Im}^-[K] : M$
$\mathsf{Im}^+[K] \leftarrow \mathsf{Im}^+[K] : C$
return M

$\underline{\text{Proc. } \mathrm{KCE}_{\mathrm{NC}}(\xi^e)}$:
$(K,M) \leftarrow \xi^e(K^*)$
if $(K,M) \in \mathsf{ML}$: return $\bot$
if $\mathsf{T}^+[K,M]$: return $\mathsf{T}^+[K,M]$
else $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}^+[K]$
$\mathsf{T}^+[K,M] := C$
$\mathsf{T}^-[K,C] := M$
$\mathsf{Im}^+[K] \leftarrow \mathsf{Im}^+[K] : C$
$\mathsf{Im}^-[K] \leftarrow \mathsf{Im}^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
return C

$\underline{\text{Proc. } \mathrm{KCD}_{\mathrm{EC}}(\xi^d)}$:
$(K,C) \leftarrow \xi^d(K^*)$
if $(K,C) \in \mathsf{CL}$: return $\bot$
if $\mathsf{T}^-[K,C]$: return $\mathsf{T}^-[K,C]$
else $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}^-[K]$
$\mathsf{T}^-[K,C] := M$
$\mathsf{T}^+[K,M] := C$
$\mathsf{Im}^-[K] \leftarrow \mathsf{Im}^-[K] : M$
$\mathsf{Im}^+[K] \leftarrow \mathsf{Im}^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
return M

$\underline{\mathsf{Game}_1^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\mathrm{KCE}_{\mathrm{NC}},\mathrm{KCD}_{\mathrm{EC}}}$
return $(b'=1)$

$\underline{\text{Proc. } \mathsf{iE}(K,M)}$:
if $\mathsf{T}_1^+[K,M]$: return $\mathsf{T}_1^+[K,M]$
if $\mathsf{T}_2^+[K,M]$: $\boxed{\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top}$
  return $\mathsf{T}_2^+[K,M]$
else $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^+[K]$
if $C \in \mathsf{Im}_2^+[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top}$
  $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^+[K] \cup \mathsf{Im}_2^+[K]$
$\mathsf{T}_1^+[K,M] := C$
$\mathsf{T}_1^-[K,C] := M$
$\mathsf{Im}_1^+[K] \leftarrow \mathsf{Im}_1^+[K] : C$
$\mathsf{Im}_1^-[K] \leftarrow \mathsf{Im}_1^-[K] : M$
return C

$\underline{\text{Proc. } \mathsf{iD}(K,C)}$:
if $\mathsf{T}_1^-[K,C]$: return $\mathsf{T}_1^-[K,C]$
if $\mathsf{T}_2^-[K,C] \neq \bot$: $\boxed{\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top}$
  return $\mathsf{T}_2^-[K,C]$
else $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^-[K]$
if $M \in \mathsf{Im}_2^-[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top}$
  $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^-[K] \cup \mathsf{Im}_2^-[K]$
$\mathsf{T}_1^-[K,C] := M$
$\mathsf{T}_1^+[K,M] := C$
$\mathsf{Im}_1^-[K] \leftarrow \mathsf{Im}_1^-[K] : M$
$\mathsf{Im}_1^+[K] \leftarrow \mathsf{Im}_1^+[K] : C$
return M

$\underline{\text{Proc. } \mathrm{KCE}_{\mathrm{NC}}(\xi^e)}$:
$(K,M) \leftarrow \xi^e(K^*)$
if $(K,M) \in \mathsf{ML}$: return $\bot$
if $\mathsf{T}_2^+[K,M]$: return $\mathsf{T}_2^+[K,M]$
if $\mathsf{T}_1^+[K,M]$: $\boxed{\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top}$
  return $\mathsf{T}_1^+[K,M]$
else $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K]$
if $C \in \mathsf{Im}_1^+[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top}$
  $C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K] \cup \mathsf{Im}_1^+[K]$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
return C

$\underline{\text{Proc. } \mathrm{KCD}_{\mathrm{EC}}(\xi^d)}$:
$(K,C) \leftarrow \xi^d(K^*)$
if $(K,C) \in \mathsf{CL}$: return $\bot$
if $\mathsf{T}_2^-[K,C]$: return $\mathsf{T}_2^-[K,C]$
if $\mathsf{T}_1^-[K,C]$: $\boxed{\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top}$
  return $\mathsf{T}_1^-[K,C]$
else $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K]$
if $M \in \mathsf{Im}_1^-[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top}$
  $M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K] \cup \mathsf{Im}_1^-[K]$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
return M

$\underline{\mathsf{Game}_2^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\mathrm{KCE}_{\mathrm{NC}},\mathrm{KCD}_{\mathrm{EC}}}$
return $(b'=1)$

$\underline{\text{Proc. } \mathsf{iE}(K,M)}$:
if $\mathsf{T}_1^+[K,M]$: return $\mathsf{T}_1^+[K,M]$
if $\mathsf{T}_2^+[K,M]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
$\boxed{\text{// return } \mathsf{T}_2^+[K,M]}$
$C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^+[K]$
if $C \in \mathsf{Im}_2^+[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
$\boxed{\text{// } C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^+[K] \cup \mathsf{Im}_2^+[K]}$
$\mathsf{T}_1^+[K,M] := C$
$\mathsf{T}_1^-[K,C] := M$
$\mathsf{Im}_1^+[K] \leftarrow \mathsf{Im}_1^+[K] : C$
$\mathsf{Im}_1^-[K] \leftarrow \mathsf{Im}_1^-[K] : M$
return C

$\underline{\text{Proc. } \mathsf{iD}(K,C)}$:
if $\mathsf{T}_1^-[K,C]$: return $\mathsf{T}_1^-[K,C]$
if $\mathsf{T}_2^-[K,C] \neq \bot$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
$\boxed{\text{// return } \mathsf{T}_2^-[K,C]}$
$M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^-[K]$
if $M \in \mathsf{Im}_2^-[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
$\boxed{\text{// } M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_1^-[K] \cup \mathsf{Im}_2^-[K]}$
$\mathsf{T}_1^-[K,C] := M$
$\mathsf{T}_1^+[K,M] := C$
$\mathsf{Im}_1^-[K] \leftarrow \mathsf{Im}_1^-[K] : M$
$\mathsf{Im}_1^+[K] \leftarrow \mathsf{Im}_1^+[K] : C$
return M

$\underline{\text{Proc. } \mathrm{KCE}_{\mathrm{NC}}(\xi^e)}$:
$(K,M) \leftarrow \xi^e(K^*)$
if $(K,M) \in \mathsf{ML}$: return $\bot$
if $\mathsf{T}_2^+[K,M]$: return $\mathsf{T}_2^+[K,M]$
if $\mathsf{T}_1^+[K,M]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
$\boxed{\text{// return } \mathsf{T}_1^+[K,M]}$
$C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K]$
if $C \in \mathsf{Im}_1^+[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
$\boxed{\text{// } C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K] \cup \mathsf{Im}_1^+[K]}$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
return C

$\underline{\text{Proc. } \mathrm{KCD}_{\mathrm{EC}}(\xi^d)}$:
$(K,C) \leftarrow \xi^d(K^*)$
if $(K,C) \in \mathsf{CL}$: return $\bot$
if $\mathsf{T}_2^-[K,C]$: return $\mathsf{T}_2^-[K,C]$
if $\mathsf{T}_1^-[K,C]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
$\boxed{\text{// return } \mathsf{T}_1^-[K,C]}$
$M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K]$
if $M \in \mathsf{Im}_1^-[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
$\boxed{\text{// } M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K] \cup \mathsf{Im}_1^-[K]}$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
return M

Figure 5: Game_0, Game_1, and Game_2.

$\underline{\mathsf{Game}_3^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\text{KCEnc},\text{KCDec}}$
return $(b'=1)$

Proc. $\mathsf{iE}(K,M)$:
Unmodified
Proc. $\mathsf{iD}(K,C)$:
Unmodified

Proc. KCEnc(ξ^e):
$(K,M) \leftarrow \xi^e(K^*)$
if $(K,M) \in \mathsf{ML}$: return $\bot$
if $\mathsf{T}_2^+[K,M]$: $\boxed{\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top}$
$\boxed{/\!/ \text{ return } \mathsf{T}_2^+[K,M]}$
if $\mathsf{T}_1^+[K,M]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^+[K,M]$
$C \twoheadleftarrow \{0,1\}^n$
if $C \in \mathsf{Im}_2^+[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top}$
$\boxed{/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K]}$
if $C \in \mathsf{Im}_1^+[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K] \cup \mathsf{Im}_1^+[K]$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
return C

Proc. KCDec(ξ^d):
$(K,C) \leftarrow \xi^d(K^*)$
if $(K,C) \in \mathsf{CL}$: return $\bot$
if $\mathsf{T}_2^-[K,C]$: $\boxed{\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top}$
$\boxed{/\!/ \text{ return } \mathsf{T}_2^-[K,C]}$
if $\mathsf{T}_1^-[K,C]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^-[K,C]$
$M \twoheadleftarrow \{0,1\}^n$
if $M \in \mathsf{Im}_2^-[K]$: $\boxed{\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top}$
$\boxed{/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K]}$
if $M \in \mathsf{Im}_1^-[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K] \cup \mathsf{Im}_1^-[K]$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
return M

$\underline{\mathsf{Game}_4^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\text{KCEnc},\text{KCDec}}$
return $(b'=1)$

Proc. $\mathsf{iE}(K,M)$:
Unmodified
Proc. $\mathsf{iD}(K,C)$:
Unmodified

Proc. KCEnc(ξ^e):
$(K,M) \leftarrow \xi^e(K^*)$
$\boxed{\begin{array}{l}\mathsf{chk} \leftarrow \bot\\ \text{for } (\xi^d,M) \in \mathsf{DL}:\\ \quad \mathsf{chk} \leftarrow \mathsf{chk} \vee \mathsf{Det}(\xi^e,0^n,\xi^d,M,2)\\ \text{if } \mathsf{chk} \neq ((K,M) \in \mathsf{ML}):\\ \quad \mathsf{Bad}_{\mathrm{xkcd}} \leftarrow \top\\ \quad \mathsf{chk} \leftarrow ((K,M) \in \mathsf{ML})\\ \text{if chk: return } \bot\end{array}}$
if $\mathsf{T}_2^+[K,M]$: $\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_2^+[K,M]$
if $\mathsf{T}_1^+[K,M]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^+[K,M]$
$C \twoheadleftarrow \{0,1\}^n$
if $C \in \mathsf{Im}_2^+[K]$: $\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top$
  $/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K]$
if $C \in \mathsf{Im}_1^+[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K] \cup \mathsf{Im}_1^+[K]$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
$\mathsf{EL} \leftarrow \mathsf{EL} : (\xi^e,C)$
return C

Proc. KCDec(ξ^d):
$(K,C) \leftarrow \xi^d(K^*)$
$\boxed{\begin{array}{l}\mathsf{chk} \leftarrow \bot\\ \text{for } (\xi^e,C) \in \mathsf{EL}:\\ \quad \mathsf{chk} \leftarrow \mathsf{chk} \vee \mathsf{Det}(\xi^e,C,\xi^d,0^n,1)\\ \text{if } \mathsf{chk} \neq ((K,C) \in \mathsf{CL}):\\ \quad \mathsf{Bad}_{\mathrm{xkcd}} \leftarrow \top\\ \quad \mathsf{chk} \leftarrow ((K,C) \in \mathsf{CL})\\ \text{if chk: return } \bot\end{array}}$
if $\mathsf{T}_2^-[K,C]$: $\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_2^-[K,C]$
if $\mathsf{T}_1^-[K,C]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^-[K,C]$
$M \twoheadleftarrow \{0,1\}^n$
if $M \in \mathsf{Im}_2^-[K]$: $\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top$
  $/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K]$
if $M \in \mathsf{Im}_1^-[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K] \cup \mathsf{Im}_1^-[K]$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
$\mathsf{DL} \leftarrow \mathsf{DL} : (\xi^d,M)$
return M

$\underline{\mathsf{Game}_5^{\Xi^e,\Xi^d,\mathcal{A}}}$:
$b \leftarrow 1$; $K^* \twoheadleftarrow \{0,1\}^k$
$(\mathsf{iE},\mathsf{iD}) \twoheadleftarrow \mathrm{Block}(k,n)$
$b' \twoheadleftarrow \mathcal{A}^{\mathsf{iE},\mathsf{iD},\text{KCEnc},\text{KCDec}}$
return $(b'=1)$

Proc. $\mathsf{iE}(K,M)$:
Unmodified
Proc. $\mathsf{iD}(K,C)$:
Unmodified

Proc. KCEnc(ξ^e):
$(K,M) \leftarrow \xi^e(K^*)$
$\mathsf{chk} \leftarrow \bot$
for $(\xi^d,M) \in \mathsf{DL}$:
  $\mathsf{chk} \leftarrow \mathsf{chk} \vee \mathsf{Det}(\xi^e,0^n,\xi^d,M,2)$
if $\mathsf{chk} \neq ((K,M) \in \mathsf{ML})$:
  $\mathsf{Bad}_{\mathrm{xkcd}} \leftarrow \top$
$\boxed{/\!/ \ \mathsf{chk} \leftarrow ((K,M) \in \mathsf{ML})}$
if chk: return $\bot$
if $\mathsf{T}_2^+[K,M]$: $\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_2^+[K,M]$
if $\mathsf{T}_1^+[K,M]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^+[K,M]$
$C \twoheadleftarrow \{0,1\}^n$
if $C \in \mathsf{Im}_2^+[K]$: $\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top$
  $/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K]$
if $C \in \mathsf{Im}_1^+[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ C \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^+[K] \cup \mathsf{Im}_1^+[K]$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K,C)$
$\mathsf{EL} \leftarrow \mathsf{EL} : (\xi^e,C)$
return C

Proc. KCDec(ξ^d):
$(K,C) \leftarrow \xi^d(K^*)$
$\mathsf{chk} \leftarrow \bot$
for $(\xi^e,C) \in \mathsf{EL}$:
  $\mathsf{chk} \leftarrow \mathsf{chk} \vee \mathsf{Det}(\xi^e,C,\xi^d,0^n,1)$
if $\mathsf{chk} \neq ((K,C) \in \mathsf{CL})$:
  $\mathsf{Bad}_{\mathrm{xkcd}} \leftarrow \top$
$\boxed{/\!/ \ \mathsf{chk} \leftarrow ((K,C) \in \mathsf{CL})}$
if chk: return $\bot$
if $\mathsf{T}_2^-[K,C]$: $\mathsf{Bad}_{\mathrm{cf}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_2^-[K,C]$
if $\mathsf{T}_1^-[K,C]$: $\mathsf{Bad}_{\mathrm{kup}} \leftarrow \top$
  $/\!/$ return $\mathsf{T}_1^-[K,C]$
$M \twoheadleftarrow \{0,1\}^n$
if $M \in \mathsf{Im}_2^-[K]$: $\mathsf{Bad}_{\mathrm{col}_2} \leftarrow \top$
  $/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K]$
if $M \in \mathsf{Im}_1^-[K]$: $\mathsf{Bad}_{\mathrm{col}_1} \leftarrow \top$
  $/\!/ \ M \twoheadleftarrow \{0,1\}^n \setminus \mathsf{Im}_2^-[K] \cup \mathsf{Im}_1^-[K]$
$\mathsf{T}_2^-[K,C] := M$
$\mathsf{T}_2^+[K,M] := C$
$\mathsf{Im}_2^-[K] \leftarrow \mathsf{Im}_2^-[K] : M$
$\mathsf{Im}_2^+[K] \leftarrow \mathsf{Im}_2^+[K] : C$
$\mathsf{ML} \leftarrow \mathsf{ML} : (K,M)$
$\mathsf{DL} \leftarrow \mathsf{DL} : (\xi^d,M)$
return M

Figure 6: Game_3, Game_4, and Game_5.

In case (a) by collecting the list of ξ^e queried to KCEnc we can break the multi-shot claw-freeness of Ξ^e (simulating the oracles as in Game_5). For case (b) to take place it must be that Det classified the query as valid. However, this leads to setting $\mathsf{Bad}_{\text{xkcd}}$ since M was returned from the KCDec oracle and hence $(\xi_1^d|_1(K*), M)$ appears on ML but Det did not detect this. We analyze the probability of setting $\mathsf{Bad}_{\text{xkcd}}$ below. The probability of setting $\mathsf{Bad}_{\text{col}_2}$ in KCEnc can be upper bounded by $q_e(q_e + q_d)/2^n$.

The probability of setting Bad_{cf} under KCDec is analyzed as above by a reduction to the multi-shot claw-freeness of Ξ^d (or setting $\mathsf{Bad}_{\text{xkcd}}$). The probability of setting $\mathsf{Bad}_{\text{col}_2}$ in KCDec is upper bounded by $q_d(q_e + q_d)/2^n$.

Finally we consider the probability of setting $\mathsf{Bad}_{\text{xkcd}}$. Flag $\mathsf{Bad}_{\text{xkcd}}$ is set exactly when $\xi^e(K^*)$ appears on ML as a result of a query ξ^d to KCDec with response M and Det says it does not, or that it does not and Det says it does. We can bound the probability of setting $\mathsf{Bad}_{\text{xkcd}}$ under KCEnc by a reduction to the multi-shot xkcd property by collecting all queries ξ^e made to KCEnc and all query-answer pairs (ξ^d, M) made to and received form KCDec in two lists and outputting them lists in the multi-shot xkcd game. The probability of setting of $\mathsf{Bad}_{\text{xkcd}}$ under KCDec is analyzed analogously.

The theorem now follows by putting the bounds established above together. □

The above theorem shows that key-unpredictability, claw-freeness, and xkcd are sufficient for achieving KCA security in the ideal-cipher model. This raises the question whether or not they are also *necessary*. Key-unpredictability is. Any adversary that can predict the key output of ξ^e can be used to win the KC game as follows: (1) ask for an encryption under $\xi^e(K^*)$ to get a ciphertext C; (2) predict $\xi^e|_1(K^*)$ as K; (3) compute $\mathsf{iD}(K, C)$ to get M; (4) compute $\mathsf{iE}(K, M)$ to get C'; (5) return $b' := 1$ if $C = C'$ and $b' := 0$ otherwise. It is easily seen that this attack returns $b' = 1$ with the same probability of predicting keys when $b = 1$. On the other hand, it only succeeds with negligible probability when $b' = 0$ as the encryption oracle and the challenge oracle use independent ideal ciphers. We deal with such attacks in case (b) of $(\$^+, \mathsf{iE}^{\pm})$ in the proof.

Claw-freeness, on the other hand, is not necessary. Consider a CDF set consisting of exactly two functions that collide with probability 1/2, e.g., $K \mapsto K$ and $K \mapsto \min(K, \overline{K})$. Then a security proof for the ideal cipher can be easily given: guess with probability 1/2 if $\min(K, \overline{K}) = K$. Next depending on the guess simulate the key-correlated oracles either using the same set of ideal cipher oracles or independent ones. However, one can also exhibit CDF sets that are not claw-free and *do* lead to a KCA on the ideal cipher. Indeed, any set of $k + 1$ functions that has claws with the identity function depending on whether or not K_i, the i-th bit of the key, is 1 can be used to recover the key one bit at a time. In practice this translates to recovering the key by observing the repetition patterns in the outputs.

Finally, there is also a pair of CDF sets which are not xkcd and can be used to attack the ideal cipher. Let $\xi^e(K) := (K, 0^n)$ and $\xi^d[i, C](K) := (K \oplus (K_i)^n =: K', C)$, that is, $K' = \overline{K}$ if $K_i = 1$ and $K' = K$ otherwise. Let C be the answer to $\text{KCEnc}(\xi^e)$. Now for $i = 1, \ldots, k$, by checking whether $\text{KCDec}(\xi^d[i, C])$ returns $\perp$ or not, one can again recover the key one bit at a time. In practice this translates to observing the equality pattern of the decrypted ciphertexts with 0^n.

Theorem 1 and Proposition 1 can be used to recover known feasibility results for RKA and KDM security of the ideal cipher as follows.

RKA security. An RKA-only CDF set Ξ associated to a set Φ is key-unpredictable iff Φ is unpredictable in the sense of [BK03]. Indeed, a predictor for a related-key derivation function ϕ can be converted to a key-predictor for $\xi\colon K \mapsto (\phi(K), M)$ for any fixed M. Conversely, a key-predictor for $(\phi(K), M)$ in particular predicts $\phi(K)$. Similarly, Ξ is claw-free iff Φ is claw-free in the sense of [BK03]. A claw in Φ can be immediately converted

to a claw in Ξ (by adding any message M). Conversely, a claw $((\phi_1(K), M_1), (\phi_2(K), M_2))$ is necessarily also a claw between ϕ_1 and ϕ_2. For the (funny) xkcd property, Det given $((\phi_1, M_1), C, (\phi_2, C_2), M, i)$ needs to check if (1) $\phi_1(K) = \phi_2(K)$ and (2) if $C = C_2$ when $i = 1$ or if $M = M_1$ when $i = 2$. By claw-freeness, (1) would only occur if $\phi_1 = \phi_2$; the second condition on the other hand is trivial to check. Thus, claw-freeness implies xkcd.

KDM SECURITY. A KDM-only Ξ associated to a KDM set Ψ is always key-unpredictable with single-shot advantage $1/2^n$. This is because KDM-only functions leave the random key unmodified, which remains information-theoretically hidden from the adversary in the key-unpredictability game. Moreover, and similarly to the RKA case, the set Ξ is claw-free iff Ψ is. Once again, this is because KDM-only CDFs do not modify the key component and hence any claw must be on the message part. For the xckd property, given $((id, \psi_1), C, (id, C_2), M, i)$, algorithm Det needs to check (1) if $id(K) = id(K)$, which always holds; and (2) if $C_2 = C$ when $i = 1$ or if $\psi_1(K) = M$ when $i = 2$. The first of these is trivially checkable. For the second, two cases could occur: either $\psi_1(K)$ is a constant function mapping to M_1, in which case Det can easily check if $M_1 = M$; or ψ_1 is non-constant. However, since Ψ contains all constant functions, ψ_1 and M can be used to break claw-freeness. Hence in this case Det rejects. So similarly to the RKA setting, claw-freeness implies xckd.

KCA SECURITY. We now show that the reach of the above feasibility result for the ideal cipher extends beyond RKA and KDM security. Suppose $k = n$ and recall the set

$$\Xi^{\oplus} := \left\{ K \mapsto (K \oplus \Delta_1, \alpha \cdot K \oplus \Delta_2) : \Delta_1, \Delta_2 \in \mathcal{M}, \alpha \in \{0, 1\} \right\} . \tag{4}$$

This set captures related keys that are computed as offsets, key-*in*dependent messages when $\alpha = 0$, and key-dependent messages which are offsets of the key when $\alpha = 1$. We identify functions in this set with tuples $(\Delta_1, \Delta_2, \alpha)$.

This set is key-unpredictable with single-shot advantage at most $1/2^n$, since xor-ing with Δ_1 acts as a permutation and K remains information-theoretically hidden. It is also claw-free with single-shot advantage at most $1/2^n$: If the offsets Δ_1 and Δ_1' for two functions ξ, ξ' are different, there are no claws. If they are the same, and $\alpha = \alpha'$, then a claw also implies $\Delta_2 = \Delta_2'$, making the two functions identical. So $\alpha \neq \alpha'$ is necessary for claws, in which case the functions collide on a random key with probability $1/2^n$.

The set is also xkcd with single-shot advantage $2/2^n$ with respect to the following detector (recall that $\mathsf{Det}(\xi^e, C, \xi^d, M, 1)$ should check whether $(\xi^e|_1(K), C) = \xi^d(K)$):

$$\begin{aligned} &\mathsf{Det}_{\oplus}\big((\Delta_1^e, \Delta_2^e, \alpha^e), C, (\Delta_1^d, \Delta_2^d, \alpha^d), M, i\big) : \\ &\qquad \text{if } i = 1 : \ \text{if } \Delta_1^e = \Delta_1^d \wedge \alpha^d = 0 \wedge \Delta_2^d = C \text{ then return “Yes”} \\ &\qquad \text{if } i = 2 : \ \text{if } \Delta_1^e = \Delta_1^d \wedge \alpha^e = 0 \wedge \Delta_2^e = M \text{ then return “Yes”} \\ &\qquad \text{else return “No”} \end{aligned} \tag{5}$$

Consider case $i = 1$ when $\mathsf{Det}_{\oplus}$ is trying to decide if $(K \oplus \Delta_1^e, C) = (K \oplus \Delta_1^d, \alpha^d \cdot K \oplus \Delta_2^d)$. If $\mathsf{Det}_{\oplus}$ returns “Yes”, then both the key components and the ciphertext components collide; the answer is thus correct. Suppose it returns “No”. If $\Delta_1^e \neq \Delta_1^d$, the key components do not collide and the answer is correct. Suppose $\Delta_1^e = \Delta_1^d$: if $\alpha^d = 1$, the probability that $\alpha^d \cdot K \oplus \Delta_2^d = C$ is $1/2^n$ since $\oplus$ acts a permutation and K is random; if $\alpha^d = 0$, then $\mathsf{Det}_{\oplus}$ checks if $\Delta_2^d = C$ and hence its answer is correct. Case $i = 2$ is dealt with similarly and we obtain overall single-shot advantage of at most $2/2^n$.

Theorem 1 thus implies that the ideal cipher is $(\Xi^{\oplus}, \Xi^{\oplus})$-KC-CCA secure.

6 KCA Security of 3-Round Even–Mansour

In this section we show that in contrast to the two-round case the three-round Even–Mansour cipher with *reuse of keys and independent permutations* $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2,\mathsf{P}_3}[K, K, K, K]$,

defined in Sec. 4.2, is $(\Xi^{\oplus}, \Xi^{\oplus})$-KC-CCA secure, with $\Xi^{\oplus}$ as defined in (4). We note that, for $\Phi^{\oplus}, \Psi^{\oplus}$ as in (2), this construction is known to be both $\Phi^{\oplus}$-RK-CCA and $\Psi^{\oplus}$-KDM-CCA secure [FP15, FKV17]. We build on these works to show KC-CCA security in the theorem.

Theorem 2 (KCA security of 3-round EM)**.** $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2,\mathsf{P}_3}[K,K,K,K]$ *is* $(\Xi^{\oplus}, \Xi^{\oplus})$-KC-CCA *secure in the random-permutation model for* P_i*. More precisely, for any adversary* $\mathcal{A}$ *against the* $(\Xi^{\oplus}, \Xi^{\oplus})$-KC-CCA *security of* $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2,\mathsf{P}_3}[K,K,K,K]$ *we have*

$$\mathbf{Adv}^{\text{kc-cca}}_{\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2,\mathsf{P}_3}[K,K,K,K]}(\Xi^{\oplus}, \Xi^{\oplus}, \mathcal{A}) \leq 2 \cdot \big(qq_e + qq_d + q_e^2 + q_d^2 + q_e q_d\big)/(2^n - q) \ ,$$

where q *is the maximum number of queries of* $\mathcal{A}$ *to* $\mathsf{P}_i^{\pm}$ *(over all* $i = 1, 2, 3$*) and* q_e *and* q_d *are the maximum number of encryption and decryption queries of* $\mathcal{A}$*, respectively.*

Proof. To analyze the KC-CCA security of Even–Mansour, our strategy will be similar to that in Theorem 1 and those in [FP15, FKV17]. As in these proofs, we replace the last-round permutation P_3^+ in KCEnc queries with a forgetful random oracle and also replace the first-round permutation P_1^- in KCDec with another forgetful random oracle. After these replacements the outputs of KCEnc and KCDec will be fully randomized and independent of the inputs; the game is thus independent of the challenge bit b.

When simulating KCEnc and KCDec with forgetful oracles, we still need to ensure that illegal queries are answered correctly with $\perp$. As in the proof of Theorem 1, we do this using a detector. In particular, we use $\mathsf{Det}_{\oplus}$ defined in (5), which on input $((\Delta_1, \Delta_2, \alpha), C, (\Delta_1', \Delta_2', \alpha'), M, 1)$ returns "Yes" iff $\Delta_1 = \Delta_1'$ and $\alpha' = 0$ and $\Delta_2' = C$ (and "No" otherwise); and on input $((\Delta_1, \Delta_2, \alpha), C, (\Delta_1', \Delta_2', \alpha'), M, 2)$ returns "Yes" iff $\Delta_1 = \Delta_1'$ and $\alpha = 0$ and $\Delta_2 = M$.

(1) For each decryption query ξ^d, oracle KCDec runs $\mathsf{Det}_{\oplus}(\xi^e, C, \xi^d, 0^n, 1)$ for all previous queries ξ^e to KCEnc, which were answered with C. If Det says "Yes" for any of these, $\text{KCDec}_{\oplus}$ returns $\perp$, else it outputs a random value from the domain.

(2) For each encryption query ξ^e, oracle KCEnc runs $\mathsf{Det}_{\oplus}(\xi^e, 0^n, \xi^d, M, 2)$ for all previous queries ξ^d to KCDec, which were answered with M. If $\mathsf{Det}_{\oplus}$ says "Yes" for any of these, KCEnc returns $\perp$, else it answers with a random value.

These replacements, however, raise the question how P_1^+ in KCEnc and P_3^- in KCDec are treated (since their respective counterparts in KCDec and KCEnc were replaced). Previous works simply leave these oracles unchanged. Our proof diverges in this aspect, because the key-dependency of *both* keys and messages gives rise to queries that cannot be guaranteed to be disjoint from the forgetful oracle, or discarded as trivial/illegal as in the KDM setting [FKV17], or not result in inconsistency due to the middle permutation P_2 being replaced in the RKA setting [FP15].

For the KCA setting we will replace P_3^- in KCDec (and analogously for P_1^+ in KCEnc) as follows. When $\mathsf{Det}_{\oplus}$ says "No" on a query $\xi^d = (\Delta_1^d, \Delta_2^d, \alpha^d)$, the game checks if each previous query $\xi^e = (\Delta_1^e, \Delta_2^e, \alpha^e)$ to KCEnc with output value C satisfies

$$C \oplus \Delta_1^e = \Delta_2^d \oplus \Delta_1^d \ .$$

If so, oracle $\tilde{\mathsf{P}}_3^-$ outputs the input U to $\$^+$ that was used to compute C. If multiple values are found, $\tilde{\mathsf{P}}_3^-$ returns $\perp$. If the latter is not the case, $\tilde{\mathsf{P}}_3^-$ is sampled as an independent permutation and consistent with P_3. Permutation $\mathsf{P}_3^{\pm}$, when directly queried, is also sampled consistently with $\tilde{\mathsf{P}}_3^-$ on all those entries that are not kept consistent with $\$^+$. More precisely, oracles $\$^+$, $\mathsf{P}_3^{\pm}$ and $\tilde{\mathsf{P}}_3^-$ write to independent lists L_1, L_2 and L_3. Oracle $\$^+$ checks consistency with none of the lists, oracle $\mathsf{P}_3^{\pm}$ checks consistency with L_2 and L_3 and oracle $\tilde{\mathsf{P}}_3^-$ checks consistency with L_1, L_2 and L_3.

Oracle P_1^+ in KCEnc is replaced with $\tilde{\mathsf{P}}_1^+$ similarly by checking if $M \oplus \Delta_1^d = \Delta_2^e \oplus \Delta_1^e$ and outputting the input U to $\$^-$ that was used to compute M.

As a result of the above replacements, the following inconsistencies could arise:

(a) Identical queries to a permutation oracle are answered differently due to a replacement.

(b) Different queries to a permutation oracle and its replacement are answered identically (and hence do not respect permutativity).

(c) Multiple candidates for U are found when sampling $\tilde{\mathsf{P}}_3^-$ or $\tilde{\mathsf{P}}_1^+$.

We now bound the probabilities of these events.

Multiple candidates This event will happen due to: (1) a collision in the outputs of $\$^+$, which happens with probability $1/2^n$ for each pair of queries to KCEnc; or (2) a collision in the output of $\$^+$ and an output entry in $\mathsf{P}_3^{\pm}$. This event is equivalent to a collision in the outputs of $\$^+$ and $\mathsf{P}_3^{\pm}$, which pertains to case (b) and is analyzed next.

($\$^+, \mathsf{P}_3^{\pm}$) A direct $\mathsf{P}_3^{\pm}$ call in either direction collides with a point queried to P_3^+ due to a query $\xi^e = (\Delta_1, \Delta_2, \alpha)$ to KCEnc. Suppose the direct query results in $V = \mathsf{P}_3^+(U)$ or $U = \mathsf{P}_3^-(V)$. There are now two cases. (a) The inputs collide, i.e.,

$$\mathsf{P}_2\big(\mathsf{P}_1(\Delta_1 \oplus \Delta_2 \oplus (1-\alpha)\cdot K^*) \oplus \Delta_1 \oplus K^*\big) \oplus \Delta_1 \oplus K^* = U \ . \tag{6}$$

If $Y := \mathsf{P}_1(\Delta_1 \oplus \Delta_2 \oplus (1-\alpha)\cdot K^*) \oplus \Delta_1 \oplus K^*$ was queried to P_2 then the adversary can compute $K^* = \mathsf{P}_2(Y) \oplus \Delta_1 \oplus U$, which is only possible with probability $1/2^n$, since the key remains information-theoretically hidden. If it was not queried, the probability that (6) holds is also $1/(2^n - q)$, since P_2 is a random permutation, where outputs are chosen in a set of size at least $2^n - q$.
The other case is that (b) the outputs collide. If the $\$^+$ query comes after the direct query, the probability of this event is $1/2^n$. Now suppose the $\$^+$ query comes first and outputs C; thus due to the collision we have that $C = V \oplus K^* \oplus \Delta_1$. But in this case the adversary can compute $K^* = V \oplus \Delta_1$, which happens with probability at most $1/2^n$ (as the key remains information theoretically hidden).

($\$^-, \mathsf{P}_1^{\pm}$) A direct $\mathsf{P}_1^{\pm}$ call in either direction collides with a point queried to $\$^-$ due to a query ξ^d to KCDec. This event is analyzed analogously to case ($\$^+, \mathsf{P}_3^{\pm}$) above.

($\$^+, \$^+$) The are two cases: (a) two inputs collide or (b) two outputs collide. In case (a) two distinct calls $\xi^e = (\Delta_1, \Delta_2, \alpha)$ and $(\xi^e)' = (\Delta_1', \Delta_2', \alpha')$ to the KCEnc oracle result in querying $\$^+$ on the same input. For this event to occur we must have that

$$\begin{aligned} &\mathsf{P}_2\big(\overbrace{\mathsf{P}_1(\underbrace{\Delta_1 \oplus \Delta_2 \oplus (1-\alpha)\cdot K^*}_{=:X}) \oplus \Delta_1 \oplus K^*}^{=:Y}\big) \oplus \Delta_1 \\ &\qquad = \mathsf{P}_2\big(\underbrace{\mathsf{P}_1(\overbrace{\Delta_1' \oplus \Delta_2' \oplus (1-\alpha')\cdot K^*}^{=:X'}) \oplus \Delta_1' \oplus K^*}_{=:Y'}\big) \oplus \Delta_1' \ . \end{aligned} \tag{7}$$

If $Y = Y'$ then $\mathsf{P}_1(X) \oplus \Delta_1 = \mathsf{P}_1(X') \oplus \Delta_1'$ and from (7) we get $\Delta_1 = \Delta_1'$. Since P_1 is a permutation, this implies that $X = X'$. If further $\alpha = \alpha'$ then we would also have $\Delta_2 = \Delta_2'$, which means $\xi^e = (\xi^e)'$, contradicting distinctness of queries. Thus $\alpha \neq \alpha'$ and hence $\Delta_2 \oplus \Delta_2' = K^*$ (since $X = X'$), which only happens with probability $1/2^n$ as it leads to guessing the key.
Assume now that $Y \neq Y'$.

- If the adversary does not query P_2 on either Y or Y' then $\mathsf{P}_2(Y)$ or $\mathsf{P}_2(Y')$ would be distributed randomly and independently of the adversary's view. Hence the probability that the above equality holds would be at most $1/(2^n - q)$ for each pair of queries.
- Suppose both Y and Y' are queried to P_2. Then P_1 must have been queried on both X or X', since otherwise the probability of computing Y or Y' would be at most $1/(2^n - q)$. If both X and X' are queried to P_1 and $\alpha = 0$ or $\alpha' = 0$, then for each query the probability of querying X (or X') to P_1 is $1/2^n$, because K^* is information-theoretically hidden. The remaining case is $\alpha = \alpha' = 1$. Since the adversary knows Y (it was queried to P_2), it can compute $K^* = Y \oplus \mathsf{P}_1(\Delta_1 \oplus \Delta_2) \oplus \Delta_1$. Again, since K^* is information-theoretically hidden, this can only happen with probability $1/2^n$.

In case (b) outputs of $\$^+$ collide with probability $1/2^n$ for each pair of queries.

$(\$^-, \$^-)$ Two distinct $\xi^d, (\xi^d)' \in \Xi^d$ to KCDec result in querying $\$^-$ on the same point. This event is analyzed analogously to case $(\$^+, \$^+)$ above.

$(\$^+, \tilde{\mathsf{P}}_3^-)$ Calls $\xi^e = (\Delta_1^e, \Delta_2^e, \alpha^e)$ to KCEnc and $\xi^d = (\Delta_1^d, \Delta_2^d, \alpha^d)$ to KCDec result in (a) querying $\tilde{\mathsf{P}}_3^-$ on an input matching an output of $\$^+$; or in (b) querying $\tilde{\mathsf{P}}_3^-$ on an input whose output matches an input of $\$^+$. Let C be the reply to query ξ^e. Then the value chosen by $\$^+$ is $C \oplus K^* \oplus \Delta_1^e$. Thus case (a) would happen if

$$C \oplus K^* \oplus \Delta_1^e = \alpha^d \cdot K^* \oplus \Delta_2^d \oplus K^* \oplus \Delta_1^d \ .$$

If $\alpha^d = 1$, then

$$C \oplus K^* \oplus \Delta_1^e = \Delta_2^d \oplus \Delta_1^d \ ,$$

which leads to guessing K^* and happens with probability at most $1/2^n$. On the other hand, if $\alpha^d = 0$, we would have that

$$C \oplus \Delta_1^e = \Delta_2^d \oplus \Delta_1^d \ .$$

But for such queries either $\tilde{\mathsf{P}}_3^-$ is sampled to be consistent with $\$^+$ (or it is not called at all when $\mathsf{Det}_\oplus$ returns "Yes").

Let us now look at case (b), which would happen if

$$\mathsf{P}_2\Big(\overbrace{\mathsf{P}_1(\underbrace{\Delta_1^e \oplus \Delta_2^e \oplus (1-\alpha^e)\cdot K^*}_{=:X}) \oplus \Delta_1^e \oplus K^*}^{=:Y}\Big) \oplus \Delta_1^e \oplus K^* = \tilde{\mathsf{P}}_3^-\big(\Delta_1^d \oplus \Delta_2^d \oplus (1-\alpha^d)\cdot K^*\big) \ . \quad (8)$$

If $\Delta_1^d \oplus \Delta_2^d \oplus (1-\alpha^d)\cdot K^*$ was directly queried to P_3^-, an inconsistency of type $(\$^+, \mathsf{P}_3^\pm)$ arises, which we dealt with above. On the other hand, it could be that this value was never queried to P_3^-.

If the KCEnc query comes before KCDec and the inputs are distinct—colliding inputs was case (a) above—$\tilde{\mathsf{P}}_3^-$ chooses a random value, which collides with the input to $\$^+$ with probability at most $1/(2^n - q)$.

Suppose now that the KCEnc query comes after KCDec. Suppose the input to KCEnc is $\xi^e = (\Delta_1^e, \Delta_2^e, \alpha^e)$. The value computed as input to $\$^+$ is independent of the output of $\tilde{\mathsf{P}}_3^-$ unless P_1^+ is kept consistent with $\$^-$ (since otherwise P_1 and P_2 are independent of $\tilde{\mathsf{P}}_3^-$). If independent, collisions happen with probability at most $1/(2^n - q)$. Otherwise, dependency implies that $M \oplus \Delta_1^d = \Delta_1^e \oplus \Delta_2^e$ where M is the output of KCDec. This query will be marked as illegal unless $\Delta_1^d \neq \Delta_1^e$. In this case the adversary has found a collision of the form

$$\mathsf{P}_2(Y' \oplus K^* \oplus \Delta_1^e) \oplus K^* \oplus \Delta_1^e = U \ ,$$

where $Y' = \mathsf{P}_2^-(U \oplus K^* \oplus \Delta_1^d) \oplus K^* \oplus \Delta_1^d$ is the value input to $\$^-$ and output by P_1^+ due to the consistency check in KCEnc. Rearranging and applying P_2^- to the above equality yields $Y' \oplus K^* \oplus \Delta_1^e = \mathsf{P}_2^-(U \oplus K^* \oplus \Delta_1^e)$. Substituting the expression for Y' we get that $\mathsf{P}_2^-(U \oplus K^* \oplus \Delta_1^d) \oplus \Delta_1^d \oplus \Delta_1^e = \mathsf{P}_2^-(U \oplus K^* \oplus \Delta_1^e)$. Since P_2^- is a permutation, this would happen iff $\Delta_1^d = \Delta_1^e$, which is an illegal query.

$(\$^-, \tilde{\mathsf{P}}_1^+)$ A call ξ^d to KCDec and a call ξ^e to KCEnc result in querying $\$^-$ and $\tilde{\mathsf{P}}_1^+$ on the same point. This event is analyzed analogously to case $(\$^+, \tilde{\mathsf{P}}_3^-)$ above.

$(\tilde{\mathsf{P}}_3^-, \mathsf{P}_3^\pm)$ A call ξ^d to KCDec and a direct query to $\mathsf{P}_3^\pm$ are such that either the inputs or the outputs collide. If the KCDec query comes *after* the direct query, since oracle $\tilde{\mathsf{P}}_3^-$ is sampled to be consistent with P_3, no inconsistency arises. If P_3 comes after KCDec then an inconsistency arises only with an entry on $\tilde{\mathsf{P}}_3^-$ that is also on $\$^+$. But in this case an inconsistency of the form $(\$^+, \mathsf{P}_3^\pm)$ has been found.

$(\tilde{\mathsf{P}}_1^+, \mathsf{P}_1^\pm)$ A call ξ^e to KCEnc and a direct query to $\mathsf{P}_1^\pm$ collide. This case is treated analogously to $(\tilde{\mathsf{P}}_3^-, \mathsf{P}_3^\pm)$.

KCEnc inconsistently rejects If $\mathsf{Det}_\oplus$ does not output an incorrect answer, $\perp$ is the correct answer. On the other hand, as shown at the end of previous section, $\mathsf{Det}_\oplus$ with $i = 1$ outputs an incorrect answer with probability at most $1/2^n$ per inputs checked.

KCDec inconsistently rejects Similarly to the above, this event occurs with probability at most $1/2^n$ per inputs checked.

The theorem follows by putting the above bounds together. □

Remark. A potential strengthening of the above theorem would be to consider a wider class of KCA attacks for which the ideal cipher is known to be secure. One possibility would be the class of key-unpredictable, claw-free, and xkcd-secure CDF sets as shown in Theorem 1. Although feasibility of this level of security claim for iterated Even–Mansour ciphers would follow from known indifferentiability [LS13, DSST17], this would require a larger number of rounds and also comes at the cost of lower levels of security. It remains an interesting open question to find the minimal number of rounds needed in the Even–Mansour ciphers that yields (Ξ^e, Ξ^d)-KC-CCA security with respect to *any* pair (Ξ^e, Ξ^d) for which the ideal cipher is also (Ξ^e, Ξ^d)-KC-CCA secure.

Remark. We note that our model does not consider the possibility of preprocessing attacks, and establishing security bounds based on the size of non-uniform advice and the number of on-line queries remains an interesting line of research for future work.

7 Authenticated Encryption

We have seen that it is possible to achieve KCA-secure blockciphers. Given this, we now turn our attention to higher-level primitives and investigate the possibility of constructing KCA-secure primitives which rely on blockciphers. A natural first step is to explore symmetric encryption. Specifically, we consider the notion of authenticated encryption with associated data (AEAD).

AE syntax. An AEAD scheme is a 3-tuple of algorithms $\mathrm{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where (1) Gen is the randomized key generation algorithm that returns a bit string K from a finite key-space $\mathcal{K}$ (which is typically $\{0,1\}^k$ for some $k \in \mathbb{N}$). (2) $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \times \mathcal{N} \times \mathcal{H} \to \mathcal{C}$ is the deterministic encryption algorithm which takes as input a key K, a message M from message space $\mathcal{M}$, a nonce N from nonce space $\mathcal{N}$, and possibly some associated

Game $\text{KC-AE}_{\text{SE}}^{\Xi^e,\Xi^d,\mathcal{A}}$:
$b \twoheadleftarrow \{0,1\}$;
$K^* \twoheadleftarrow \mathcal{K}$
$b' \twoheadleftarrow \mathcal{A}^{\text{KCEnc},\text{KCDec}}$
return $(b' = b)$

Proc. KCEnc(ξ^e):
$(K, M, N, H) \leftarrow \xi^e(K^*)$
$C_1 \leftarrow \mathsf{Enc}(K, M, N, H)$
if $T[K, M, N, H] = \mathsf{undef}$:
  $T[K, M, N, H] \twoheadleftarrow \{0,1\}^{|C_1|}$
$C_0 \leftarrow T[K, M, N, H]$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K, C_b, N, H)$
return C_b

Proc. KCDec(ξ^d, C):
$(K, N, H) \leftarrow \xi^d(K^*)$
if $(K, C, N, H) \in \mathsf{CL}$: return $\perp$
$M_1 \leftarrow \mathsf{Dec}(\xi^d, C, N, H)$
$M_0 \leftarrow \perp$
return M_b

Figure 7: the game defining the KC-AE security of SE with key-correlated inputs and private headers and nonces (which are not revealed to the adversary). The adversary is required not to repeat nonces in its KCEnc queries.

header data H from header-space $\mathcal{H}$ and returns a ciphertext C from some ciphertext-space $\mathcal{C}$. We write this as $C \leftarrow \mathsf{Enc}(K, M, N, H)$. (3) The deterministic decryption algorithm $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \times \mathcal{N} \times \mathcal{H} \to \mathcal{M} \cup \{\perp\}$ takes a key K, a ciphertext C, a nonce N, and possibly some associated header data H and returns either a plaintext M or the failure symbol $\perp \notin \mathcal{M}$. Correctness requires that $\mathsf{Dec}(K, \mathsf{Enc}(K, M, N, H), N, H) = M$ for all values of the inputs above.

AE security. We define the AE-security of $\text{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ by considering the game described on the left in Fig. 7. The AE advantage of an adversary $\mathcal{A}$ against an AEAD scheme is defined by

$$\mathbf{Adv}^{\text{ae}}_{\text{SE}}(\mathcal{A}) := 2 \cdot \Pr\left[\text{AE}^{\mathcal{A}}_{\text{SE}}\right] - 1 \ .$$

We require that $\mathcal{A}$ is nonce-respecting in that it does not repeat nonces in its *encryption* queries (but it may repeat them in decryption or across encryption and decryption queries).

KC-AE security. Let SE be an AEAD scheme as defined above. Let Ξ^e and Ξ^d be CDF sets with signatures $\xi^e : \mathcal{K} \longrightarrow \mathcal{K} \times \mathcal{M} \times \mathcal{N} \times \mathcal{H}$ and $\xi^d : \mathcal{K} \longrightarrow \mathcal{K} \times \mathcal{N} \times \mathcal{H}$ respectively. As is in the blockcipher case, Ξ^e is used to generate correlated keys *and* key-correlated messages. In addition to this, for authenticated encryption, we allow *all* inputs to the encryption algorithm to be key-correlated. Unlike the blockcipher case however, we restrict the use of Ξ^d to generate correlated keys, nonces, and headers as it seems unnatural to derive a key-correlated ciphertext for decryption. If $\xi^e \in \Xi^e$ and $\xi^d \in \Xi^d$ we note that the syntactical difference to SE when considering KC-AE security is that the encryption oracle in the scheme only takes ξ^e as input, while the decryption oracle takes (ξ^d, C) as described in the $\text{KC-AE}^{\Xi^e,\Xi^d,\mathcal{A}}_{\text{SE}}$ game on the right in Fig. 7. The KC-AE advantage of an adversary $\mathcal{A}$ against AEAD is defined by

$$\mathbf{Adv}^{\text{kc-ae}}_{\text{SE}}(\Xi^e, \Xi^d, \mathcal{A}) := 2 \cdot \Pr\left[\text{KC-AE}^{\Xi^e,\Xi^d,\mathcal{A}}_{\text{SE}}\right] - 1 \ .$$

We require that nonces are not repeated in KCEnc queries.

Security against key-correlated attacks is a strengthening of the standard notion of AE-security for a symmetric encryption scheme. As the syntax shows, the input to SE consists of K, M, N, and H. A natural question may arise as to which inputs we allow to be key-correlated. Although we would like to provide guarantees against *any* correlated inputs, there are two immediate issues with header security. It can (sometimes) be the case that headers are not private, and if they are heavily key-correlated, an adversary could trivially retrieve the key. For this reason, we require that header data is private. But a more subtle point, as shown in [BK11], is that even with private headers, it is possible for an adversary to perform a full key-recovery attack by taking advantage of the pattern of errors returned while list-checking during the decryption process which

may be key-dependent. This attack does not apply in our setting, as we return only one error symbol ($\perp$) for both illegal queries and a failed decryption process, and as such, an adversary cannot use this to gain advantage.[7]

When headers or nonces *are* public—formally this means revealing the header H or N under the KCEnc and KCDec oracles—we can achieve security by requiring that headers or nonces are key-*uncorrelated* in the sense that their values can be computed from the description of input CDFs without the need for K^*. Looking ahead, this requirement would allow a proof strategy similar to one below in the private setting to be also applied in the public setting, with the difference that the values of headers and nonces would now be computed from the descriptions of the input CDFs.

The HwN Transform. We introduce the Hash-with-Nonce (HwN) transform with the goal of achieving KCA-security for symmetric encryption. Working in the random-oracle model, HwN converts a conventional AE-secure AEAD scheme SE to a new one, $\overline{\mathrm{SE}} = \mathrm{HwN}[\mathrm{SE}, \mathsf{Gen}] = (\mathsf{Gen}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$, which we will prove to be KC-AE secure. Gen is its key-generation algorithm, its key length is k and nonce length is n. Its encryption and decryption algorithms are defined as follows, where $\mathsf{H} : \mathcal{K} \times \mathcal{N} \longrightarrow \mathcal{K}$ is a random oracle.

$\overline{\mathsf{Enc}}(K, M, N, H)$:	$\overline{\mathsf{Dec}}(K, C, N, H)$:
$C \leftarrow \mathsf{Enc}(\mathsf{H}(K\|N), M, N, H)$	$M \leftarrow \mathsf{Dec}(\mathsf{H}(K\|N), C, N, H)$
return C	return M

To show that $\overline{\mathrm{SE}}$ is (Ξ^e, Ξ^d)-KC-AE secure it will be convenient to reduce to a multi-user variant of the AE game. Multi-user authenticated encryption (MUAE) allows an adversary to choose the number of keys to target, analogous to the number of users in the scheme. In addition to taking a message, nonce, and header, the encryption oracle takes an index i (indicating user number), and returns encryptions under a key K_i, or a random string of the same length. Decryption always returns the error symbol, except in the case the real algorithms are being used where the message M is returned by decrypting C under K_i. The adversary cannot ask user i to decrypt a ciphertext that it previously obtained from this user. In a scheme with n users, it follows via a simple hybrid argument (see Appendix B) that any AE-secure AEAD scheme is also MUAE secure.

The convenience in using MUAE lies in the fact that we can leverage the programmability of random oracles to implicitly map the hash values $\mathrm{H}(\xi_i^e|_1(K^*)|N_i)$ of the i-th query ξ_i^e to KCEnc, to a new randomly and independently chosen key in the MUAE game. This also decouples the key used in encryption (i.e., $\mathrm{H}(\xi_i^e|_1(K^*)|N_i)$) from the key-correlated messages $\xi_i^e|_2(K^*)$, key-correlated nonces $\xi_i^e|_3(K^*)$, and key-correlated headers $\xi_i^e|_4(K^*)$. Overall, our reduction proceeds by choosing a K^*, computing the key-correlated message, nonce, and header, faithfully using this key, and implicitly setting $\mathrm{H}(\xi_i^e|_1(K^*)|N)$ to keys K_i in the MUAE game. For this analysis, we need that the keys K_i remain hidden from the adversary's point of view, which we argue is the case as long the queried functions are key-unpredictable. We first give a definition formalizing the unpredictability of correlated keys before formally stating and proving the security guarantees of $\overline{\mathrm{SE}}$

Theorem 3 (KCA security of HwN)**.** *Let* SE *be an* AE*-secure* AEAD *scheme. Let* Ξ^e *and* Ξ^d *be two CDF sets. Then for any* (Ξ^e, Ξ^d)-KC-AE *adversary* $\mathcal{A}$ *against the scheme* $\overline{\mathrm{SE}}$ *obtained by applying the* HwN *transform to* SE *that makes* q *random oracle queries,* q_e *encryption queries and* q_d *decryption queries, there are adversaries* $\mathcal{B}$*,* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *such that*

$$\mathbf{Adv}^{\text{kc-ae}}_{\overline{\mathrm{SE}}}(\Xi^e, \Xi^d, \mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{mu-ae}}_{\mathrm{SE}}(\mathcal{B}) + 2 \cdot \mathbf{Adv}^{\text{m-kup}}_{\Xi^e}(\mathcal{C}_1) + 2 \cdot \mathbf{Adv}^{\text{m-kup}}_{\Xi^d}(\mathcal{C}_2) \ ,$$

[7] Even if we define a model with different symbols to denote different types of decryption failures, we may be still able to obtain header security, but possibly requiring the key-claw-freeness of correlation-derivation functions.

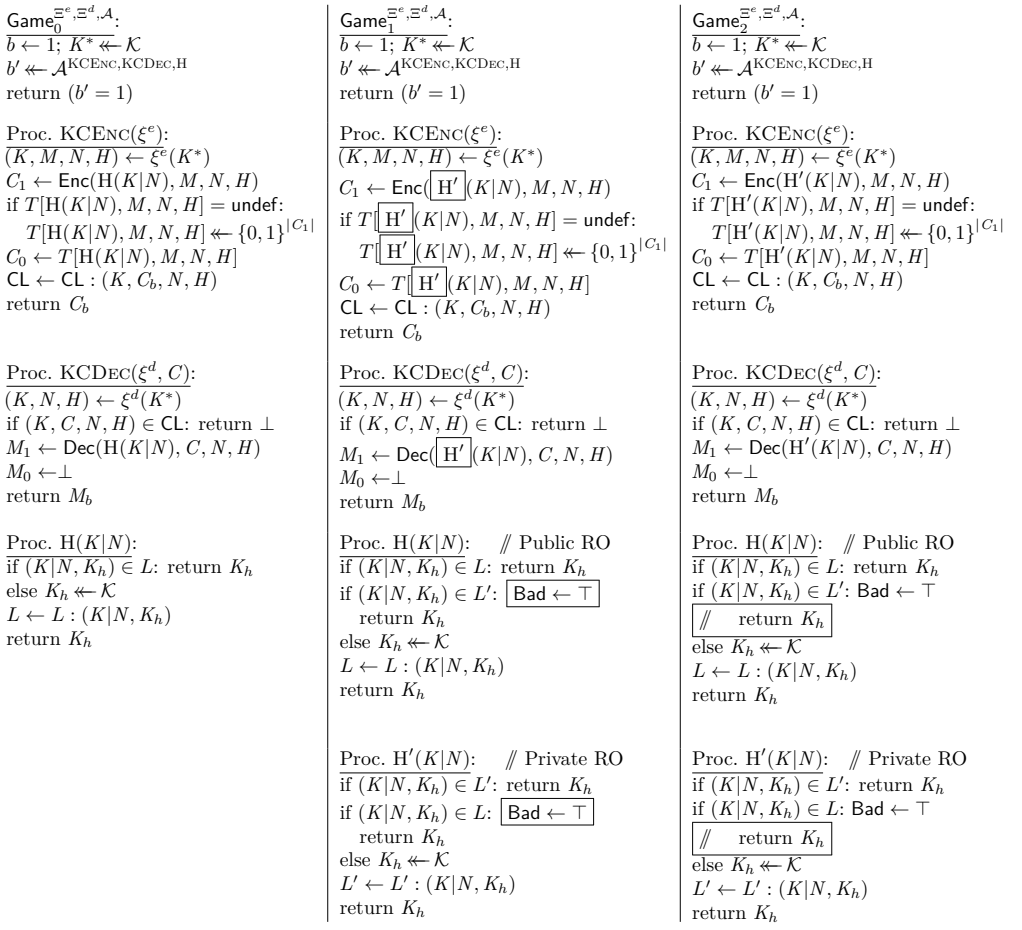


Figure 8: Game_0 shows the KC-AE game with respect to $\overline{\mathrm{SE}}$ and challenge bit 1. Game_1 separates the lists for H and H$'$. There are no functional changes. Game_2 then fully decouples H and H$'$.

where adversary $\mathcal{B}$ makes at most $q_e + q_d$ queries to Init*, at most q_e queries to encryption, and at most q_d queries to decryption, adversary $\mathcal{C}_1$ outputs two lists of sizes at most q_e and q, and adversary $\mathcal{C}_2$ outputs two lists of sizes at most q_d and q.*

Proof. The proof proceeds via a sequence of games as shown in Figs. 8 and 9. The three games in Fig. 8 run according to the KC-AE game with the HwN transform applied, and with $b = 1$. When $b = 1$, all oracle queries to KCEnc return encryptions of key-derived inputs under related keys. Queries to KCDec return the plaintext, or the error symbol $\perp$ if the ciphertext is deemed illegitimate. The three games in Fig. 9 run according to the KC-AE game with the HwN transform applied, and with $b = 0$. In this scenario, all oracle queries to KCEnc return random ciphertexts, and KCDec always returns the error symbol $\perp$. All games, adversaries, and transitions are outlined below.

Game_0 describes the KC-AE game with respect to the transformed scheme and $b = 1$.

Game_1 is identical to Game_0 with one conceptual change: it proceeds by separating the book-keeping list for H into cases when H is accessed within KCEnc and KCDec and when directly accessed by the adversary. More precisely, in Game_1, records of adversary $\mathcal{A}$'s queries to the random oracle H will be recorded in a list L, whereas

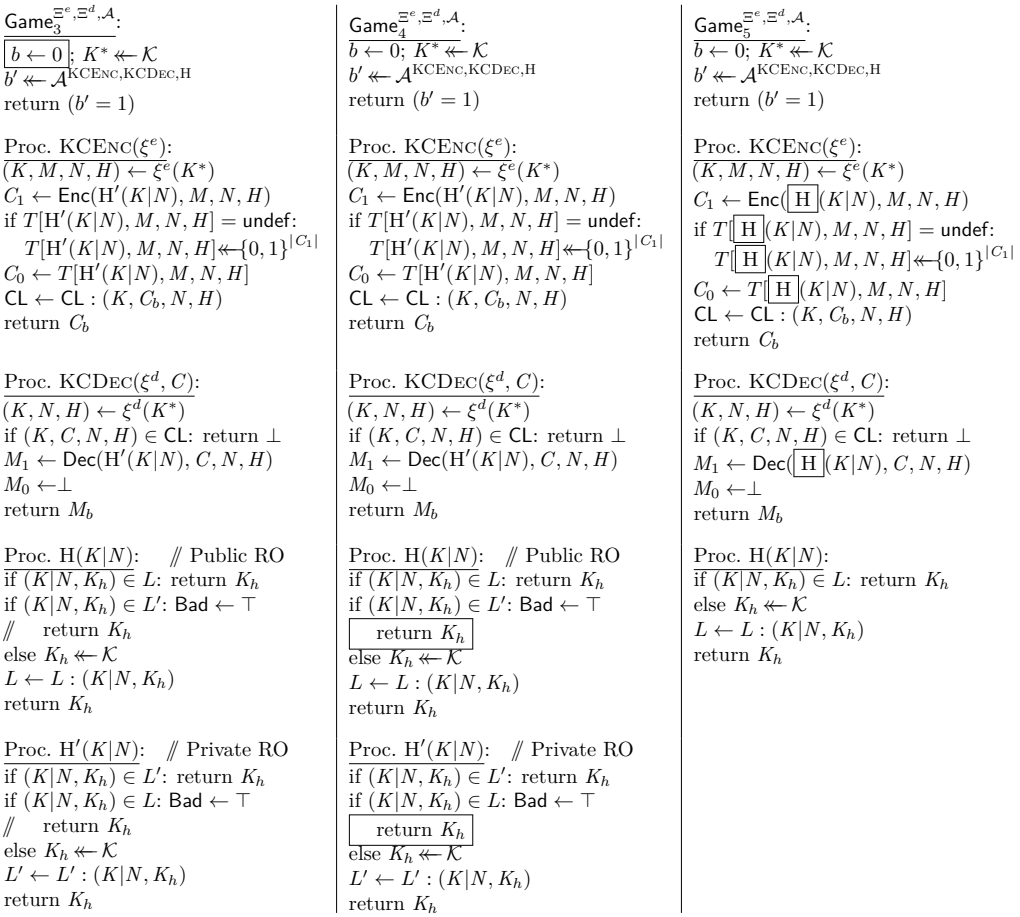


Figure 9: Game_3 changes the challenge bit to 0. Game_4 merges H and H′. Game_5 is functionally identical to Game_4 with merged H and H′. Game_5 is the KC-AE game with respect to $\overline{\text{SE}}$ and challenge bit 0.

the oracle queries from KCEnc and KCDec will be recorded in a list L' by oracle denoted H′. A bad flag, Bad, is set if H′ is queried on a key/nonce pair that has already been queried by $\mathcal{A}$ to H. In this game, H and H′ jointly implement H in Game_0 and event Bad prepares us to fully decouple H and H′ later:

$$\Pr[\mathsf{Game}_0^{\mathcal{A}}] = \Pr[\mathsf{Game}_1^{\mathcal{A}}] \ .$$

Game_2 is identical to Game_1 up to the point that Bad is set. If this event occurs within H′, records are no longer returned. By the fundamental lemma of game playing [BR06], we bound the difference between Game_2 and Game_1 as

$$\Pr[\mathsf{Game}_1^{\mathcal{A}}] - \Pr[\mathsf{Game}_2^{\mathcal{A}}] \leq \Pr[\mathsf{Game}_2^{\mathcal{A}} \text{ sets } \mathsf{Bad}] \ .$$

Event Bad corresponds to a non-empty intersection between the two lists L and L'. Below, we will bound $\Pr[\mathsf{Bad}]$ in Game_1 by showing that (1) this probability is close to the probability of setting Bad in Game_2 down to the MUAE security of SE; and (2) the probability of setting Bad in Game_2 is small down to the key unpredictability of Ξ^e and Ξ^d.

Game_3 switches the challenge bit to $b = 0$. We show below that any adversarial advantage in distinguishing Game_2 and Game_3 can be converted, via an adversary $\mathcal{B}_1$, against the MUAE game:

$$\Pr[\mathsf{Game}_2^{\mathcal{A}}] - \Pr[\mathsf{Game}_3^{\mathcal{A}}] = \mathbf{Adv}_{\mathrm{SE}}^{\text{mu-ae}}(\mathcal{B}_1) \ .$$

Game_4 reverses the change in Game_2 and merges random oracles H and H′. We have that

$$\Pr[\mathsf{Game}_3^{\mathcal{A}}] - \Pr[\mathsf{Game}_4^{\mathcal{A}}] \leq \Pr[\mathsf{Game}_3^{\mathcal{A}} \text{ sets } \mathsf{Bad}] \ ,$$

where we choose Game_3 to analyze the probability of Bad. Below we construct adversaries to show that this probability is bounded above by the key-unpredictability advantages against Ξ^e and Ξ^d.

Game_5 is identical to Game_4 except for a reversal of the conceptual change seen between Game_0 and Game_1. The random oracles are completely re-coupled, and all queries by $\mathcal{A}$, KCEnc and KCDec are written to a single list L. There are no procedural changes in this game:

$$\Pr[\mathsf{Game}_4^{\mathcal{A}}] = \Pr[\mathsf{Game}_5^{\mathcal{A}}] \ .$$

The reductions. We now give the details of the bounds for the three transitional changes above.

Adversary $\mathcal{B}_1$ plays the MUAE game and is built based on an adversary $\mathcal{A}$ that attempts to distinguish Game_2 and Game_3 as follows. Adversary $\mathcal{B}_1$ initializes q keys *and chooses* a K^*. It then simulates the encryption and decryption oracles by explicitly computing the actual key, and also message when answering encryption, used using K^*. In doing so, $\mathcal{B}_1$ *implicitly programs* $\mathrm{H}(\xi_i^e(K^*), N_i)$ to K_i, one of the q keys initialized by $\mathcal{B}_1$'s challenger. Since nonces are not reused, and each K_i is chosen randomly, this will constitute a perfect simulation of the random oracle H′. When $b = 0$, this is a perfect simulation of Game_4 environment for $\mathcal{A}$ by $\mathcal{B}_1$. However when $b = 1$, although simulation encryption is good, care must be taken in decryption as decryption of (K, C, N, H) is not allowed in the KC-AE game if (K, C, N, H) was added to list CL. Adversary $\mathcal{B}_1$ thus keeps track of these values and answers with $\perp$ when such a query arises. Note that $\mathcal{B}_1$ once again uses its knowledge K^* and further that this modification does not affect the simulation when $b = 0$. Adversary $\mathcal{B}_1$ continues in this way and returns the bit that $\mathcal{A}$ outputs. Hence,

$$\mathbf{Adv}_{\mathrm{SE}}^{\text{mu-ae}}(\mathcal{B}_1) = \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] = \Pr[\mathsf{Game}_2^{\mathcal{A}}] - \Pr[\mathsf{Game}_3^{\mathcal{A}}] \ .$$

Adversaries $\mathcal{C}_i$ for $i = 1, 2$ are used to show that the probability of Bad is small by the key-unpredictability of the queried functions. Adversaries $\mathcal{C}_i$ for $i = 1, 2$ run $\mathcal{A}$ in Game_3 by *simply simulating its encryption and decryption oracles with* \$ *and* $\perp$ *respectively.* They also lazily sample the random oracles H for $\mathcal{A}$. Adversary $\mathcal{C}_1$ keeps track of all queried functions to KCEnc and adversary $\mathcal{C}_2$ keeps track of all queried functions to KCDec. Both adversaries also keep track of all queries to H. When $\mathcal{A}$ terminates, $\mathcal{C}_i$ simply outputs the list of all queried functions that it keeps track of and H queries its two lists of guesses in the multi-shot key-unpredictability game. (See Fig. 10 (middle) for the details.) Whenever flag Bad is triggered, either $\mathcal{C}_1$ or $\mathcal{C}_2$ wins the key-unpredictability game against Ξ^e or Ξ^d respectively:

$$\Pr[\mathsf{Game}_3^{\mathcal{A}} \text{ sets } \mathsf{Bad}] \leq \mathbf{Adv}_{\Xi^e}^{\text{m-kup}}(\mathcal{C}_1) + \mathbf{Adv}_{\Xi^d}^{\text{m-kup}}(\mathcal{C}_2) \ .$$

Adversary $\mathcal{B}_1^{\text{Init,Enc,Dec}}$:
$i \leftarrow 0$; $K^* \twoheadleftarrow \mathcal{K}$
Call $\mathcal{B}_1$.Init $q_e + q_d$ times
$b' \twoheadleftarrow \mathcal{A}^{\text{KCEnc,KCDec,H}}$
return b'

KCEnc(ξ^e):
$(K, M, N, H) \leftarrow \xi^e(K^*)$
if $T[K, N] = \mathsf{undef}$:
  $i \leftarrow i + 1$; $T[K, N] \leftarrow i$
$C \leftarrow \mathcal{B}_1$.Enc$(T[K, N], M, N, H)$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K, C, N, H)$
return C

KCDec(ξ^d, C):
$(K, N, H) \leftarrow \xi^d(K^*)$
if $T[K, N] = \mathsf{undef}$:
  $i \leftarrow i + 1$; $T[K, N] \leftarrow i$
if $(K, C, N, H) \in \mathsf{CL}$: return $\perp$
$M \leftarrow \mathcal{B}_1$.Dec$(T[K, N], C, N, H)$
return M

H($K|N$):
if $(K|N, K_h) \in L$: return K_h
else $K_h \twoheadleftarrow \mathcal{K}$
$L \leftarrow L : (K|N, K_h)$
return K_h

Adversary $\mathcal{C}_i$:
$K^* \twoheadleftarrow \mathcal{K}$
$b' \twoheadleftarrow \mathcal{A}^{\text{KCEnc,KCDec,H}}$
return $(\mathsf{L}_i, \mathsf{L}')$

KCEnc(ξ^e):
$\mathsf{L}_1 \leftarrow \mathsf{L}_1 : \xi^e$
$C \twoheadleftarrow \{0,1\}^{|M|+\tau}$
return C

KCDec(ξ^d, C):
$\mathsf{L}_2 \leftarrow \mathsf{L}_2 : \xi^d$
return $\perp$

H($K|N$):
if $(K|N, K_h) \in L$:
  return K_h
else $K_h \twoheadleftarrow \mathcal{K}$
$L \leftarrow L : (K|N, K_h)$
$\mathsf{L}' \leftarrow \mathsf{L}' : K$
return K_h

Adversary $\mathcal{B}_2^{\text{Init,Enc,Dec}}$:
$i \leftarrow 0$; $K^* \twoheadleftarrow \mathcal{K}$
Call $\mathcal{B}_2$.Init q times
$b' \twoheadleftarrow \mathcal{A}^{\text{KCEnc,KCDec,H}}$
for $(\xi) \in \mathsf{L}_1 \wedge (K, N) \in \mathsf{L}'$
  if $\xi|_1(K^*) = K \wedge \xi|_3(K^*) = N$:
    return 1
return 0

KCEnc(ξ^e):
$\mathsf{L}_1 \leftarrow \mathsf{L}_1 : (\xi^e)$
$(K, M, N, H) \leftarrow \xi^e(K^*)$
if $T[K, N] = \mathsf{undef}$:
  $i \leftarrow i + 1$; $T[K, N] \leftarrow i$
$C \leftarrow \mathcal{B}_2$.Enc$(T[K, N], M, N, H)$
$\mathsf{CL} \leftarrow \mathsf{CL} : (K, C, N, H)$
return C

KCDec(ξ^d, C):
$\mathsf{L}_1 \leftarrow \mathsf{L}_1 : (\xi^d)$
$(K, N, H) \leftarrow \xi^d(K^*)$
if $T[K, N] = \mathsf{undef}$:
  $i \leftarrow i + 1$; $T[K, N] \leftarrow i$
$M \leftarrow \mathcal{B}_2$.Dec$(T[K, N], C, N, H)$
if $(K, C, N, H) \in \mathsf{CL}$: return $\perp$
return M

H($K|N$):
if $(K|N, K_h) \in L$: return K_h
else $K_h \twoheadleftarrow \mathcal{K}$
$L \leftarrow L : (K|N, K_h)$
$\mathsf{L}' \leftarrow \mathsf{L}' : (K, N)$
return K_h

Figure 10: On the left is the description of adversary $\mathcal{B}_1$ simulating the challenger of $\mathcal{A}$ in the KC-AE game whilst interacting in the MUAE environment. In the middle is the description of adversaries $\mathcal{C}_i$ simulating the oracle requests by $\mathcal{A}$ in the KC-AE game when $b = 0$ and whilst running in the key unpredictability environment. On the right is the code for an adversary $\mathcal{B}_2$ simulating $\mathcal{A}$ in the KC-AE game whilst interacting in the MUAE environment.

Adversary $\mathcal{B}_2$ is used to bound the difference between the probabilities of setting Bad in games Game_3 and Game_4. This is done via another reduction to the MUAE game which attempts to decide the challenge bit by observing if Bad is set. Algorithm $\mathcal{B}_2$ runs $\mathcal{A}$ similarly to algorithm $\mathcal{B}_1$. It however keeps track of the queried functions to KCEnc and KCDec as well as H. It uses its knowledge of K^* to verify if Bad has occurred. When the MUAE challenge bit of $\mathcal{B}_2$ is 1, algorithm $\mathcal{B}_2$ runs $\mathcal{A}$ in an environment identical to that of Game_2; hence the probability of $\mathcal{B}_2$ outputting 0 is exactly that of Bad in Game_2. Similarly, the probability of $\mathcal{B}_2$ outputting 1 is exactly that of Bad occurring in Game_3. Hence

$$\mathbf{Adv}^{\text{mu-ae}}_{\text{SE}}(\mathcal{B}_2) = \Pr[\mathsf{Game}_2^{\mathcal{A}} \text{ sets Bad}] - \Pr[\mathsf{Game}_3^{\mathcal{A}} \text{ sets Bad}] \ .$$

The theorem follows by putting the bounds established above together, and noting that

$$\mathbf{Adv}^{\text{kc-ae}}_{\overline{\text{SE}}}(\Xi^e, \Xi^d, \mathcal{A}) = \Pr[\mathsf{Game}_0^{\mathcal{A}}] - \Pr[\mathsf{Game}_5^{\mathcal{A}}] \ .$$

In the theorem statement, we choose $\mathcal{B}$ to be $\mathcal{B}_1$ or $\mathcal{B}_2$ having the bigger advantage. □

Remark. The above theorem can be extended to a misuse-resilient and key-correlated setting, whereby nonces may be repeated at encryption for a construction similar to HwN that also hashes the header information. A similar proof strategy can be used to analyse its KC-MRAE security at the expense of introducing three extra conditions: (1) the base scheme is MRAE secure; (2) the sets Ξ^e and Ξ^d are claw-free; and (3) (Ξ^e, Ξ^d) is xkcd so that illegal decryption queries can be detected.

Acknowledgments

We thank the reviewers of the IACR Transactions on Symmetric Cryptology (ToSC) for their detailed comments, which helped to improve the quality of the paper. We also thank Martijn Stam for his participation in the early stages of this work.

Farshim was supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 339563 - CryptoCloud). Fuchsbauer was supported by the *MSR–Inria Joint Centre* and the French ANR project *EfTrEC* (ANR-16-CE39-0002).

References

[AFPW11] Martin R. Albrecht, Pooya Farshim, Kenneth G. Paterson, and Gaven J. Watson. On cipher-dependent related-key attacks in the ideal-cipher model. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 128–145. Springer, Heidelberg, February 2011.

[App16] Benny Applebaum. Garbling XOR gates"for free" in the standard model. *Journal of Cryptology*, 29(3):552–576, July 2016.

[BCK11] Mihir Bellare, David Cash, and Sriram Keelveedhi. Ciphers that securely encipher their own keys. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *ACM CCS 11*, pages 423–432. ACM Press, October 2011.

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, Heidelberg, December 2011.

[BDH14] Florian Böhl, Gareth T. Davies, and Dennis Hofheinz. Encryption schemes secure under related-key and key-dependent message attacks. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 483–500. Springer, Heidelberg, March 2014.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 37–51. Springer, Heidelberg, May 1997.

[BDMN16] Lilian Bossuet, Nilanjan Datta, Cuauhtemoc Mancillas-López, and Mridul Nandi. Elmd: A pipelineable authenticated encryption and its hardware implementation. *IEEE Trans. Computers*, 65(11):3318–3331, 2016.

[Ber10] Daniel J. Bernstein. E-mail Discussion among the Participants of the Early Symmetric Crypto Seminar 2010, 2010.

[BF15] Manuel Barbosa and Pooya Farshim. The related-key analysis of Feistel constructions. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, volume 8540 of *LNCS*, pages 265–284. Springer, Heidelberg, March 2015.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision Diffie-Hellman. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 108–125. Springer, Heidelberg, August 2008.

[Bih94a] Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, December 1994.

[Bih94b] Eli Biham. New types of cryptanalytic attacks using related keys (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 398–409. Springer, Heidelberg, May 1994.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, Heidelberg, May 2003.

[BK09] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 1–18. Springer, Heidelberg, December 2009.

[BK11] Mihir Bellare and Sriram Keelveedhi. Authenticated and misuse-resistant encryption of key-dependent data. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 610–629. Springer, Heidelberg, August 2011.

[BKL+12] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations - (extended abstract). In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 45–62. Springer, Heidelberg, April 2012.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 231–249. Springer, Heidelberg, August 2009.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, Heidelberg, May / June 2006.

[BRS03] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In Kaisa Nyberg and Howard M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 62–75. Springer, Heidelberg, August 2003.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 513–525. Springer, Heidelberg, August 1997.

[BW99] Alex Biryukov and David Wagner. Slide attacks. In Lars R. Knudsen, editor, *FSE'99*, volume 1636 of *LNCS*, pages 245–259. Springer, Heidelberg, March 1999.

[CCS09] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 351–368. Springer, Heidelberg, April 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An identity escrow scheme with appointed verifiers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 388–407. Springer, Heidelberg, August 2001.

[CPS14] Benoit Cogliati, Jacques Patarin, and Yannick Seurin. Security amplification for the composition of block ciphers: Simpler proofs and new results. In Antoine Joux and Amr M. Youssef, editors, *SAC 2014*, volume 8781 of *LNCS*, pages 129–146. Springer, Heidelberg, August 2014.

[CS15] Benoit Cogliati and Yannick Seurin. On the provable security of the iterated Even-Mansour cipher against related-key and chosen-key attacks. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 584–613. Springer, Heidelberg, April 2015.

[DR01] Joan Daemen and Vincent Rijmen. The wide trail design strategy. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 222–238. Springer, Heidelberg, December 2001.

[DSST17] Yuanxi Dai, Yannick Seurin, John P. Steinberger, and Aishwarya Thiruvengadam. Indifferentiability of iterated Even-Mansour ciphers with non-idealized key-schedules: Five rounds are necessary and sufficient. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 524–555. Springer, Heidelberg, August 2017.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, June 1997.

[FKV17] Pooya Farshim, Louiza Khati, and Damien Vergnaud. Security of even–mansour ciphers under key-dependent messages. *IACR Trans. Symm. Cryptol.*, 2017(2):84–104, 2017.

[FP15] Pooya Farshim and Gordon Procter. The related-key security of iterated Even-Mansour ciphers. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 342–363. Springer, Heidelberg, March 2015.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, Heidelberg, March 2011.

[Har09] D. G. Harris. Generic ciphers are more vulnerable to related-key attacks than previously thought. International Workshop on Coding and Cryptography, WCC 2009, 2009.

[HK07] Shai Halevi and Hugo Krawczyk. Security under key-dependent inputs. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 07*, pages 466–475. ACM Press, October 2007.

[IK04] Tetsu Iwata and Tadayoshi Kohno. New security proofs for the 3GPP confidentiality and integrity algorithms. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 427–445. Springer, Heidelberg, February 2004.

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

[Knu93] Lars R. Knudsen. Cryptanalysis of LOKI91. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 196–208. Springer, Heidelberg, December 1993.

[LS13] Rodolphe Lampe and Yannick Seurin. How to construct an ideal cipher from a small set of public permutations. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 444–463. Springer, Heidelberg, December 2013.

[MP04] Ueli M. Maurer and Krzysztof Pietrzak. Composition of random systems: When two weak make one strong. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 410–427. Springer, Heidelberg, February 2004.

Game $\text{RK-CCA}_{\text{BC}}^{\Phi,\mathcal{A}}$:	Proc. $\text{RKEnc}(\phi, M)$:	Proc. $\text{RKDec}(\phi, C)$:
$b \twoheadleftarrow \{0,1\}$; $K^* \twoheadleftarrow \mathcal{K}$	$K \leftarrow \phi(K^*)$	$K \leftarrow \phi(K^*)$
$(\mathsf{iE}, \mathsf{iD}) \twoheadleftarrow \text{Block}(\mathcal{K}, \mathcal{M})$	$C \leftarrow \mathsf{E}(K, M)$	$M \leftarrow \mathsf{D}(K, C)$
$b' \twoheadleftarrow \mathcal{A}^{\text{RKEnc},\text{RKDec}}$	if $b = 0$: $C \leftarrow \mathsf{iE}(K, M)$	if $b = 0$: $M \leftarrow \mathsf{iD}(K, C)$
return $(b' = b)$	return C	return M

Game $\text{KDM-CCA}_{\text{BC}}^{\Psi,\mathcal{A}}$:	Proc. $\text{KDMEnc}(\psi)$:	Proc. $\text{Dec}(C)$:
$b \twoheadleftarrow \{0,1\}$; $K^* \twoheadleftarrow \mathcal{K}$	$M \leftarrow \psi(K^*)$	if $C \in \mathsf{CL}$: return $\perp$
$(\mathsf{iE}, \mathsf{iD}) \twoheadleftarrow \text{Block}(\mathcal{K}, \mathcal{M})$	$C \leftarrow \mathsf{E}(K^*, M)$	$M \leftarrow \mathsf{D}(K^*, C)$
$b' \twoheadleftarrow \mathcal{A}^{\text{KDMEnc},\text{Dec}}$	if $b = 0$: $C \leftarrow \mathsf{iE}(K^*, M)$	if $b = 0$: $M \leftarrow \mathsf{iD}(K^*, C)$
return $(b' = b)$	$\mathsf{CL} \leftarrow \mathsf{CL} : C$; return C	return M

Figure 11: (Top) Game defining the Φ-RK-CCA security of a blockcipher $\text{BC} = (\mathsf{E}, \mathsf{D})$. We require that $\phi \in \Phi$ for all queries ϕ. (Bottom) Game defining the Ψ-KDM-CCA security of a blockcipher $\text{BC} = (\mathsf{E}, \mathsf{D})$ with key space $\mathcal{K}$ and message space $\mathcal{M}$. We require that $\psi \in \Psi$ for all queries ψ.

Game $\text{KC-CCA}_{\text{BC}}^{\Xi^e,\Xi^d,\mathcal{A}}$:	Proc. $\text{KCEnc}(\xi^e)$:	Proc. $\text{KCDec}(\xi^d)$:
$b \twoheadleftarrow \{0,1\}$; $K^* \twoheadleftarrow \mathcal{K}$	$(K, M) \leftarrow \xi^e(K^*)$	$(K, C) \leftarrow \xi^d(K^*)$
$(\mathsf{iE}, \mathsf{iD}) \twoheadleftarrow \text{Block}(\mathcal{K}, \mathcal{M})$	if $(K, M) \in \mathsf{ML}$: return $\perp$	if $(K, C) \in \mathsf{CL}$: return $\perp$
$b' \twoheadleftarrow \mathcal{A}^{\text{KCEnc},\text{KCDec}}$	$C \leftarrow \mathsf{E}(K, M)$	$M \leftarrow \mathsf{D}(K, M)$
return $(b' = b)$	if $b = 0$: $C \leftarrow \mathsf{iE}(K, M)$	if $b = 0$: $M \leftarrow \mathsf{iD}(K, M)$
	$\mathsf{CL} \leftarrow \mathsf{CL} : (K, C)$	$\mathsf{ML} \leftarrow \mathsf{ML} : (K, M)$
	return C	return M

Figure 12: Game for (Ξ^e, Ξ^d)-KC-CCA security of a blockcipher $\text{BC} = (\mathsf{E}, \mathsf{D})$ from Sec. 3.

A Relation between KCA, RKA, and KDM

RKA and KDM security. Fig. 11 gives the definitions of RKA security [BK03] and KDM security [FKV17] for blockciphers. The advantage functions are defined in the standard way (as for KCA, Equation (1) in Sec. 3). The CPA versions are obtained naturally by disallowing decryption queries.

We now prove that for RKA-only (resp. KDM-only) claw-free sets KCA and RKA (resp. KDM) security are equivalent. For convenience we restate Fig. 1 as Fig. 12 and Proposition 1 from Section 3.3 as Propositions 2 and 3 below.

A.1 KC(RK-only) $\Leftrightarrow$ RK

Proposition 2. *Let* BC *be a blockcipher. Let* Ξ^e *and* Ξ^d *be two RKA-only CDF sets with* $\Xi^e = \Xi^d = \Phi \times \Gamma$. *Then for every adversary* $\mathcal{B}$ *there exists an adversary* $\mathcal{A}$ *such that*

$$\mathbf{Adv}_{\text{BC}}^{\text{rk-cca}}(\Phi, \mathcal{B}) \leq \mathbf{Adv}_{\text{BC}}^{\text{kc-cca}}(\Xi^e, \Xi^d, \mathcal{A}) + q^2/2^n \ ,$$

where n *is the block length and* q *is the maximum number of queries* $\mathcal{B}$ *makes. Moreover, for every adversary* $\mathcal{A}$ *there exist adversaries* $\mathcal{B}$ *and* $\mathcal{B}^{\text{cf}}$ *such that*

$$\mathbf{Adv}_{\text{BC}}^{\text{kc-cca}}(\Xi^e, \Xi^d, \mathcal{A}) \leq q^2 \cdot (\mathbf{Adv}_{\Phi}^{\text{cf}}(\mathcal{B}^{\text{cf}}) + 1/2^n) + \mathbf{Adv}_{\text{BC}}^{\text{rk-cca}}(\Phi, \mathcal{B}) \ ,$$

where q *is the maximum number of queries adversary* $\mathcal{A}$ *makes.*

Recall that (Ξ^e, Ξ^d) is RKA-only if $\Xi^e = \Xi^d = \Phi \times \Gamma$ for an RKA set Φ of functions mapping keys to keys (where Γ is the set of constant functions over $\mathcal{M}$). The difference

between games RK and KC is that in the former there are no lists ML and CL, thus all queries are allowed. When showing KC⇒RK, the reduction must therefore answer all queries, while its KC challenger might reply with $\perp$. The reduction will therefore keep local lists CL' and ML', which simulate the challenger's blacklists.

The KC challenger, when queried $\text{KCEnc}(\xi^e = (\phi^e, M))$ stores pairs $(K = \phi^e(K^*), C = \mathsf{E}(K, M))$ in its list CL. The reduction does not know the values K and will store pairs (C, M) in a list CL'. In the KC game, queries $\text{KCDec}(\xi^d = (\phi^d, C))$ with $(\phi^d(K^*), C) \in \mathsf{CL}$ are answered with $\perp$, whereas in the RK game, $\text{RKDec}(\phi^d, C)$ always returns $\mathsf{D}(\phi^d(K^*), C)$. To answer such queries, the reduction looks for an entry (C, M') for some M' in its list CL'. If (1) there is only one such entry, it returns M', which perfectly simulates RKDec: if $(C, M') \in \mathsf{CL}'$ and $\text{KCDec}(\xi^d = (\phi^d, C))$ returned $\perp$ then $(\phi^e(K^*), C) \in \mathsf{CL}$ and if there is only one entry with C, we must have $\phi^d(K^*) = \phi^e(K^*)$. If (2) there are several entries $(C, M_0), (C, M_1) \in \mathsf{CL}'$, the reduction aborts the simulation and returns 1 (that is, it guesses that it is interacting with the actual blockcipher). As we show in the lemma below, we can bound the probability of this type of collision in CL' in case the reduction is interacting with an ideal cipher, and therefore the probability of aborting and returning a wrong guess. The list ML and queries to the KCEnc oracle are dealt with analogously.

Lemma 1. *Consider game* $\text{KC-CCA}_{\dot{\mathrm{B}}\dot{\mathrm{C}}}^{\Xi^e, \Xi^d, \mathcal{A}}$ *with b fixed to 0. Then the probability that there are two entries $(K_0, C), (K_1, C) \in \mathsf{CL}$ with $\mathsf{iD}(K_0, C) \neq \mathsf{iD}(K_1, C)$ is at most $q^2/2^n$, where q is the number of queries made by $\mathcal{A}$. The same holds for the probability of two entries $(K_0, M), (K_1, M) \in \mathsf{ML}$ with $\mathsf{iE}(K_0, M) \neq \mathsf{iE}(K_1, M)$.*

Proof. Let us consider CL (the case ML is completely analogous). In order for a pair (K, C) to be added to CL, the adversary must make a query $\text{KCEnc}(\xi^e = (\phi^e, M))$ such that $K = \phi^e(K^*)$ and $C = \mathsf{iE}(K, M)$.

Since $(\mathsf{iE}, \mathsf{iD})$ is ideal, the probability that after q queries the adversary finds $K_0, K_1, M_0 \neq M_1$ such that $\mathsf{iE}(K_0, M_0) = \mathsf{iE}(K_1, M_1) =: C$, *without* having queried both $\mathsf{iD}(K_0, C)$ *and* $\mathsf{iD}(K_1, C)$ before, is at most $q^2/2^n$. On the other hand, when the adversary queries both $\text{KCDec}(\phi_0, C)$ and $\text{KCDec}(\phi_1, C)$ with $\phi_i(K^*) = K_i$, then (K_0, M_0) and (K_1, M_1), with $M_i := \mathsf{iD}(K_i, C)$, are both added to ML. But then any call to $\text{KCEnc}(\xi^e = (\phi_i', M_i))$ with $\phi'(K^*) = K_i$ will be answered with $\perp$, meaning that (K_i, C) will not be added to CL. So the only way for the event to happen is that the adversary guesses $K_0, K_1, M_0 \neq M_1$ which map to the same C. □

KC ⇒ RK. We now prove the first statement of Proposition 2. Consider an adversary $\mathcal{B}$ against RK; we construct a reduction $\mathcal{A}$ against KC, which maintains local lists CL' and ML', allowing it to simulate $\mathcal{B}$'s oracles even when its own oracles return $\perp$.

Whenever adversary $\mathcal{B}$ queries RKEnc on (ϕ, M), reduction $\mathcal{A}$ queries its own oracle KCEnc on $\xi^e\colon K \mapsto (\phi(K), M)$. If its answer is $\perp$ then there must be an entry (M, C') in ML'. If there is more than one such entry, $\mathcal{A}$ stops and returns 1; otherwise it returns C'. If $\mathcal{A}$'s KCEnc oracle returned a value $C \neq \perp$, $\mathcal{B}$ returns C and adds (C, M) to its local list CL'.

Analogously, if $\mathcal{B}$ queries RKDec on (ϕ, C), reduction $\mathcal{A}$ queries its own oracle KCDec on $\xi^d\colon K \mapsto (\phi(K), C)$. If its answer is $\perp$ then there must be an entry (C, M') in CL'. If there is more than one such entry, $\mathcal{A}$ stops and returns 1; otherwise it returns M'. If $\mathcal{A}$'s KCDec oracle returned a value $M \neq \perp$, $\mathcal{B}$ returns M and adds (M, C) to its list ML'.

If $\mathcal{A}$ has not stopped the simulation, it forwards $\mathcal{B}$'s output b'.

First note that to any entry $(K, C) \in \mathsf{CL}$ of $\mathcal{A}$'s challenger corresponds an entry $(C, \mathsf{D}(K, M))$ in $\mathcal{A}$'s local list CL' (and similarly for ML and ML'). From Lemma 1 we thus have that in case $b = 0$, the probability that there are two entires $(C, M_0), (C, M_1)$ with $M_0 \neq M_1$ in CL' is at most $q^2/2^n =: \epsilon$. Thus, the probability that $\mathcal{A}$ stops the simulation

in case $b = 0$ is bounded by ϵ. In case $b = 1$, if $\mathcal{A}$ stops the simulation, it outputs the correct bit. More formally, let E_i denote the event that $\mathcal{A}$ stops the simulation when its challenger's bit $b = i$ and let KC_i denote game $\text{KC-CCA}_{\mathrm{BC}}^{\Xi^e,\Xi^d,\mathcal{A}}$ with bit b fixed to i and similarly for RK_i. Then we have:

$$\begin{aligned}
\mathbf{Adv}_{\mathrm{BC}}^{\text{rk-cca}}(\Phi,\mathcal{B}) + 1 &= \Pr[\mathrm{RK}_0|E_0]\Pr[E_0] + \Pr[\mathrm{RK}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{RK}_1|E_1]\Pr[E_1] + \Pr[\mathrm{RK}_1|\neg E_1]\Pr[\neg E_1] \\
&= \Pr[\mathrm{RK}_0|E_0]\Pr[E_0] + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{RK}_1|E_1]\Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&\leq 1 \cdot \epsilon + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] + 1 \cdot \Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \Pr[\mathrm{KC}_0|E_0]\Pr[E_0] + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{KC}_1|E_1]\Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \mathbf{Adv}_{\mathrm{BC}}^{\text{kc-cca}}(\Xi^e, \Xi^d, \mathcal{A}) + 1 \ ,
\end{aligned}$$

where for the second equality we used that games RK and KC have the same output conditioned on $\neg E$; the inequality uses $\Pr[E_0] \leq \epsilon$ by Lemma 1; and the second to last equality follows from $\mathcal{B}$'s behavior in case of E happening, thus $\Pr[\mathrm{KC}_0|E_0] = 0$ and $\Pr[\mathrm{KC}_1|E_1] = 1$. This concludes this direction of the proof. □

The above simulation strategy for does not work for the converse implication RK $\Rightarrow$ KC, as best illustrated by the following example: Consider a blockcipher for which there exist $K \neq K' \in \mathcal{K}$ and $M \in \mathcal{M}$ with $\mathsf{E}(K, M) = \mathsf{E}(K', M) = C$; now consider a KC adversary $\mathcal{A}$ that queries $\text{KCEnc}(\phi, M)$ with $\phi(K^*) = K$; thus $(K, C) \in \mathsf{CL}$. Later $\mathcal{A}$ queries $\text{KCDec}(\phi', C)$ and $\text{KCDec}(\phi'', C)$ where $\phi'(K^*) = K'$ and $\phi''(K^*) = K$. Reduction $\mathcal{B}$ can forward these queries to its RK oracle and will get M in both cases. But $\text{KCDec}(\phi', C)$ should be answered with M (since $(K', C) \notin \mathsf{CL}$), whereas $\text{KCDec}(\phi', C)$ should be answered with $\perp$. Thus, storing (C, M) in a local list CL' doesn't help, since both queries would make $\mathcal{B}$ look at this entry, but should result in different actions.

The reduction can thus not perfectly simulate the KC experiment; although this situation is very unlikely to occur for the ideal cipher, the reduction cannot abort and return 1 (as in the above proof), since the event that the adversary found a tuple (K, K', M, C) that would be hard to find for the ideal cipher is not detectable by the reduction (in contrast to the proof for KC $\Rightarrow$ RK).

The proof will now rely on claw-freeness of the set Φ. If it was perfectly claw-free, then the reduction could perfectly simulate entries $(K, C) \in \mathsf{CL}$ by entries $(\phi, C) \in \mathsf{CL}'$ with $\phi(K^*) = K$. But if claw-freeness is only computational then the adversary could find ϕ, ϕ' with $\phi(K^*) = \phi'(K^*)$ for the key K^* chosen by the game, but which would not be a claw for a random key and thus cannot be output as a solution in the claw-freeness game.

However, we can show that until the adversary finds a claw, the ideal game can be simulated without having chosen any key K^* at all. If the adversary thus finds a claw, it must be one for a random key with non-negligible probability as well. We start with proving this last observation in the following lemma.

Lemma 2. *Consider game* $\text{RK-CCA}_{\mathrm{BC}}^{\Phi,\mathcal{B}}$ *with b fixed to 0 and define event E as follows: the adversary $\mathcal{B}$ makes two* RKEnc *queries (ϕ, M) and (ϕ', M), with $\phi \neq \phi'$, both answered by the same C; or it makes two* RKDec *queries (ϕ, C), (ϕ', C), with $\phi \neq \phi'$, both answered by the same M; or it makes two queries* $\text{RKEnc}(\phi, M)$*, answered by C and* $\text{RKDec}(\phi', C)$*, answered by M, with $\phi \neq \phi'$.*

Then there exists an adversary $\mathcal{A}^{\mathrm{cf}}$ such that $\Pr[E] \leq q^2 \cdot (\mathbf{Adv}_{\Phi}^{\mathrm{cf}}(\mathcal{A}^{\mathrm{cf}}) + 1/2^n) =: \epsilon_2$*, where q is the number of oracle queries $\mathcal{B}$ makes in game* RK-CCA*.*

Proof. The crucial observation is that game $\text{RK-CCA}_{\mathrm{BC}}^{\Phi,\mathcal{B}}$ when b is fixed to 0 can be simulated without knowing choosing K^* up to the event E happening, that is, the point

where the adversary produces the first claw. Let $(\phi_i)_{i=1}^q$ denote the list of all functions queried to RKEnc or RKDec. We construct an adversary $\mathcal{A}^{\text{cf}}$ against claw-freeness as follows: it first guesses $i_0, i_1 \in [1, q]$ and then simulates the game for $\mathcal{B}$ until the i_1-th query as follows:

$\mathcal{A}^{\text{cf}}$ keeps a list L with entries of the form (ϕ, M, C). Whenever $\mathcal{B}$ queries $\text{RKEnc}(\phi, M)$, it checks whether there is some entry $(\phi, M, C') \in L$ and if so, returns C'. Else it samples $C \leftarrow \mathcal{M}$, adds (ϕ, M, C) to L and returns C. Whenever $\mathcal{B}$ queries $\text{RKDec}(\phi, C)$, it checks whether there is some entry $(\phi, M', C) \in L$ and if so, returns M'. Else it samples $M \leftarrow \mathcal{M}$, adds (ϕ, M, C) to L and returns M. When $\mathcal{B}$ makes its i_1-the query, $\mathcal{A}^{\text{cf}}$ stops and returns (ϕ_{i_0}, ϕ_{i_1}).

Assume event E occurred and $\mathcal{A}^{\text{cf}}$ guessed correctly that the i_1-th query was the first query after which E occurred and the i_0-th query was the one it collided with. Consider $\mathcal{A}^{\text{cf}}$'s challenger, which chooses $K^* \leftarrow \mathcal{K}$ and checks whether $\phi_{i_0}(K^*) = \phi_{i_1}(K^*)$. If so, then $\mathcal{A}^{\text{cf}}$ won the claw-freeness game. Else the probability that $\mathcal{A}^{\text{cf}}$ would have answered the i_1-th query so that E still occurred is $1/2^n$. □

RK ⇒ KC. This direction relies on the claw-freeness of the set Φ. Let $\mathcal{A}$ be a KC adversary; we construct a reduction $\mathcal{B}$ against RK that maintains two local lists ML' and CL' and aborts whenever event E, defined in Lemma 2, occurs.

Whenever $\mathcal{A}$ queries KCEnc for $\xi^e \colon K \mapsto (\phi(K), M)$, reduction $\mathcal{B}$ queries $\text{RKEnc}(\phi, M)$ to receive C. It first checks whether there has been a "colliding" RKEnc query: if there is an entry (ϕ', C, M) in its list CL' with $\phi' \neq \phi$ then $\mathcal{B}$ aborts and returns 1.

To answer the query $\text{KCEnc}(\phi, M)$, an actual KC challenger would check whether $(\phi(K^*), M) \in \mathsf{ML}$. Instead, $\mathcal{B}$ checks its local list ML': if $(\phi, M, C) \in \mathsf{ML}'$, it returns $\perp$; if there is an entry $(\phi' \neq \phi, M, C)$ then $\mathcal{B}$ aborts the simulation and returns 1; if there is no entry $(\cdot, M, C) \in \mathsf{ML}'$, it adds (ϕ, C, M) to its local list CL' and returns C.

KCDec queries for $\xi^d \colon K \mapsto (\phi(K), C)$ are dealt with analogously: $\mathcal{B}$ first queries $\text{RKDec}(\phi, C)$ to receive M. If there is an entry (ϕ', M, C) in ML' with $\phi' \neq \phi$ then $\mathcal{B}$ aborts and returns 1. To answer $\text{KCDec}(\phi, C)$, an actual KC challenger would now check whether $(\phi(K^*), C) \in \mathsf{CL}$. Instead, $\mathcal{B}$ checks its local list CL': if $(\phi, C, M) \in \mathsf{CL}'$ it returns $\perp$; if there is an entry $(\phi' \neq \phi, C, M)$ then $\mathcal{B}$ aborts the simulation and returns 1; if there is no entry $(\cdot, C, M) \in \mathsf{CL}'$, it adds (ϕ, M, C) to its local list ML' and returns M.

We show that the simulation is perfect if $\mathcal{B}$ does not abort; consider a query $\text{KCDec}(\phi, C)$: (1) If $(\phi, C, M) \in \mathsf{CL}'$ then there must have been a query $\text{KCEnc}(\phi, M)$ with $C = \mathsf{E}(\phi(K^*), M)$; thus a KC challenger would have added $(\phi(K^*), C)$ to CL and the above KCDec query would have been answered by $\perp$. (2) If there is no entry $(\cdot, C, M) \in \mathsf{CL}'$ then $(\phi(K^*), C)$ cannot be in CL, since whenever it is added to CL, an entry $(\cdot, C, \mathsf{D}(\phi(K^*), C)$ would have been added to CL'. Perfect simulation of KCEnc queries is argued completely analogously.

On the other hand, we have that $\mathcal{B}$ aborts precisely when E, as defined in Lemma 2, occurs. With ϵ as defined in the latter, we thus have

$$
\begin{aligned}
\mathbf{Adv}_{\text{BC}}^{\text{kc-cca}}(\Xi^e, \Xi^d, \mathcal{A}) + 1 &= \Pr[\text{KC}_0|E_0]\Pr[E_0] + \Pr[\text{KC}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\text{KC}_1|E_1]\Pr[E_1] + \Pr[\text{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&= \Pr[\text{KC}_0|E_0]\Pr[E_0] + \Pr[\text{RK}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\text{KC}_1|E_1]\Pr[E_1] + \Pr[\text{RK}_1|\neg E_1]\Pr[\neg E_1] \\
&\leq 1 \cdot \epsilon + \Pr[\text{RK}_0|\neg E_0]\Pr[\neg E_0] + 1 \cdot \Pr[E_1] + \Pr[\text{RK}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \Pr[\text{RK}_0|E_0]\Pr[E_0] + \Pr[\text{RK}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\text{RK}_1|E_1]\Pr[E_1] + \Pr[\text{RK}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \mathbf{Adv}_{\text{BC}}^{\text{rk-cca}}(\Phi, \mathcal{B}) + 1 \ ,
\end{aligned}
$$

where for the second equality we used that games RK and KC have the same output conditioned on $\neg E$; the inequality uses $\Pr[E_0] \leq \epsilon$ by Lemma 2; and the second to last equality follows from $\mathcal{B}$'s behavior in case of E happening, thus $\Pr[\mathrm{RK}_0|E_0] = 0$ and $\Pr[\mathrm{RK}_1|E_1] = 1$. This concludes this direction of the proof. □

A.2 KC(KDM-only) ⇔ KDM

Proposition 3. *Let* BC *be a blockcipher. Let* Ξ^e *and* Ξ^d *be two KDM-only CDF sets with* $\Xi^e = \{id\} \times \Psi$ *and* $\Xi^d = \{id\} \times \Gamma$. *Then for every adversary* $\mathcal{A}$ *there exists an adversary* $\mathcal{B}$ *such that*

$$\mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{A}) = \mathbf{Adv}^{\text{kdm-cca}}_{\text{BC}}(\Psi, \mathcal{B}) \ ,$$

and for every adversary $\mathcal{B}$ *there exist adversaries* $\mathcal{A}$ *and* $\mathcal{A}^{\text{cf}}$ *such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\text{BC}}(\Psi, \mathcal{B}) \leq q^2 \cdot \mathbf{Adv}^{\text{cf}}_{\Psi}(\mathcal{A}^{\text{cf}}) + \mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{A}) \ ,$$

where q *is the maximum number of queries adversary* $\mathcal{B}$ *makes.*

We show that our definition is equivalent to the KDM definition from [FKV17] for claw-free sets Ψ. Recall that (Ξ^e, Ξ^d) is KDM-only if $\Xi^e = \{id\} \times \Psi$ and $\Xi^d = \{id\} \times \Gamma$, where Ψ is a KDM set of functions mapping keys to messages and Γ is the set of constant functions $K \mapsto M$ for all $M \in \mathcal{M}$; we assume $\Gamma \subseteq \Psi$.

We first observe that games KC and KDM are similar in that every entry C in CL in game KDM corresponds to (K^*, C) in CL in KC. The crucial difference is that in KDM there is no list ML, thus all KDMEnc queries are allowed, whereas KC disallows queries in ML. KDM implying KC can be shown because the reduction can perfectly simulate the list ML.

The converse (KC ⇒ KDM) is trickier to show and it relies on claw-freeness of the set Ψ. To illustrate this, consider the following attack against KDM, which does not work against KC: for two random $C_0, C_1 \in \mathcal{M}$, the adversary queries $\textsc{Dec}(C_0)$ and $\textsc{Dec}(C_1)$ to receive M_0 and M_1. It then queries $\text{KDM}\textsc{Enc}(\psi_{M_0,M_1})$ with $\psi\colon K^* \mapsto M_{K_1^*}$, where K_1^* is the first bit of K^*. When receiving C_i as a reply, the adversary learned that i is the first bit of K^* and it can use the same strategy to learn all other bits of K^*. Note that in the KC game, the two decryption queries add (K^*, M_0) and (K^*, M_1) to ML and consequently $\text{KC}\textsc{Enc}(\xi^e := id \times \psi_{M_0,M_1})$ is answered with $\perp$.

KDM ⇒ KC. To show that KDM implies KC, consider a KC adversary $\mathcal{A}$ and let us construct a reduction $\mathcal{B}$ against KDM. $\mathcal{B}$ basically simulates the KC oracles for $\mathcal{A}$ by forwarding $\mathcal{A}$'s queries to its own KDM oracles. To simulate game KC correctly, $\mathcal{B}$ maintains its own list ML'. Whenever $\mathcal{A}$ queries $\text{KC}\textsc{Dec}(\xi^d)$ for $\xi_2 \in \Xi^d$, we have $\xi_2\colon K \mapsto (K, C)$ for some $C \in \mathcal{M}$. Reduction $\mathcal{B}$ queries C to $\textsc{Dec}$ to obtain M, which it returns to $\mathcal{A}$; $\mathcal{B}$ also appends (M, C) to its list ML' (whereas an actual KC challenger would have appended (K^*, M) to its list ML).

Whenever $\mathcal{A}$ queries $\text{KC}\textsc{Enc}(\xi_1)$ with $\xi_1\colon K \mapsto (K, \psi(K))$ for some $\psi \in \Psi$, the reduction queries $\text{KDM}\textsc{Enc}(\psi)$ to obtain C. If for some M' there is an entry $(M', C) \in \mathsf{ML}'$: $\mathcal{B}$ returns $\perp$; otherwise, it returns C.

To see that this correctly simulates the list ML that $\mathcal{B}$'s challenger maintains, observe that (K^*, M) is in the latter iff $(M, \mathsf{E}(K^*, M))$ is in $\mathcal{B}$'s local list ML' and since $\mathsf{E}(K^*, \cdot)$ is a permutation, each C uniquely determines the corresponding M, thus $\mathcal{B}$'s simulation is perfect. We thus have $\mathbf{Adv}^{\text{kc-cca}}_{\text{BC}}(\Xi^e, \Xi^d, \mathcal{A}) = \mathbf{Adv}^{\text{kdm-cca}}_{\text{BC}}(\Psi, \mathcal{B})$.

KC ⇒ KDM. In this direction, we assume an adversary $\mathcal{B}$ against KDM and construct $\mathcal{A}$ against KC. This time, $\mathcal{A}$ will not be able to perfectly simulate the game; however, $\mathcal{A}$ can detect when it is not able to. In this case $\mathcal{A}$ aborts and returns $b' = 1$ (it guesses that it is talking to the real blockcipher). As for in the proof for KC ⇒ RK, we show

that if $b = 0$ (the ideal case), the probability of $\mathcal{A}$ aborting is bounded—in contrast to the previous proof, we bound it by the advantage of breaking claw-freeness.

We start by defining the following event E in game KDM-CCA$^{\Psi,\mathcal{B}}_{\mathrm{BC}}$: let $(M_i)_i$ denote the messages returned by the DEC oracle and let $(\psi_j)_j$ denote the queries made to the KDMENC oracle; let K^* denote the challenge key. We say E occurs if we either have (1) $\psi_{j_0}(K^*) = \psi_{j_1}(K^*)$ for some $j_0 \neq j_1$; or (2) $\psi_j(K^*) = M_i$ for some j, i. We show the following:

Lemma 3. *Consider game* KDM-CCA$^{\Psi,\mathcal{B}}_{\mathrm{BC}}$ *with b fixed to 0 and let E be the event defined above. Then there exists an adversary $\mathcal{A}^{\mathrm{cf}}$ such that* $\Pr[E] \leq q^2 \cdot \mathbf{Adv}^{\mathrm{cf}}_{\Psi}(\mathcal{A}^{\mathrm{cf}})$*, where q is the number of oracle queries $\mathcal{B}$ makes in game* KDM-CCA.

Proof. First note that there exists some first query j_1 that makes E occur, and some query $j_0 < j_1$, which will be the query with which the j_1-th query "collides". $\mathcal{A}^{\mathrm{cf}}$ starts with guessing j_0 and j_1 and simulates the game until the j_1-th query as follows: any query DEC(C_i) is answered with a random $M_i \leftarrow \mathcal{M}$; any query KDMENC$(\psi_i)$ is answered with a random $C_i \leftarrow \mathcal{M}$ (all repeated queries are answered consistently).

If the j_0-th query was a query DEC(C_i) answered by M_i then define $\psi_{j_0} \colon K \mapsto M_i$; if it was a query KDMENC(ψ) then let $\psi_{j_0} := \psi$; define ψ_{j_1} analogously. $\mathcal{A}^{\mathrm{cf}}$ returns (ψ_{j_0}, ψ_{j_1}).

First note that if $\mathcal{A}$ guessed j_1 correctly then the simulation is perfect, as for the random key K^*, the same message was never queried twice (via encryption or decryption). Moreover, if $\mathcal{A}$ also guessed j_0 correctly then (ψ_{j_0}, ψ_{j_1}) is indeed a collision and $\mathcal{A}$ wins the claw-freeness game. We thus have $\mathbf{Adv}^{\mathrm{cf}}_{\Psi}(\mathcal{A}^{\mathrm{cf}}) \geq 1/q^2 \cdot \Pr[E]$, which proves the lemma. □

We now give a reduction $\mathcal{A}$ against KC that simulates the KDM game to adversary $\mathcal{B}$ by forwarding all queries to its own oracle. When $\mathcal{A}$ cannot answer a KDMENC query because its own oracle KCENC replies $\perp$ it aborts the simulation. In order to use the above lemma, $\mathcal{A}$ will also abort when it could continue the simulation. In particular, $\mathcal{A}$ aborts and returns 1 when one of the two things happen:

(1) $\mathcal{B}$ makes two queries ψ, ψ' to KDMENC, which when forwarded to $\mathcal{A}$'s RKENC oracle return the same C; or

(2) $\mathcal{A}$ forwards a KDMENC query to its KDMENC oracle which is answered with $\perp$.

Note that these two cases define precisely the event E above. We thus have that $\mathcal{A}$ aborts the simulation whenever E occurs. If $\mathcal{A}$ does not abort the simulation, it returns whatever $\mathcal{B}$ outputs.

Let KC$_i$ denote game KC-CCA$^{\Xi^e, \Xi^d, \mathcal{A}}_{\mathrm{BC}}$ with bit b fixed to i and E_i the event that E occurs in this case. Define KDM$_i$ analogously and let $\epsilon := q^2 \cdot \mathbf{Adv}^{\mathrm{cf}}_{\Psi}(\mathcal{A}^{\mathrm{cf}})$. Then we have:

$$
\begin{aligned}
\mathbf{Adv}^{\text{kdm-cca}}_{\mathrm{BC}}(\Psi, \mathcal{B}) + 1 &= \Pr[\mathrm{KDM}_0|E_0]\Pr[E_0] + \Pr[\mathrm{KDM}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{KDM}_1|E_1]\Pr[E_1] + \Pr[\mathrm{KDM}_1|\neg E_1]\Pr[\neg E_1] \\
&= \Pr[\mathrm{KDM}_0|E_0]\Pr[E_0] + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{KDM}_1|E_1]\Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&\leq 1 \cdot \epsilon + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] + 1 \cdot \Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \Pr[\mathrm{KC}_0|E_0]\Pr[E_0] + \Pr[\mathrm{KC}_0|\neg E_0]\Pr[\neg E_0] \\
&\quad + \Pr[\mathrm{KC}_1|E_1]\Pr[E_1] + \Pr[\mathrm{KC}_1|\neg E_1]\Pr[\neg E_1] \\
&= \epsilon + \mathbf{Adv}^{\text{kc-cca}}_{\mathrm{BC}}(\Xi^e, \Xi^d, \mathcal{A}) + 1 \ ,
\end{aligned}
$$

where for the second equality we used that games KDM and KC have the same output conditioned on $\neg E$; the inequality uses $\Pr[E_0] \leq \epsilon$ by Lemma 3; and the second to last equality follows from $\mathcal{B}$'s behavior in case of E happening, thus $\Pr[\mathrm{KC}_0|E_0] = 0$ and $\Pr[\mathrm{KC}_1|E_1] = 1$. This concludes this direction of the proof.

Game $\mathrm{MUAE}^{\mathcal{A}}_{\mathrm{SE}}$:	Proc. $\textsc{Enc}(i, M, N, H)$:	Proc. $\textsc{Dec}(i, C, N, H)$:
$b \twoheadleftarrow \{0,1\}; v \leftarrow 0$	if $\neg(1 \le i \le v)$: return $\bot$	if $\neg(1 \le i \le v)$: return $\bot$
$b' \twoheadleftarrow \mathcal{A}^{\textsc{Init},\textsc{Enc},\textsc{Dec}}$	$C_1 \leftarrow \mathsf{Enc}(K_i, M, N, H)$	if $(i, C, N, H) \in \mathsf{CL}$: return $\bot$
return $(b' = b)$	$C_0 \twoheadleftarrow \{0,1\}^{\lvert C_1 \rvert}$	$M_1 \leftarrow \mathsf{Dec}(K_i, C, N, H)$
	$\mathsf{CL} \leftarrow \mathsf{CL} : (i, C_b, N, H)$	$M_0 \leftarrow \bot$
Proc. $\textsc{Init}()$:	return C_b	return M_b
$v \leftarrow v + 1$; $K_v \twoheadleftarrow \mathcal{K}$		

Figure 13: Game $\mathrm{MUAE}^{\mathcal{A}}_{\mathrm{AE}}$ defining the multi-user AE-security of a symmetric encryption scheme SE.

$\mathsf{Game}^{\mathcal{A}}_j$:	Proc. $\textsc{Enc}(i, M, N, H)$:	Proc. $\textsc{Dec}(i, C, N, H)$:
$b \leftarrow 1; v \leftarrow 0$	if $\neg(1 \le i \le v)$: return $\bot$	if $\neg(1 \le i \le v)$: return $\bot$
$b' \twoheadleftarrow \mathcal{A}^{\textsc{Init},\textsc{Enc},\textsc{Dec}}$	$C \leftarrow \mathsf{Enc}(K_i, M, N, H)$	if $(i, C, N, H) \in \mathsf{CL}$: return $\bot$
return $(b' = 1)$	if $i \le j$: $C \leftarrow \{0,1\}^{\lvert C \rvert}$	if $i \le j$: $M \leftarrow \bot$
	$\mathsf{CL} \leftarrow \mathsf{CL} : (i, C, N, H)$	$M \leftarrow \mathsf{Dec}(K_i, C, N, H)$
Proc. $\textsc{Init}()$:	return C	return M
$v \leftarrow v+1$; $K_v \twoheadleftarrow \mathcal{K}$		

Figure 14: Game_j (with $0 \le j \le n$) showing MUAE game which returns random values when $i \le j$ and an encryption under K_i otherwise.

B Multi-User to Single-User Reduction for AE

Multi-user security. We define the MUAE-security of $\mathrm{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ by considering the game described in Fig. 13. The MUAE advantage of an adversary $\mathcal{A}$ against an MUAE game is defined by

$$\mathbf{Adv}^{\text{mu-ae}}_{\mathrm{SE}}(\mathcal{A}) := 2 \cdot \Pr\left[\mathrm{MUAE}^{\mathcal{A}}_{\mathrm{SE}}\right] - 1 \ .$$

Note that nonces may not be repeated for each user in Enc.

Theorem 4. *Let* SE *be an authenticated encryption scheme. Then for any adversary* $\mathcal{A}$ *attacking the* MUAE*-security of* SE *while making at most* n *queries to* Init*, there exist* AE *adversaries* $\mathcal{B}_1, \ldots, \mathcal{B}_n$ *such that*

$$\mathbf{Adv}^{\text{mu-ae}}_{\mathrm{SE}}(\mathcal{A}) \le \sum_{i=1}^{n} \mathbf{Adv}^{\text{ae}}_{\mathrm{SE}}(\mathcal{B}_i) \ .$$

Proof. The proof proceeds via a standard hybrid argument. We start by defining a sequence of games Game_j (for $0 \le j < n$) as shown in Fig. 14. Game_j runs similarly to the MUAE game except that if i (the index of the key) is less than or equal to j, then ciphertext C is randomly generated. For $i > j$, the encryption algorithm is used under key K_i. Thus Game_0 is identical to the MUAE game with $b = 1$ and Game_n is identical to the MUAE game with $b = 0$.
Let $\mathcal{A}$ be any adversary in the MUAE game. To bound the advantage of $\mathcal{A}$, note that

$$\begin{aligned}\mathbf{Adv}^{\text{mu-ae}}_{\mathrm{SE}}(\mathcal{A}) &:= \Pr[\mathsf{Game}^{\mathcal{A}}_0] - \Pr[\mathsf{Game}^{\mathcal{A}}_n] \\ &= \sum_{j=0}^{n-1} (\Pr[\mathsf{Game}^{\mathcal{A}}_j] - \Pr[\mathsf{Game}^{\mathcal{A}}_{j+1}]) \ .\end{aligned}$$

Thus, for $j = 0, \ldots, n-1$ it remains to bound

$$\Pr[\mathsf{Game}^{\mathcal{A}}_j] - \Pr[\mathsf{Game}^{\mathcal{A}}_{j+1}] \ .$$

Adversary $\mathcal{B}_j^{\text{Enc},\text{Dec}}$:
$b' \twoheadleftarrow \mathcal{A}^{\text{Init},\text{Enc},\text{Dec}}$
return b'

Init():
if $i = j+1$: $K_i \leftarrow \perp$
else $K_i \twoheadleftarrow \mathcal{K}$

Enc(i, M, N, H):
$C \leftarrow \mathsf{Enc}(K_i, M, N, H)$
if $i \leq j$:
  $C \twoheadleftarrow \{0,1\}^{|C|}$
if $i = j+1$:
  $C \leftarrow \mathcal{B}_j.\text{Enc}(M, N, H)$
return C

Dec(i, C, N, H):
if $i \leq j : M \leftarrow \perp$
if $i = j+1$:
  $M \leftarrow \mathcal{B}_j.\text{Dec}(C, N, H)$
else $M \leftarrow \mathsf{Dec}(K_i, C, N, H)$
return M

Figure 15: Adversary $\mathcal{B}$ in the single-user AE game based on a multi-user AE adversary $\mathcal{A}$.

To this end, we will rely on the underlying security of SE.

Claim. *For any j, and for any adversary $\mathcal{A}$, there exists a $\mathcal{B}_j$ such that*

$$\Pr[\mathsf{Game}_j^{\mathcal{A}}] - \Pr[\mathsf{Game}_{j+1}^{\mathcal{A}}] = \mathbf{Adv}_{\text{SE}}^{\text{ae}}(\mathcal{B}_j) \ .$$

The key observation is that in Game_j, for the first j keys queried, j uniform strings are returned as the ciphertext and for the remaining keys, $n-j$ encryptions are returned. In Game_{j+1}, $j+1$ uniform strings and $n-j-1$ encryptions are returned. Leveraging this difference of one real encryption at the $j+1$ position, we can construct an AE adversary $\mathcal{B}_j$ to emulate the environment of an MUAE adversary $\mathcal{A}$ to break the single user AE security.

Adversary $\mathcal{B}_j$ works as shown in Fig. 15. When the challenge bit b in the AE game (that $\mathcal{B}_j$ is playing) is 1, adversary $\mathcal{A}$'s view of the environment is identical to that in Game_j. When $\mathcal{B}_j$'s challenge bit is $b = 0$, adversary $\mathcal{A}$'s view is identical to that in Game_{j+1}. When $\mathcal{A}$ outputs a bit b', games Game_j and Game_{j+1} check if $b' = 1$. This is equivalent to outputting b', which is also the bit that $\mathcal{B}_j$ outputs. In other words,

$$\mathbf{Adv}_{\text{SE}}^{\text{ae}}(\mathcal{B}_j) = \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] = \Pr[\mathsf{Game}_j^{\mathcal{A}}] - \Pr[\mathsf{Game}_{j+1}^{\mathcal{A}}] \ .$$

□