IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 2, pp. 84–104. DOI:10.13154/tosc.v2017.i2.84-104

Security of Even–Mansour Ciphers under Key-Dependent Messages

Pooya Farshim[1], Louiza Khati[1,2] and Damien Vergnaud[1]

[1] École normale supérieure (ENS), The French National Centre for Scientific Research (CNRS) & INRIA, PSL Research University, Paris, France
pooya.fashim@ens.fr,louiza.khati@di.ens.fr,damien.vergnaud@ens.fr
[2] Oppida, Montigny Le Bretonneux, France

Abstract. The iterated Even–Mansour (EM) ciphers form the basis of many blockcipher designs. Several results have established their security in the CPA/CCA models, under related-key attacks, and in the indifferentiability framework. In this work, we study the Even–Mansour ciphers under key-dependent message (KDM) attacks. KDM security is particularly relevant for blockciphers since non-expanding mechanisms are convenient in setting such as full disk encryption (where various forms of key-dependency might exist). We formalize the folklore result that the ideal cipher is KDM secure. We then show that EM ciphers meet varying levels of KDM security depending on the number of rounds and permutations used. One-round EM achieves some form of KDM security, but this excludes security against offsets of keys. With two rounds we obtain KDM security against offsets, and using different round permutations we achieve KDM security against all permutation-independent claw-free functions. As a contribution of independent interest, we present a modular framework that can facilitate the security treatment of symmetric constructions in models that allow for correlated inputs.

Keywords: Even–Mansour · KDM security · Ideal Cipher · Provable Security.

1 Introduction

1.1 Background

Early on, the seminal paper of Goldwasser and Micali [GM84] pointed out that semantic security may not hold if the adversary gets to see an encryption of the secret key. This practice was generally perceived as a dangerous use of an encryption scheme but several studies have revealed that this security notion is both theoretically and practically important (such as encrypted storage systems such as BitLocker [BHHO08] where the encryption key may be stored in the page file and thus encrypted along with the disk content).

An encryption scheme is said to be *Key-Dependent Message (KDM) secure* if it is secure even against an attacker who can encrypt messages that depend on the secret key. Formally, security is defined with respect to a set Φ of functions ϕ mapping keys to messages for which the adversary can obtain key-dependent encryptions. This security notion was first formalized by Black, Rogaway and Shrimpton [BRS03] for symmetric encryption and was subsequently extensively studied for both symmetric and asymmetric cryptosystems (see, e.g., [BRS03, HK07, HU08, MTY11, DS14, App14, LLJ15]).

A fundamental problem in cryptography is to construct secure blockciphers from simpler primitives. The Even–Mansour (EM) construction introduced in [EM93] is the simplest blockcipher known based on a single public permutation P on n-bit strings. It uses two

Received: 2017-03-01, Accepted: 2017-05-01, Published: 2017-06-19

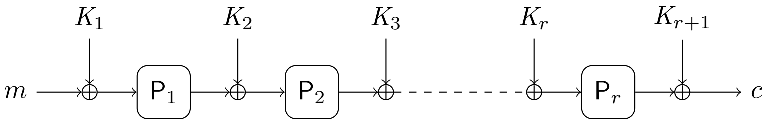

Figure 1: The r-round iterated Even–Mansour cipher.

independent n-bit keys (K_1, K_2) and on input an n-bit plaintext m, it outputs

$$\mathsf{EM}^{\mathsf{P}}((K_1, K_2), m) = K_2 \oplus \mathsf{P}(K_1 \oplus m) \ .$$

Its generalization, the *iterated Even–Mansour* construction (also known as the *key-alternating cipher*) was proposed by Daemen and Rijmen [DR01] as an abstraction of the design paradigm of substitution-permutation networks. Given r permutations $\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_r$ on n-bit strings and $(r+1)$ keys of length n, the r-round iterated EM construction, given as input an n-bit plaintext m, outputs

$$\mathsf{EM}^{\mathsf{P}_1,\ldots,\mathsf{P}_r}((K_1,\ldots,K_{r+1}), m) = K_{r+1} \oplus \mathsf{P}_r(K_r \oplus \mathsf{P}_{r-1}(\cdots(K_3 \oplus \mathsf{P}_2(K_2 \oplus \mathsf{P}_1(K_1 \oplus m)))\cdots)) \ .$$

This construction has become the object of abundant analysis and many recent blockciphers follow this design (e.g., Present [BKL+07] and PRINCE [BCG+12]).

If one models the underlying permutations as public random permutations, it is sometimes possible to prove the nonexistence of generic attacks against the iterated Even–Mansour construction (i.e., attacks that are possibly independent of a particular instantiation of the permutations $\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_r$). If the adversary is only given black-box oracle access to these random permutations, the iterated Even–Mansour cipher was proved to achieve several security notions such as traditional indistinguishability (see [CLL+14] and references therein), security against related-key attacks [FP15, CS15], security in the multi-user setting [ML15, HT16] or indifferentiability from ideal ciphers[1] (see [DSST17] and references therein).

In this paper, we continue this line of work and study the iterated Even–Mansour ciphers under key-dependent message attacks.

1.2 Contributions

Our main technical contribution is the proposal of a new modular framework to analyze the KDM security (and possibly also other forms of security under correlated inputs) for blockciphers. Our approach is to start with a blockcipher and gradually modify its oracles with independent ones until we arrive at a construction whose outputs are uniformly and independently distributed. In the particular case of Even–Mansour ciphers, we will replace at most two of the underlying permutations (namely P_1 and P_r) with oracles that completely randomize the outputs of the cipher (in both directions for decryption and encryption queries respectively); see Figure 2.

We consider a general security game where an adversary $\mathcal{A}$ has access to an oracle through two different interfaces. The approach consists in studying the conditions under which the security game can be modified (in an indistinguishable way for $\mathcal{A}$) so that the second interface provides access to an independent instance of this oracle. We also analyze the conditions under which this oracle can further be replaced by a *forgetful* oracle that completely removes dependency of outputs on inputs. For KDM security, we then have to prove that this replacement by forgetful oracles (after splitting) can be performed

[1] This security notion roughly ensures that the construction "behaves" in some well-defined sense as an ideal cipher.

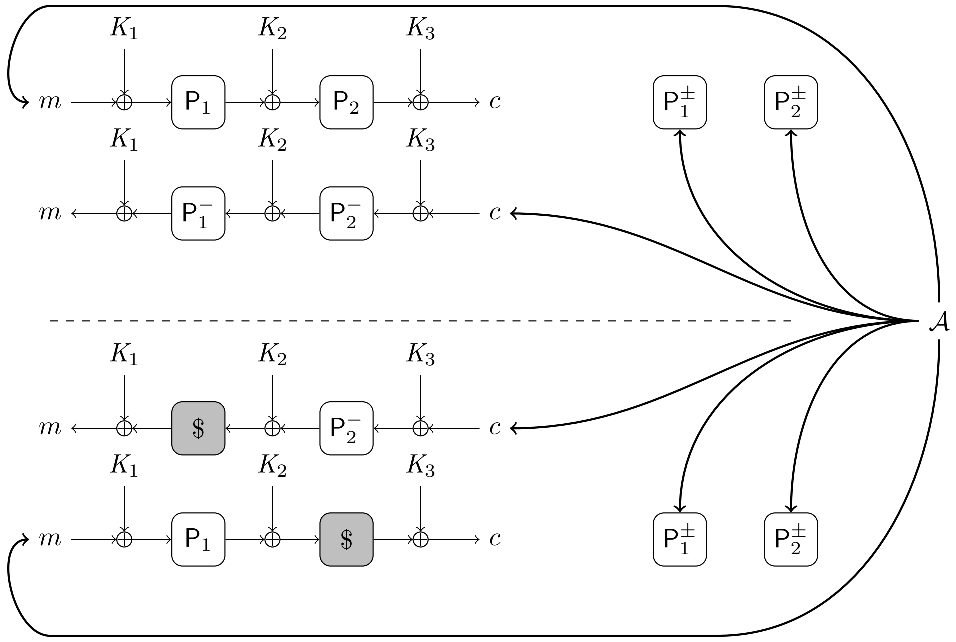

Figure 2: KDM-CCA analysis for 2-round Even–Mansour.

indistinguishably if the set of key-dependent messages functions that the adversary has at its disposal satisfy certain well-defined conditions. These conditions reduce to checking that the adversary $\mathcal{A}$ does not query these oracles on the same inputs (in the backward and forward direction in the case of random permutations). This general technique allows us to analyze the r-round iterated EM construction in a unified way. It is potentially applicable for any number of rounds r but here we only apply to three EM cases with $r = 1, 2$ (the latter with(out) permutation reuse) and also the KDM security of the ideal cipher.

We first show that our *"splitting and forgetting"* technique is applicable to analyze the KDM security of the ideal cipher. Halevi and Krawczyk [HK07] prove that the ideal cipher achieves KDM security if one restricts the function class Φ to be a singleton and containing a function that is independent of the ideal cipher itself. Using our strategy, we can prove the KDM security of the ideal cipher against adversaries with significantly larger classes of KDM functions, including functions that may depend on the ideal cipher. In the particular case where the functions are independent of the ideal cipher itself, it is sufficient to assume that the set of functions is *claw-free*, i.e., when distinct functions disagree on random inputs.

We then analyze the KDM security of the 1-round EM construction in the random-permutation model. We consider only sets of functions that are independent of underlying permutation (but our method can be extended to handle functions that depend on the underlying random permutation). We first present a simple attack that excludes the practically relevant case of KDM security with respect to the identity function (and more generally any offset of the key). On the positive side, we prove using our framework that the 1-round EM construction actually achieves KDM security under chosen-ciphertext attacks if the set of functions available to the attacker is claw-free and *offset-free*, i.e., when functions do not offset the key by a constant.

We apply the above method to study the KDM security of the 2-rounds EM construction in two configurations. We present a simple *slide attack* [BW99] on a variant with both permutation and key reuse where $K_1 = K_2 = K_3$ and $\mathsf{P}_1 = \mathsf{P}_2$ are used within the construction. The set of KDM functions considered contains the identity function (or more generally any key offsets). We also present a simple attack with complexity $2^{n/2}$ on the most general version. We then apply the framework to prove that 2-round EM achieves KDM security under chosen-ciphertext attacks if the set of functions available to the attacker is only claw-free as long as different permutations are used. When one reuses the same permutation $\mathsf{P}_1 = \mathsf{P}_2$ (because of efficiency reasons or because only one "good" public permutation is available), we prove that EM achieves KDM-CCA security if the set of functions available to the attacker is claw-free and also *offset-xor-free*, meaning that functions do not output offsets of xor of two of the keys.

Our framework is general enough to be applied to other symmetric constructions and/or other security models. Indeed, we believe this approach can be used to re-derive the RKA security of EM ciphers [FP15] or that for Feistel networks [BF15] in a more modular way.

2 Preliminaries

Notation. We let $\mathbb{N} := \{0, 1, \dots\}$ denote the set of non-negative integers, and $\{0,1\}^*$ denote 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 decodable encoding of X and Y. The length of a string X is denoted by $|X|$. By $x \twoheadleftarrow S$ we mean sampling x uniformly from set S. All lists are initialized to empty and all bad flags to false. Throughout \$ denotes a *forgetful* oracle over some domain and range that on each input in the domain (even repeated ones) returns a uniformly chosen random element from the range. For a deterministic oracle machine $M^{\mathcal{O}}$ we denote by $\mathbf{Q}(M^{\mathcal{O}}(x))$ the list of query/answer pairs made to and received from $\mathcal{O}$ when M is run on input x. For a list L of pairs (x, y), which may have repeats, we denote by $\mathsf{Dom}(L)$ the list of first entries x and by $\mathsf{Rng}(L)$ the list of second entries. 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].

Blockciphers. Given two non-empty subsets $\mathcal{K}$ and $\mathcal{M}$ of $\{0,1\}^*$, called the key space and the message space respectively, we let $\mathrm{Block}(\mathcal{K}, \mathcal{M})$ denote the set of all functions $\mathsf{E} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{M}$ 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 $\mathsf{D} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{M}$. A blockcipher for key space $\mathcal{K}$ and message space $\mathcal{M}$ is a triple of efficient algorithms $\mathrm{BC} := (\mathsf{K}, \mathsf{E}, \mathsf{D})$ such that $\mathsf{E} \in \mathrm{Block}(\mathcal{K}, \mathcal{M})$ and its inverse is D. In more detail, K is the randomized key-generation algorithm which returns a key $K \in \mathcal{K}$. Typically $\mathcal{K} = \{0,1\}^k$ for some $k \in \mathbb{N}$ called the key length, and K endows it with the uniform distribution. Algorithm E is the deterministic enciphering algorithm with signature $\mathsf{E} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{M}$. Typically $\mathcal{M} = \{0,1\}^n$ for some $n \in \mathbb{N}$ called the block length. (3) D is the deterministic deciphering algorithm with signature $\mathsf{D} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{M}$. Thus a blockcipher is correct in the sense that for all $K \in \mathcal{K}$ and all $M \in \mathcal{M}$ we have that $\mathsf{D}(K, \mathsf{E}(K, M)) = M$. It is also length preserving. (Note that length preservation follows from correctness if $\mathcal{M} = \{0,1\}^n$). A permutation on $\mathcal{M}$ is simply a blockcipher with key space $\mathcal{K} = \{\varepsilon\}$. We denote a permutation with P and its inverse with P^-. A permutation can be trivially obtained from blockcipher (by fixing the key). For a blockcipher $\mathrm{BC} := (\mathsf{E}, \mathsf{D})$, notation $\mathcal{A}^{\mathrm{BC}}$ denotes oracle access to both E and D for $\mathcal{A}$. We abbreviate $\mathrm{Block}(\{0,1\}^k, \{0,1\}^n)$ by $\mathrm{Block}(k, n)$ and $\mathrm{Block}(\{\varepsilon\}, \{0,1\}^n)$ by $\mathrm{Perm}(n)$.

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) with key space $\mathcal{K}$ and message space $\mathcal{M}$ is a model of computation where all parties, honest or otherwise, have oracle access to a uniformly chosen random element in $\mathrm{Block}(\mathcal{K}, \mathcal{M})$ together with its inverse. The ideal-cipher model when restricted to $\mathcal{K} = \{\varepsilon\}$ gives rise to the random-permutation model (RPM).

Even–Mansour ciphers. The (iterated) Even–Mansour ciphers consider the problem of constructing a blockcipher with a large key space from a single, or a small number of, permutations. Formally, the r-round Even–Mansour cipher in a model of computation with r permutations $\mathsf{P}_1^{\pm}, \ldots, \mathsf{P}_r^{\pm}$ with domain $\mathcal{M} = \{0,1\}^n$ is a blockcipher with key space $\mathcal{K} = \{0,1\}^{(r+1)n}$ and enciphering and deciphering algorithms

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

The EM ciphers can be also 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 $\mathsf{EM}^{\mathsf{P}_1,\ldots,\mathsf{P}_r}[K_1, K_2, \ldots, K_{r+1}]$.

KDM functions. A key-dependent-message (KDM) function/circuit for key space $\mathcal{K}$ and message space $\mathcal{M}$ is a deterministic and stateless circuit $\phi : \mathcal{K} \longrightarrow \mathcal{M}$. A KDM set Φ is simply a set of KDM functions ϕ on the same key and message spaces. We assume membership in KDM sets can be efficiently decided. An *oracle* KDM function $\phi^{\mathcal{O}} : \mathcal{K} \longrightarrow \mathcal{M}$ is a KDM function with oracle gates.

KDM security. We now formalize security of blockciphers under key-dependent message and chosen-ciphertext attacks (KDM-CCA). We do this in the $\mathcal{O}$-hybrid model of computation where oracle access to $\mathcal{O}$ sampled from some oracle space OSp is granted to all parties. For example, in the context of Even–Mansour ciphers, $\mathcal{O}(i, x, \sigma \in \{\pm\}) := \mathsf{P}_i^{\sigma}(x)$ for some random permutations $\mathsf{P}_i^{\pm}$. We therefore grant access to $\mathcal{O}$ to the KDM functions and the adversary. Security is now defined in the standard way via indistinguishability from the ideal cipher under a random key as shown in Figure 3.

```
Game KDM-CCA$^{\mathcal{A},\Phi}_{\mathrm{BC}^{\mathcal{O}}}$
  $\mathcal{O} \twoheadleftarrow \mathrm{OSp}$
  $L \leftarrow [\,]$
  $b \twoheadleftarrow \{0,1\}$
  $K \twoheadleftarrow \mathcal{K}$
  $(i\mathsf{E}, i\mathsf{D}) \twoheadleftarrow \mathrm{Block}(\mathcal{K}, \mathcal{M})$
  $b' \twoheadleftarrow \mathcal{A}^{\mathcal{O},\mathrm{KDMEnc},\mathrm{Dec}}$
  Return $(b' = b)$

Proc. KDMEnc($\phi^{\mathcal{O}}$)
  If $\phi^{\mathcal{O}} \notin \Phi$ Return $\perp$
  $M \leftarrow \phi^{\mathcal{O}}(K)$; $C \leftarrow \mathsf{E}^{\mathcal{O}}(K, M)$
  If $b = 1$ Then $C \leftarrow i\mathsf{E}(K, M)$
  $L \leftarrow L : C$; Return $C$

Proc. Dec($C$):
  If $C \in L$ Return $\perp$
  If $b = 1$ Return $i\mathsf{D}(K, C)$
  Return $\mathsf{D}^{\mathcal{O}}(K, C)$
```

Figure 3: Game defining Φ-KDM-CCA security for a blockcipher.

The adversary can ask for key-dependent encryption for functions $\phi^{\mathcal{O}} \in \Phi$ and decryption of ciphertexts of its choice.[2] To allow for expressive KDM sets and rule out trivial attacks, we do not allow decryption of ciphertexts that were obtained from the encryption oracle (as otherwise the key can be recovered by decrypting key-dependent ciphertexts). Given blockcipher $\mathrm{BC}^{\mathcal{O}}$ and an oracle KDM set Φ, we define the advantage of an adversary $\mathcal{A}$ against $\mathrm{BC}^{\mathcal{O}}$ with respect to Φ as

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathrm{BC}^{\mathcal{O}}}(\mathcal{A}, \Phi) := 2 \cdot \Pr\left[\text{KDM-CCA}^{\mathcal{A},\Phi}_{\mathrm{BC}^{\mathcal{O}}}\right] - 1 \ .$$

[2] Note that we do not allow for key-dependent ciphertexts (KDC) in this work as the practical motivations are somewhat limited.

Feasibility of Φ-KDM-CCA security very much depends on the KDM functions available in Φ. For instance, if Φ contains the constant functions only, we recover the standard (strong) PRP notion of security, which is feasible under standard assumptions, in the RPM. On the other hand, the set Φ cannot be arbitrary. Consider for instance functions ϕ_i that zero all bits of K except the i-th one. Then using encryptions of $\phi_i(K)$ as well as those for $M_{b,i} := 1^{i-1}|b|0^{n-i}$ for $i = 1, \ldots, n$ and $b \in \{0,1\}$ once can recover the key one bit at a time. For other sets, however, feasibility may or may not be possible. In the coming sections, we study this question for the EM ciphers.

3 Analysis via Forgetful Oracle Replacement

Our strategy to prove KDM security for blockciphers is to gradually modify their internals until we arrive at constructions whose outputs are uniformly and independently distributed. For instance, in the case of EM ciphers we will replace one or more of their underlying permutations with *forgetful* random oracles. These will completely randomize the outputs of the cipher. To argue that this replacement can be performed indistinguishably, we will impose certain restrictions on how the adversary can interact with the cipher, and in particular the set of KDM functions at its disposal will be restricted.

In this section we present a more general result that comes with a number of advantages: (1) it allows reusing parts of the analyses across different constructions, (2) it highlights the overall proof strategy and how various assumptions are used with it, and (3) it is potentially applicable to setting beyond KDM security, and/or to other constructions.

3.1 A framework for security analyses

The blockciphers that we analyze are constructed in a model of computation where all parties have access to some oracle $\mathcal{O}$.[3] These oracles will be sampled from some oracle space OSp. We start with two assumptions on oracles that are of interest to us.

Splittability. Let $\mathbf{sp}(L_1, L_2)$ be a binary relation on lists L_1 and L_2. We say oracle $\mathcal{O}$ *splits* under **sp** if access to $\mathcal{O}$ through two interfaces can be modified in an indistinguishable way so that the second interface provides access to an independent instance $\mathcal{O}'$. Formally, we define the advantage of $\mathcal{D}$ in the split game as

$$\mathbf{Adv}^{\text{split}}_{\text{OSp}}(\mathcal{D}) := 2 \cdot \Pr\left[\text{Split}^{\mathcal{D}}_{\text{OSp}}\right] - 1 \ ,$$

where game $\text{Split}^{\mathcal{D}}_{\text{OSp}}$ is shown in Figure 4.

```
Game $\text{Split}^{\mathcal{D}}_{\text{OSp}}$
  $\mathcal{O}, \mathcal{O}' \twoheadleftarrow \text{OSp}$
  $b \twoheadleftarrow \{0,1\}$
  $b' \twoheadleftarrow \mathcal{D}^{\text{O},\textsc{Chal}}$
  If $\mathbf{sp}(L_1, L_2)$ Then $b' \leftarrow 0$
  Return $(b = b')$
```

```
Proc. $\text{O}(x)$
  $y \leftarrow \mathcal{O}(x)$
  $L_1 \leftarrow L_1 : (x, y)$
  Return $y$
```

```
Proc. $\textsc{Chal}(x)$
  If $b = 0$ Then $y \twoheadleftarrow \mathcal{O}(x)$
  Else $y \leftarrow \mathcal{O}'(x)$
  $L_2 \leftarrow L_2 : (x, y)$
  Return $y$
```

Figure 4: Game defining oracle splittability with respect to relation **sp**.

An alternative definition would quantify over all $\mathcal{D}$ that do not trigger **sp**. Although **sp** is publicly checkable, this does not necessarily mean that every $\mathcal{D}$ can be modified to one with comparable advantage that never triggers **sp**: the relation also depends on oracle outputs, which are outside the control of the distinguisher. Although, relations that we

[3] Access to multiple oracles can be modeled via domain separation.

study here have the extra property that $\mathcal{D}$ *can* be modified to avoid triggering **sp**, not all $\mathcal{D}$ will be able to perform this check. In particular certain *two-stage* distinguishers $\mathcal{D}$ cannot check for **sp** as the information needed for this check is spread among its two stages. For such $\mathcal{D}$ we need to keep **sp** in the game description.

Forgetful switching. We define the advantage of an algorithm $\mathcal{D}$ in the switch game as

$$\mathbf{Adv}^{\text{forget}}_{\text{OSp}}(\mathcal{D}) := 2 \cdot \Pr\left[\text{Forget}^{\mathcal{D}}_{\text{OSp}}\right] - 1 \ ,$$

where game $\text{Forget}^{\mathcal{D}}_{\text{OSp}}$ is formalized in Figure 5. Relation **fg** in this game will typically check for some form of repetitions in oracle queries. Replacing an oracle with a forgetful one removes any dependency of outputs on inputs.

```
Game $\text{Forget}^{\mathcal{D}}_{\text{OSp}}$              Proc. $\textsc{Chal}(x)$
  $\mathcal{O} \twoheadleftarrow \text{OSp}; b \twoheadleftarrow \{0,1\}$      If $b = 0$ Then $y \twoheadleftarrow \mathcal{O}(x)$
  $b' \twoheadleftarrow \mathcal{D}^{\textsc{Chal}}$                   Else $y \leftarrow \$(x)$
  If $\mathbf{fg}(L)$ Then $b' \leftarrow 0$          $L \leftarrow L : (x, y)$
  Return $(b = b')$                   Return $y$
```

Figure 5: Game defining forgetful switching property.

Consider now a modified split game (m-Split) that (1) totally drops the **sp** check and (2) when $b = 1$ uses a forgetful oracle \$ in place of $\mathcal{O}'$ under the Chal procedure. We have the following result.

Theorem 1. *Let* OSp *be a lazily samplable oracle. Let* **fg** *(resp.* **sp***) be, by slight abuse of notation, the event that $\mathcal{D}$ triggers the check* **fg** *(resp.* **sp***) in the m-Split game. Then for any $\mathcal{D}$ in the modified split game we have a $\mathcal{D}'$ such that*

$$\mathbf{Adv}^{\text{m-split}}_{\text{OSp}}(\mathcal{D}) \leq \mathbf{Adv}^{\text{split}}_{\text{OSp}}(\mathcal{D}) + 2 \cdot \mathbf{Adv}^{\text{forget}}_{\text{OSp}}(\mathcal{D}') + \Pr[\mathcal{D} \textit{ sets } \mathbf{fg} | b = 1] + \Pr[\mathcal{D} \textit{ sets } \mathbf{sp} | b = 1] \ .$$

Proof. The proof follows 6 games hops and applies the the fundamental lemma of game playing as follows. Below, we let G_i be the event that $\mathcal{D}$ outputs $b' = 0$ in game i.

Game$_0$: This is the $\text{m-Split}^{\mathcal{D}}_{\text{OSp}}$ game with $b = 0$ (i.e., with respect to oracles $(\mathcal{O}, \mathcal{O})$): $\Pr[G^{\mathcal{D}}_0] = \Pr[\text{m-Split}^{\mathcal{D}}_{\text{OSp}} | b = 0]$.

Game$_1$: In this game we introduce the **sp** check. This only increases the probability that $b' = 0$: $\Pr[G^{\mathcal{D}}_0] - \Pr[G^{\mathcal{D}}_1] \leq 0$.

Game$_2$: In this game we use an independent oracle $\mathcal{O}'$ for the challenge oracle. A direct reduction shows that: $\Pr[G^{\mathcal{D}}_1] - \Pr[G^{\mathcal{D}}_2] \leq \mathbf{Adv}^{\text{split}}_{\text{OSp}}(\mathcal{D})$.

Game$_3$: In this game we drop the **sp** check. This game and the previous one are identical until a flag $\mathbf{sp}_3$ corresponding to check **sp** is set: $\Pr[G^{\mathcal{D}}_2] - \Pr[G^{\mathcal{D}}_3] \leq \Pr[\mathcal{D} \text{ sets } \mathbf{sp}_3 | b = 1]$.

Game$_4$: In this game we introduce the **fg** check. This only increases the probability that $b' = 0$: $\Pr[G^{\mathcal{D}}_3] - \Pr[G^{\mathcal{D}}_4] \leq 0$.

Game$_5$: In this game we use a forgetful oracle \$ for the challenge oracle. Since $\mathcal{O}$ is assumed to be lazily samplable (and $\mathcal{O}$ and \$ are independent of $\mathcal{O}'$) via a direct reduction and simulation of $\mathcal{O}$ we get that for some $\mathcal{D}'$: $\Pr[G^{\mathcal{D}}_4] - \Pr[G^{\mathcal{D}}_5] \leq \mathbf{Adv}^{\text{forget}}_{\text{OSp}}(\mathcal{D}')$.

Game$_6$: Finally, we drop the **fg** check. Not that this game is identical to m-Split$^{\mathcal{D}}_{\text{OSp}}$ game with $b = 1$. This game and the previous one are identical until a flag $\mathbf{fg}_6$ is set: $\Pr[G^{\mathcal{D}}_5] - \Pr[G^{\mathcal{D}}_6] \leq \Pr[\mathcal{D} \text{ sets } \mathbf{fg}_6 | b = 1]$.

We now bound the probability that $\mathcal{D}$ sets $\mathbf{sp}_3$ when $b = 1$. Let $\mathbf{sp}_i$ be the flag analogous to $\mathbf{sp}_3$ in Game i. Note that the probability of $\mathbf{sp}_3$ and that of $\mathbf{sp}_4$ are the same as checking condition **fg** has no effect on setting these flags. Using the fact that **sp** is publicly checkable (and hence can be used to define a distinguisher) we get that for some $\mathcal{D}''$

$$\Pr[\mathcal{D} \text{ sets } \mathbf{sp}_4 | b = 1] - \Pr[\mathcal{D} \text{ sets } \mathbf{sp}_5 | b = 1] \leq \mathbf{Adv}^{\text{forget}}_{\text{OSp}}(\mathcal{D}'').$$

Finally, the probability of setting $\mathbf{sp}_6$ is identical to that of $\mathbf{sp}_5$ as, once again, **fg** has no effect on these flags. The theorem now follows by adding the above inequities. □

We now consider a class of *two-stage* adversaries $\mathcal{D} = (\mathcal{A}, \mathcal{B})$. Adversary $\mathcal{A}$ can access the first oracle interface directly: this models the public availability of the oracle. Its access to the second interface, however, is restricted and is through algorithm $\mathcal{B}$ only. This algorithm holds information K unavailable to $\mathcal{A}$. It receives messages z from $\mathcal{A}$ and returns an output after interacting with the oracles through two interfaces. Formally, we say $\mathcal{D}$ is two stage if it can be written in the form shown in Figure 6 (left) for some algorithms $\mathcal{A}$ and $\mathcal{B}$. The operation of $\mathcal{D} = (\mathcal{A}, \mathcal{B})$ in the split game is shown on the right. Although algorithm $\mathcal{A}$ can be typically arbitrary, we will put restrictions on the operation of $\mathcal{B}$. For example, in the KDM setting $\mathcal{A}$ will correspond to the KDM adversary and $\mathcal{B}$ will model the operation of a blockcipher on key-dependent messages. More concretely, for 1-round EM:

$$\mathcal{B}^{\text{O},\textsc{Chal}}(K = (K_1, K_2), z = \phi) := \mathsf{EM}^{\text{O},\textsc{Chal}}[K_1, K_2](\phi^{\text{O}}(K_1, K_2)) \ .$$

We also assume that algorithm $\mathcal{B}$ is *stateless*; that is, it does not store any local state and each time is run afresh on K and the incoming input z.[4] This means each instance of $\mathcal{B}(K, z_i)$ can be run independently. We also assume $\mathcal{B}$ is deterministic, and hence also that $\mathcal{A}$ queries $\mathcal{B}$ with distinct inputs z. Finally, we assume that $\mathcal{B}$ has simulatable outputs: its outputs on a random K and any z are indistinguishable from \$ when it is run with respect to oracles $(\mathcal{O}, \$)$.

```
Algo. $\mathcal{D}^{\text{O},\textsc{Chal}}$
  $K \twoheadleftarrow \mathcal{K}$
  $b' \twoheadleftarrow \mathcal{A}^{\text{O},\text{B}}$
  Return $b'$

Proc. B($z$)
  Return $\mathcal{B}^{\text{O},\textsc{Chal}}(K, z)$
```

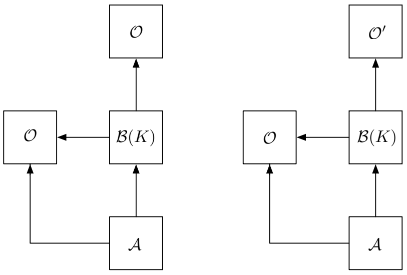

Figure 6: Two-stage adversaries and their operation in the split game.

We now consider the probability of setting **sp** or **fg** in m-Split for $\mathcal{D}$ that take the above form. We consider a setting where **sp** and **fg** can be expressed as disjunctions of simpler checks on pairs of distinct entries from the lists. More precisely, we assume for some algorithm **val**:

$$\mathbf{sp}(L_1, L_2) := \bigvee_{i,j} \mathbf{val}(L_1[i], L_2[j]) \quad \text{and} \quad \mathbf{fg}(L_2) := \bigvee_{i \neq j} \mathbf{val}(L_2[i], L_2[j]) \qquad (\star)$$

[4] Any $(\mathcal{A}, \mathcal{B})$ with a stateful $\mathcal{B}$ can be modified to $(\mathcal{A}', \mathcal{B}')$ with stateless $\mathcal{B}'$ and an $\mathcal{A}'$ that sends the entire history of previous messages to $\mathcal{B}'$. This allows $\mathcal{B}'$ to recompute the state of $\mathcal{B}$. This modification however increases the query complexity of $\mathcal{B}$, and might not preserve other properties required from $\mathcal{B}$.

where $L[i]$ denotes the i-th element of the list L (which may contain repeats). Each clause depends on at most 2 elements. Hence a clause can be set by two entries corresponding to one of the following cases.

Sp1 : A direct $\mathcal{O}$ query of $\mathcal{A}$ and a challenge query of $\mathcal{B}(K, z_1)$ for some z_1.

Sp2 : An $\mathcal{O}$ query of $\mathcal{B}(K, z_1)$ and a challenge query of $\mathcal{B}(K, z_2)$ (possibly with $z_1 = z_2$).

Fg : Two challenge queries made by $\mathcal{B}(K, z_1)$ and $\mathcal{B}(K, z_2)$ with $z_1 \neq z_2$.

Therefore, triggering events Sp1 and Sp2 is equivalent to triggering **sp** and Fg is equivalent to **fg**. Note, for a $\mathcal{B}$ that does not place any $\mathcal{O}$ calls, event Sp2 never happens.

3.2 Some concrete cases

To applying the above theorem to KDM attacks, we start by observing that random oracles H, permutations $\mathsf{P}^{\pm}$ and ideal cipher $\mathsf{E}^{\pm}$ can be expressed as a single oracle $\mathcal{O}$ via encodings

$$\mathcal{O}(x) := \mathsf{H}(x) \ , \qquad \mathcal{O}(\sigma, x) := \mathsf{P}^{\sigma}(x) \ , \qquad \mathcal{O}(\sigma, k, x) := \mathsf{E}^{\sigma}(k, x) \ .$$

Using the standard PRP/PRF switching lemma [BR06] these oracles enjoy forgetful switching with respect to checks $\mathbf{fg}_{\mathrm{ro}}$, $\mathbf{fg}_{\mathrm{rp}}$, and $\mathbf{fg}_{\mathrm{ic}}$ defined via Equations ($\star$) and

$$\begin{aligned} \mathbf{val}_{\mathrm{ro}}((x_1, y_1), (x_2, y_2)) &:= (x_1 = x_2) \ , \\ \mathbf{val}_{\mathrm{rp}}((\sigma_1, x_1, y_1), (\sigma_2, x_2, y_2)) &:= (\sigma_1 {=} \sigma_2 \wedge x_1 {=} x_2) \vee (\sigma_1 {\neq} \sigma_2 \wedge (x_1 {=} y_2 \vee x_2 {=} y_1)) \ , \\ \mathbf{val}_{\mathrm{ic}}((\sigma_1, k_1, x_1, y_1), (\sigma_2, k_2, x_2, y_2)) &:= (k_1 = k_2) \wedge \mathbf{val}_{\mathrm{rp}}((\sigma_1, x_1, y_1), (\sigma_2, x_2, y_2)) \ . \end{aligned}$$

Note that these conditions are publicly checkable. The advantage terms for q-query adversaries $\mathcal{D}$ and domain size 2^n are

$$\mathbf{Adv}^{\mathrm{forget}}_{\mathrm{Perm}(n)}(\mathcal{D}) \leq q^2/2^n \qquad \text{and} \qquad \mathbf{Adv}^{\mathrm{forget}}_{\mathrm{Block}(k,n)}(\mathcal{D}) \leq q^2/2^n \ .$$

These oracles also split with respect to $\mathbf{sp}_{\mathrm{ro}}$, $\mathbf{sp}_{\mathrm{rp}}$, and $\mathbf{sp}_{\mathrm{ic}}$ associated to their respective **val** above. This is immediate for random oracles (with advantage zero) as the systems (H, H) and $(\mathsf{H}, \mathsf{H}')$ are identical as long as the two interfaces are not queried on the same input. Splitting for ideal ciphers, and random permutations where $K = \varepsilon$, is proved easily.

Theorem 2 (Splitting for the ideal cipher)**.** *For any adversary $\mathcal{A}$ making at most q_1 queries to its first oracle and q_2 queries to its second oracle we have that*

$$\mathbf{Adv}^{\mathrm{split}}_{\mathrm{Block}(k,n)}(\mathcal{D}) \leq \frac{q_1 q_2}{2^n} \ .$$

Proof. Consider an adversary $\mathcal{D}$ with oracle access to $\mathcal{O}_1$ and $\mathcal{O}_2$. Algorithm $\mathcal{D}$ cannot ask the same query to its two oracles and it cannot ask to decrypt or encrypt a query to $\mathcal{O}_1$ that has been queried or obtained to $\mathcal{O}_2$ and inversely. $\mathcal{D}$ has to distinguish between two systems (E, E) and $(\mathsf{E}, \tilde{\mathsf{E}})$ where E and $\tilde{\mathsf{E}}$ are two independent ideal ciphers. After the attack $\mathcal{D}$ ends up with two lists L_1 and L_2 containing, respectively, the q_1 queries made to $\mathcal{O}_1$ and the q_2 queries made to $\mathcal{O}_2$. The only event that can enable $\mathcal{D}$ to trigger **sp** is an entry $(\sigma_1, k_1, x_1, y_1) \in L_1$ and another $(\sigma_2, k_2, x_2, y_2) \in L_2$ such that $(\sigma_1 \neq \sigma_2 \wedge (x_1 = y_2 \vee x_2 = y_1))$. The probability of this event is bounded by $q_1 q_2 / 2^n$. □

Relation between splitting and switching. Any oracle with forgetful replacement also allows for splitting: start with $(\mathcal{O}, \mathcal{O})$, replace *both* oracles to get $(\$, \$)$ and now switch the first oracle back to get $(\mathcal{O}, \$)$. This reduction, however, restricts the class of attacks that can be considered. Indeed in this reduction we would need to rely on $\mathbf{fg}(L_1 : L_2) \vee \mathbf{fg}(L_1)$ which imposes no repeat queries to the first oracle. This oracle is also used by $\mathcal{B}$, and hence we would have to assume that it does not place repeated queries to it. It might appear that this is not a problem as "without loss of generality" such repeat queries can be dealt with using lists. This, however, is not the case as different instances of $\mathcal{B}$ often cannot freely communicate with each other their *local* lists.

4 KDM Security of the Ideal Cipher

The KDM security for an ideal cipher is formulated as in Figure 3 with respect to an oracle $\mathcal{O}$ that implements an ideal cipher and a trivial construction $\mathrm{BC}^{\mathcal{O}}$ that simply uses $\mathcal{O}$ to encipher and decipher inputs. We formulate a set of sufficient conditions on a KDM set Φ that allows us to establish KDM security for the ideal cipher.

Claw-freeness. We define the (single-try) claw-freeness advantage of $\mathcal{A}$ against a KDM set Φ as

$$\mathbf{Adv}^{\mathrm{cf}}_{\mathrm{OSp},\Phi}(\mathcal{A}) := \Pr[\phi_1^{\mathcal{O}} \neq \phi_2^{\mathcal{O}} \wedge \phi_1^{\mathcal{O}}(K) = \phi_2^{\mathcal{O}}(K) : \\ \mathcal{O} \twoheadleftarrow \mathrm{OSp}; K \twoheadleftarrow \{0,1\}^k; (\phi_1^{\mathcal{O}}, \phi_2^{\mathcal{O}}) \twoheadleftarrow \mathcal{A}^{\mathcal{O}}] .$$

We define the (multi-try) claw-freeness advantage $\mathbf{Adv}^{\mathrm{mcf}}_{\mathrm{OSp},\Phi}(\mathcal{A})$ by considering $\mathcal{A}$ that return two *lists* of sizes q_1 and q_2 and claws are checked for two distinct ϕ's coming from the two lists. A simple guessing arguments shows that for any multi-try $\mathcal{A}$ there is a single-try $\mathcal{A}'$ such that:

$$\mathbf{Adv}^{\mathrm{mcf}}_{\mathrm{OSp},\Phi}(\mathcal{A}) \leq q_1 q_2 \cdot \mathbf{Adv}^{\mathrm{cf}}_{\mathrm{OSp},\Phi}(\mathcal{A}') .$$

Informally, Φ is claw-free if the above advantage is "small" for every "reasonable" $\mathcal{A}$. When the KDM function are independent of $\mathcal{O}$ we may omit sampling of OSp from the game and notation.

The KDM set corresponding to xoring constants into the key:

$$\Phi^{\oplus} := \{\phi_i[\Delta] : (K_1, \ldots, K_{r+1}) \mapsto K_i \oplus \Delta : \Delta \in \mathcal{K}\} \cup \{(K_1, \ldots, K_{r+1}) \mapsto \Delta : \Delta \in \mathcal{K}\} .$$

is claw-free since the probability that $K_i \oplus \Delta_1 = K_j \oplus \Delta_2$ is 0 if $i = j$ and $\Delta_1 \neq \Delta_2$, and is negligible if $i \neq j$.

Query-independence. We define the query-independence advantage of $\mathcal{A}$ against a KDM set Φ with respect to oracle space $\mathrm{OSp} := \mathrm{Block}(k, n)$ as

$$\mathbf{Adv}^{\mathrm{qi}}_{\mathrm{Block}(k,n),\Phi}(\mathcal{A}) := \Pr\left[\phi_1^{\mathcal{O}}(K) \in \mathbf{Q}^{+}_{K}(\phi_2^{\mathcal{O}}(K)) \text{ or } C \in \mathbf{Q}^{-}_{K}(\phi_2^{\mathcal{O}}(K)) : \\ \mathcal{O} \twoheadleftarrow \mathrm{Block}(k,n); K \twoheadleftarrow \{0,1\}^k; (C, \phi_1^{\mathcal{O}}, \phi_2^{\mathcal{O}}) \twoheadleftarrow \mathcal{A}^{\mathcal{O}}\right] .$$

Here we have used the convention $\mathcal{O}(\sigma, K, M) := \mathsf{E}^{\sigma}(K, M)$. Note that any oracle-free KDM set is query-independent (*i.e.* has zero query-independence advantage).

We now prove that the ideal cipher is KDM secure for claw-free and query-independent KDM sets.

Theorem 3 (Ideal cipher KDM security)**.** *Let Φ be a KDM set for keys of length k and messages of length n. Suppose Φ is claw-free and query-independent as defined above. Then the ideal cipher is* Φ-KDM-CCA *secure. More precisely, for any adversary $\mathcal{A}$ against the* Φ-KDM-CCA *security of the ideal cipher for* $\mathrm{Block}(k,n)$*, there is an adversary $\mathcal{C}_1$ against the claw-freeness of Φ and an adversary $\mathcal{C}_2$ against the query-independence of Φ such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathrm{Block}(k,n),\Phi}(\mathcal{A}) \leq q_1 q/2^n + 2q^2/2^n + q_1 q/2^k + q^2(\mathbf{Adv}^{\mathrm{cf}}_{\mathrm{Block}(k,n),\Phi}(\mathcal{C}_1) + 2/2^n) + \\ + 2q^2(\mathbf{Adv}^{\mathrm{qi}}_{\mathrm{Block}(k,n),\Phi}(\mathcal{C}_2) + q_\phi/2^n) .$$

Here q_1 is an upper bound on the number of direct queries of $\mathcal{A}$ to the ideal cipher (in either direction), q an upper bound on the number of challenge queries (globally), and q_ϕ an upper bound on the number of oracle queries of KDM functions. Adversaries $\mathcal{C}_1$ and $\mathcal{C}_2$ place at most q_1 queries to their ideal cipher oracles.

Proof. Let $\mathcal{A}'$ be a KDM-CCA adversary. We consider a two-stage adversary $(\mathcal{A}, \mathcal{B})$ against the modified split game as follows. Algorithm $\mathcal{A}$ runs $\mathcal{A}'$ and answers its ideal cipher queries using its own ideal-cipher oracle. It answers a KDM query ϕ of $\mathcal{A}'$ by forwarding $(+, \phi)$ to its $\mathcal{B}$ algorithm that is shown in Figure 7. It answers a decryption query C of $\mathcal{A}'$ by forwarding $(-, C)$ to $\mathcal{B}$. KDM functions are deterministic and stateless and we assume $\mathcal{A}'$ does not place repeat queries. Hence neither does $\mathcal{A}$. Recall that according to the rules of the KDM game no ciphertext obtained from encryption can be subsequently decrypted. We also assume, without loss of generality, that if a message M is obtained as a result of a ciphertext, then the *constant* function mapping keys to M cannot be queried to the encryption oracle (since the result is already known). We note that algorithm $\mathcal{B}$ is stateless, deterministic and places a single Chal query. It is also simulatable as when Chal implements \$ then so does $\mathcal{B}$.

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (+, \phi))$      Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (-, C))$
  $M \leftarrow \phi^{\mathrm{O}}(K)$
  $C \leftarrow \textsc{Chal}(+, K, M)$                 $M \leftarrow \textsc{Chal}(-, K, C)$
  Return $C$                                Return $M$
```

Figure 7: Algorithm $\mathcal{B}$ used for KDM analysis of the ideal cipher.

It is easy to see that when Chal implements the original (non-replaced) oracle (i.e., when $b = 0$ in the m-Split game), algorithms $(\mathcal{A}, \mathcal{B})$ runs $\mathcal{A}'$ in the KDM game with $b = 0$. When Chal implements a replaced ideal-cipher oracle, algorithms $(\mathcal{A}, \mathcal{B})$ run $\mathcal{A}'$ in the KDM game with $b = 1$. We emphasize that we are relying on the *modified* split game here as the split game performs the **sp** that does not exist in the KDM game. Hence

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathrm{Block}(k,n)}(\mathcal{A}', \Phi) \leq \mathbf{Adv}^{\text{m-split}}_{\mathrm{Block}[k,n]}(\mathcal{A}, \mathcal{B}) \ .$$

Applying Theorem 1, it remains to bound the probability that $\mathcal{B}$ meets the three validity events Sp1, Sp2 and Fg with respect to $\mathbf{sp}_{\mathrm{ic}}(L_1, L_2)$ and $\mathbf{fg}_{\mathrm{ic}}(L_2)$ based on $\mathbf{val}_{\mathrm{ic}}$ defined above.

Let us start with Fg. This event is triggered with $z_1 \neq z_2$. Suppose $\sigma_1 = \sigma_2 = +$. In this case adaptivity can be ignored since the event does not depends on the value R. Hence $\mathcal{C}$ must output $\phi_1^? \neq \phi_2^?$ such that

$$(+, K, \phi_1^{\mathcal{O}}(K)) = (+, K, \phi_2^{\mathcal{O}}(K))$$

This is equivalent to winning claw-freeness for Φ. When $\sigma_1 = \sigma_2 = -$ the event cannot be triggered as the ciphertexts must be distinct.

Let us consider now $\sigma_1 = +$ and $\sigma_2 = -$. Then $\mathcal{C}$ outputs $(+, \phi_1)$, receives a random value R, and then outputs $(-, C_2)$. Let R' be the output for the latter. Now it is either that (1) $\mathcal{B}(+, \phi_1)$ queries forward challenge on R', or (2) $\mathcal{B}(-, C_2)$ queries backward challenge on R. The former takes place with probability $1/2^n$ as R' is chosen after ϕ_1. The latter can be triggered when $C_2 = R$. But this is a disallowed queried by the rules of the KDM game: no encryption output can be decrypted.

Let $\sigma_1 = -$ and $\sigma_2 = +$. Then $\mathcal{C}$ outputs $(-, C_1)$, receives a random value R, and then outputs $(+, \phi_2)$. Let R' be the output for the latter. Now it is either that (1) $\mathcal{B}(-, C_1)$ queries backward challenge on R'. This happens with probability $1/2^n$. Or that (2) $\mathcal{B}(+, \phi_2)$ queries forward challenge on R. This can be triggered in two ways: (2.1) ϕ_2 is different from the constant function mapping all inputs to R. In this case a claw is found. (2.2) ϕ_2 is the constant function mapping to R. But we have disallowed such queries.

Let us now look at Sp1. Since queries always include keys, the value x output by $\mathcal{C}$ must also include the key. The probability of guessing the key (given possibly a random value R) is at most $1/2^k$.

If the KDM functions are oracle-independent, event Sp2 cannot be triggered and the analysis is done. For oracle-dependent KDM Sp2 can be triggered with z_1 and z_2 which correspond to either two forward or one forward and one backward query. (Since backward queries are oracle-independent, Sp2 cannot be triggered using two backward $\mathcal{B}$ queries.)

Suppose $i = 1$. If $z_1 = (-, *)$ then $L'_1 = [\,]$. So we assume $z_1 = (+, \phi_1)$. In what follows L'_1 is formed first and then L'_2.

(1) Suppose $z_2 := (+, \phi_2)$. Then L'_2 consists of a single forward entry. (1.1) A forward entry in L'_1 with a forward entry in L'_2 trigger Sp2. This violates query-independence with a reduction that simply simulates R for $\mathcal{C}$. (1.2) A backward entry in L'_1 with a forward entry in L'_2 trigger Sp2. (1.2.1) An input in L'_1 matches an output in L'_2. Since the output in L'_2 is chosen randomly and independently of inputs in L'_1, this happens with probability at most $q_\phi/2^n$, assuming the KDM functions make at most q_ϕ oracle queries. (1.2.1) An output in L'_1 matches an input in L'_2. The outputs in L'_1 are random subject to permutativity. Value R seen by $\mathcal{C}$ is simply a random value independent of outputs in $L'1$. Hence the probability of the single entry in L'_2 matching one of the outputs in L'_1 is at most $q_\phi/2^n$.

(2) Suppose $z_2 := (-, C)$. Then L'_2 consists of a single backward entry. (2.1) A forward entry in L'_1 with a backward entry in L'_2 trigger Sp2. (2.1.1) An input in L'_1 matches an output in L'_2. Since the output in L'_2 is random and independent of L'_1, this happens with probability $q_\phi/2^n$. (2.1.2) An output in L'_1 matches an input in L'_2. Since the outputs in L'_1 are random subject to permutativity and R is random and independent of these values, this happens with probability $q_\phi/2^n$. (2.2) A backward entry in L'_1 with a backward entry in L'_2 trigger Sp2. This violates query-independence.

Suppose now $i = 2$. If $z_2 = (-, *)$ then $L'_1 = [\,]$. So we assume $z_2 = (+, \phi_2)$. In what follows L'_2 is formed first and then L'_1.

(3) Suppose $z_1 := (+, \phi_1)$. Then L'_2 consists a single forward entry. (3.1) A forward entry in L'_1 with a forward entry in L'_2 trigger Sp2. This violates query-independence with a reduction that simply simulates R for $\mathcal{C}$. (3.2) A backward entry in L'_1 with a forward entry in L'_2 trigger Sp2. (3.2.1) An input in L'_1 matches an output in L'_2. Note that the KDM function can be chosen based on R, the output in L'_2. This violated query-independence. (3.2.1) An output in L'_1 matches an input in L'_2. Since the outputs in L'_1 are random subject to permutativity, and L'_2 is chosen before L'_1, this happens with probability $q_\phi/2^n$.

(4) Suppose $z_1 := (-, C)$. Then L'_2 consists of a single backward entry. (4.1) A forward entry in L'_1 with a backward entry in L'_2 trigger Sp2. (4.1.1) An input in L'_1 matches an output in L'_2. Note that the KDM function can be chosen based on R, the output in L'_2. This violated query-independence. (4.1.2) An output in L'_1 matches an input in L'_2. Since the outputs in L'_1 are random subject to permutativity, and L'_2 is chosen before L'_1, this happens with probability $q_\phi/2^n$. (4.2) A backward entry in L'_1 with a backward entry in L'_2 trigger Sp2. This violates query-independence.

Only one of the above cases need to be considered, which justifies the final term in the advantage upper bound. □

Remark. The converse of the above theorem does not hold. The set $\Phi := \{\phi_1(K) := K, \phi_2(K) := K \oplus \mathrm{MSB}(K)\}$ is *not* claw-free as $\phi_1(K) = \phi_2(K)$ with probability $1/2$. KDM security with respect to this set, however, can be proven along the following lines. Instead of simulatability of $\mathcal{B}^{\mathcal{O},\$}$, demand simulation with the help of a *claw-detection* oracle. This is an oracle that given ϕ_1 and ϕ_2 returns $(\phi_1(K) = \phi_2(K))$. This means that we can modify the validity game to one which allows $\mathcal{C}$ access to a claw-detection oracle. For oracle-free KDM functions this condition boils down to unpredictability of the key in the presence of a claw detection oracle. This results in a characterization that is tight, as predicting the key under claws can be easily used to win the KDM game for ideal cipher (since claws can be read off from the outputs of the cipher).

5 KDM Attacks on Even–Mansour

In this section we present KDM attacks on the iterated Even–Mansour ciphers. First, 1-round Even–Mansour is not KDM secure under chosen-plaintext attacks for any set Φ containing functions that offset the key, i.e., with respect to $\phi(K_1, K_2) := K_1 \oplus \Delta$. Indeed, enciphering $\phi(K_1, K_2)$ gives $\mathsf{P}(\Delta) \oplus K_2$ and hence key K_2 can be recovered after computing $\mathsf{P}(\Delta)$. Our result in the next section excludes such Φ.

We next consider 2-round EM in different configurations: the two permutations can be set to be identical or independent, and there are five possible key schedules. The simplest possible (and also the most efficient) construction uses a single random permutation and the same n-bit key in the two rounds. In the resulting scheme $\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K, K, K]$, only one key needs to be securely stored and a unique random permutation has to be implemented. Unfortunately, this cipher is vulnerable to a *sliding attack*[5] [BW99] if the set Φ contains the key offset functions.

Indeed, if the function $\phi(K) := K \oplus \mathsf{P}^{-1}(0^n)$ belongs to Φ, the attacker can simply query its encryption on it to get $C_1 = \mathsf{EM}^{\mathsf{P},\mathsf{P}}[K, K, K](\phi(K)) = \mathsf{P}(K) \oplus K$. It also obtains the encryption of 0^n as $C_2 = \mathsf{EM}^{\mathsf{P},\mathsf{P}}[K, K, K](0^n) = \mathsf{P}(\mathsf{P}(K) \oplus K) \oplus K$. The attacker can now recover the key as $\mathsf{P}(C_1) \oplus C_2$. The adversary $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K,K,K]}$, formally described in Figure 8 (left), can recover the key and this attack can easily be adapted to any number of rounds if all internal permutations are identical and all keys are equal. We note that $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K,K,K]}$ can trigger the following event and does not respect $\mathbf{fg}_{\mathrm{rp}}$ for $\mathcal{B}^{\mathsf{P},\textsc{Chal}}$: $(\sigma_1, x_1, y_1) = (\sigma_1, x_1, y_1)$ with $\sigma_1 = +$; $x_1 = \mathsf{P}(\Delta \oplus K) \oplus K$; $y_1 = \mathsf{P}(\mathsf{P}(\Delta \oplus K) \oplus K) \oplus K$.

This attack can be adapted to the key schedule $[K_1, K_2, K_2]$ as described by the adversary $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1,K_2,K_2]}$ shown in Figure 8 (right). The function ϕ_2 is now different and aims to cancel the key K_1 and replace it by K_2 to bring the setting back to one where a single key is used.

We also show that the iterated Even–Mansour construction cannot achieve KDM-CPA security beyond the birthday bound (for any number of rounds $r \geq 2$) if the set Φ contains the key offset functions (even if the random permutations and the keys are different). The adversary can simply query the KDMEnc oracle on $q_1 \geq 1$ different messages (independent of the key) $m_1, \ldots, m_{q_1}$ and store the corresponding plaintext/ciphertext pairs (m_i, c_i) for $i \in \{1, \ldots, q_1\}$ in some hash table (indexed by the ciphertext values). The adversary can then query the KDMEnc oracle on $q_2 \geq 1$ key offset functions $\phi_i(K) = K_1 \oplus \Delta_j$ with different offsets Δ_j for $j \in \{1, \ldots, q_2\}$. For each corresponding ciphertext $z_j = \mathrm{KDMEnc}(\phi_j)$, the adversary then looks for it in the hash table. If there exist c_i such that $z_j = c_i$ for $i \in \{1, \ldots, q_1\}$, then, since EM is a permutation, the adversary knows that $m_i = \Delta_j \oplus K_1$ and can retrieve K_1 as $m_i \oplus \Delta_j$. If $q_1 \cdot q_2 \simeq 2^n$ then, with high probability, the adversary will find such a collision and therefore the first round key. The

[5] This attack can be generalized readily for iterated r-rounds EM construction if it uses a single random permutation and the same n-bit key for all rounds (irrelevant of the value $r \geq 2$).

```
Adversary $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K,K,K]}$
  Chooses a value $\Delta$
  $y_1 \leftarrow \mathsf{P}(-,\Delta)$
  $\phi_1(k) := y_1 \oplus K$
  $y_2 \leftarrow \text{KDMEnc}(\phi_1)$
  $\phi_2(k) := \Delta$
  $y_2' \leftarrow \text{KDMEnc}(\phi_2)$
  $y_1' \leftarrow \mathsf{P}(+,y_2)$
  $k \leftarrow y_1' \oplus y_2'$
  $y \leftarrow \Delta \oplus k$
  $y_1'' \leftarrow \mathsf{P}(+,y)$
  If $y_2 = y_1'' \oplus k$ Return 1
  $b \twoheadleftarrow \{0,1\}$
  Return $b$
```

```
Adversary $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1,K_2,K_2]}$
  Chooses a value $\Delta$
  $y_1 \leftarrow \mathsf{P}(-,\Delta)$
  $\phi_1(k) := y_1 \oplus K_1$
  $y_2 \leftarrow \text{KDMEnc}(\phi_1)$
  $\phi_2(k) := K_1 \oplus K_2 \oplus \Delta$
  $y_2' \leftarrow \text{KDMEnc}(\phi_2)$
  $y_1' \leftarrow \mathsf{P}(+,y_2)$
  $k_2 \leftarrow y_1' \oplus y_2'$
  $y \leftarrow \Delta \oplus k_2$
  $y_1'' \leftarrow \mathsf{P}(+,y)$
  If $y_2 = y_1'' \oplus k_2$ Return 1
  $b \twoheadleftarrow \{0,1\}$
  Return $b$
```

Figure 8: Adversaries $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K,K,K]}$ and $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1,K_2,K_2]}$.

complexity to find the first round key is thus $O(2^{n/2})$ queries to KDMEnc to find the first key and the attack can be repeated to find the other keys. The overall complexity to recover the full secret key is thus $O((r-1)\cdot 2^{n/2})$ queries to KDMEnc (since the two keys of the last round can be obtained easily as described above).

6 KDM Security of Even–Mansour Ciphers

6.1 One-round Even–Mansour

We study the KDM security of the basic Even–Mansour cipher with only a single round. We show that this construction achieves nontrivial forms of KDM security.

Offset-freeness. We define the offset-freeness advantage of $\mathcal{A}$ against a KDM set Φ consisting of functions $\phi : \{0,1\}^{2n} \longrightarrow \{0,1\}^n$ as

$$\mathbf{Adv}^{\text{offset}}_{\Phi}(\mathcal{A}) := \Pr[\phi(K_1,K_2) = K_1 \oplus X : (K_1,K_2) \twoheadleftarrow \{0,1\}^{2n}; (\phi,X) \twoheadleftarrow \mathcal{A}] \ .$$

Our next result shows that one-round EM is KDM-secure against oracle-free claw-free and offset-free KDM sets. Note that xor-ing with constants is *not* offset-free.

Theorem 4. *Let Φ be an oracle-free KDM mapping $2n$-bit keys to n-bit messages. Suppose Φ is offset-free and claw-free. Then $\mathsf{EM}^{\mathsf{P}}[K_1,K_2]$ is Φ-KDM-CCA secure. More precisely, for any adversary $\mathcal{A}$ against the Φ-KDM-CCA security of $\mathsf{EM}^{\mathsf{P}}[K_1,K_2]$, there are adversaries $\mathcal{C}_1$ and $\mathcal{C}_2$ against the offset-free and claw-free properties of Φ such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathsf{EM}^{\mathsf{P}}[K_1,K_2],\Phi}(\mathcal{A}) \leq q_1q/2^n + 2q^2/2^n + q_1q(2\cdot\mathbf{Adv}^{\text{offset}}_{\Phi}(\mathcal{C}_1)+4/2^n) + q^2(2\cdot\mathbf{Adv}^{\text{cf}}_{\Phi}(\mathcal{C}_2)+2/2^n) \ ,$$

where q_1 is the number of queries of $\mathcal{A}$ to $\mathsf{P}^{\pm}$ and q is the number of challenge queries of $\mathcal{A}$ in either direction.

Proof. The proof structure is analogous to that for the KDM security of the ideal cipher. For $\mathcal{A}'$ a KDM-CCA adversary, we consider a two-stage adversary $(\mathcal{A},\mathcal{B})$ against the modified split game as follows. Algorithm $\mathcal{A}$ will run $\mathcal{A}'$ as before forwarding its queries to algorithm $\mathcal{B}$ shown in Figure 9. We assume $\mathcal{A}$ does not place repeat queries, respects the rules of the KDM game, and if it obtains a message M as a result of decrypting a ciphertext C, it does not query the constant function mapping to M to encryption. Note

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (+, \phi))$
    $(K_1, K_2) \leftarrow K$
    $X \leftarrow \phi(K_1, K_2) \oplus K_1$
    $Y \leftarrow \textsc{Chal}(+, X)$
    $C \leftarrow Y \oplus K_2$
    Return $C$
```

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (-, C))$
    $(K_1, K_2) \leftarrow K$
    $Y \leftarrow C \oplus K_2$
    $X \leftarrow \textsc{Chal}(-, Y)$
    $M \leftarrow X \oplus K_1$
    Return $M$
```

Figure 9: Algorithm $\mathcal{B}$ used in the KDM analysis of one-round EM.

that algorithm $\mathcal{B}$ is stateless, deterministic, simulatable, and places a single Chal query. This leads to the first two terms in the advantage upper bound.

Since we are only considering oracle-independent KDM functions we do not need to consider event Sp2. We consider Sp1 next. This event corresponds to finding a collision between a direct query of $\mathcal{A}$ and a challenge query of $\mathcal{B}$. The adversary can trigger this event in a number of ways as follows.

(1) Two forward queries (X, ϕ) are such that $\phi(K_1, K_2) \oplus K_1 = X$. This violates offset-freeness. Note that the order of the queries and their adaptivity do not matter as the winning condition is independent of the oracle output.

(2) Two backward queries (X, C) are such that $C \oplus K_2 = X$. This amounts to guessing K_2, which happens with probability at most $1/2^n$.

(3) A forward X and a backward C are such that: (3.1) $X = R \oplus K_1$ for a possibly known R (the output of $\mathcal{B}$). This amounts to guessing K_1, which happens with probability at most $1/2^n$. (3.2) $C \oplus K_2 = \mathsf{P}(X)$. This amounts to guessing K_2, which happens with probability at most $1/2^n$.

(4) A backward X and a forward ϕ are such that: (4.1) $X = R \oplus K_2$ for a possibly known R. This amounts to guessing K_2, which happens with probability at most $1/2^n$. (4.2) $\phi(K_1, K_2) \oplus K_1 = \mathsf{P}^-(X)$. This violates offset-freeness.

The third term in the advantage bound in the statement of the theorem follows from a union bound.

We now consider the Fg event, which corresponding to finding two collisions between two distinct challenge queries of $\mathcal{B}$. The adversary can trigger this event in a number of ways.

(1) Two forward queries are such that $\phi_1 \neq \phi_2$ and $\phi_1(K_1, K_2) \oplus K_1 = \phi_2(K_1, K_2) \oplus K_1$. This violates claw-freeness.

(2) Two backward queries are such that $C_1 \oplus K_2 = C_2 \oplus K_2$ and $C_1 \neq C_2$. This is not possible.

(3) A forward ϕ and a backward C such that C is chosen first and: (3.1) $\phi(K_1, K_2) \oplus K_1 = R \oplus K_1$ where ϕ can possibly depend on R. If $\phi(K_1, K_2)$ is the constant function mapping to R, this query is not allowed by our restriction above. Otherwise a claw with constant function mapping to R is found. (3.2) $C \oplus K_2 = R' \oplus K_2$. Since R' is randomly and independently chosen, this happens with probability $1/2^n$.

(4) A forward ϕ and a backward C such that ϕ is chosen first and: (4.1) $C \oplus K_2 = R \oplus K_2$. Here C can possibly depend on R. This violates the KDM rule that output ciphertexts (R here) are not subsequently decrypted. (4.2) $\phi(K_1, K_2) \oplus K_1 = R' \oplus K_1$. Since R' is randomly and independently chosen, this happens with probability $1/2^n$.

The forth term in the advantage bound follows from a union bound. □

6.2 Two-round Even–Mansour with independent permutations

As mentioned above, offset-freeness excludes the case of KDM security against key offsets. We ask if by addition of extra rounds to the Even–Mansour ciphers can boost KDM-CCA security against this class. In this section we show the addition of a single extra round with *independent* permutations is sufficient for this. (In the next subsection, we will consider using a single permutation.) We consider an oracle $\mathcal{O}$ that allows access to two permutations via $\mathcal{O}(i,\sigma,x) := \mathsf{P}_i^{\sigma}(x)$. This is simply an ideal cipher oracle with two key values $i = 1, 2$. Hence splitting and forgetting apply to this oracle. Our proof strategy is as before, but to avoid offset-freeness we only replace the *last* permutation in forward queries and the *first* permutation in backward queries. These will be sufficient to ensure that the outputs in both directions are randomized.

Theorem 5. *Let* Φ *be a KDM set that is claw-free. Then* $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2}[K_1,K_2,K_3]$ *is* Φ-KDM-CCA *secure. More precisely, for any adversary* $\mathcal{A}$ *against the* Φ-KDM-CCA *security of* $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2}[K_1,K_2,K_3]$*, there is an adversary* $\mathcal{C}$ *against the claw-free property of* Φ *such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2}[K_1,K_2,K_3],\Phi}(\mathcal{A}) \le 15q_1q/2^n + 2q^2 \cdot (\mathbf{Adv}^{\text{cf}}_{\Phi}(\mathcal{C}) + 1/2^n)\ ,$$

where q_1 *is the number of queries of* $\mathcal{A}$ *to* $\mathsf{P}_i^{\pm}$ *(globally) and* q *is the number of challenge queries of* $\mathcal{A}$ *in either direction.*

Proof. The proof structure is analogous to that previous cases and we describe the associated algorithm $\mathcal{B}$ in Figure 10. It is easy to verify that this algorithm responds with KDM queries under $\mathsf{EM}^{\mathsf{P}_1,\mathsf{P}_2}[K_1,K_2,K_3]$ and ϕ and satisfies the requirements of statelessness, determinism, etc. as before. We emphasize that this algorithm does not make use of queries of the form $\textsc{Chal}(1,+,\cdot)$ or $\textsc{Chal}(2,-,\cdot)$. This means that queries to $\mathrm{O}(1,+,X_1)$ (i.e., those to P_1) and queries to $\mathrm{O}(2,-,X_1)$ (i.e., those to P_2^-) can be arbitrary and without any restrictions.

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K,(+,\phi))$
  $(K_1,K_2,K_3) \leftarrow K$
  $X_1 \leftarrow \phi(K_1,K_2,K_3) \oplus K_1$
  $X_2 \leftarrow \mathrm{O}(1,+,X_1)$
  $X_3 \leftarrow \textsc{Chal}(2,+,X_2 \oplus K_2)$
  $C \leftarrow X_3 \oplus K_3$
  Return $C$
```

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K,(-,C))$
  $(K_1,K_2,K_3) \leftarrow K$
  $X_3 \leftarrow C \oplus K_3$
  $X_2 \leftarrow \mathrm{O}(2,-,X_3)$
  $X_1 \leftarrow \textsc{Chal}(1,-,X_2 \oplus K_2)$
  $M \leftarrow X_1 \oplus K_1$
  Return $M$
```

Figure 10: Algorithm $\mathcal{B}$ used in the KDM analysis of two-round EM with two permutations.

We start with Sp1. This event can be triggered in one of the following ways corresponding to choice of an input to an internally replaced oracle and a direct oracle query.

(1) Forward inputs ϕ and X such that $\mathsf{P}_1(\phi(K) \oplus K_1) \oplus K_2 = X$. We argue the probability of this event is upper-bounded by $2q_1q/2^n$. There are two cases to be considered: (1.1) The value $\phi(K) \oplus K_1$ has been queried to P_1. But this means the adversary can use $\mathsf{P}_1(\phi(K) \oplus K_1)$ and X to compute K_2. For each ϕ and all q_1 choices of X the probability is $q_1/2^n$ and thus $q_1q/2^n$ over all ϕ. (1.2) The value $\phi(K) \oplus K_1$ has not been queried to P_1 and $\mathsf{P}_1(\phi(K) \oplus K_1)$ is randomly chosen outside the view of the adversary over a set of size at least $(2^n - q_1)$. Hence this case happens with probability at most $1/(2^n - q_1)$ which is $\le 2/2^n$ for $q_1 \le 2^n/2$. We thus get an overall probability of $2q_1q/2^n$. We will use this line of argument below and other proofs later on.

(2) Backward inputs C and X such that $\mathsf{P}_2^-(C_2 \oplus K_3) \oplus K_2 = X$. This amounts to guessing K_2 with probability $2q_1q/2^n$.

(3) Forward ϕ and backward X such that for a known random R: (3.1) $R \oplus K_3 = X$. This amounts to guessing K_3 with probability $q_1q/2^n$ over all ϕ and X. (3.2) $\mathsf{P}_1(\phi(K) \oplus K_1) \oplus K_2 = \mathsf{P}_2^-(X)$. This amounts to guessing K_2 with probability $2q_1q/2^n$.

(4) Backward C and forward X such that for a known random R: (4.1) $R \oplus K_1 = X$. This amounts to guessing K_1 with probability $q_1q/2^n$ over all ϕ and X. (4.2) $\mathsf{P}_2^-(C_2 \oplus K_3) \oplus K_2 = \mathsf{P}_1(X)$. This amounts to guessing K_2 with probability $2q_1q/2^n$.

We now look at event Fg. This event can be triggered in the following ways.

(1) Forward inputs $\phi_1 \neq \phi_2$ such that $\mathsf{P}_1(\phi_1(K) \oplus K_1) \oplus K_2 = \mathsf{P}_1(\phi_2(K) \oplus K_1) \oplus K_2$. This violates claw-freeness.

(2) Backward inputs $C_1 \neq C_2$ such that $C_1 \oplus K_3 = C_2 \oplus K_3$. This is impossible.

Note that only P_2 in the forward direction and P_1^- in the backward direction are replaced. Hence these are all the collisions that need to be taken care of. The second term in the advantage bound follows.

Since $\mathcal{B}$ uses $\mathcal{O}$, we need to also consider Sp2. This event can be triggered in the following ways.

(1) Inputs ϕ_1 and C_2 such that ϕ_1 is chosen after seeing a random R and: (1.1) $\phi_1(K) \oplus K_1 = R \oplus K_1$. This is either a repeat query (when ϕ_1 is constant) or breaks claw-freeness. (1.2) $\mathsf{P}_1(\phi_1(K) \oplus K_1) = \mathsf{P}_2^-(C_2 \oplus K_3) \oplus K_2$. This amounts to guessing K_2 with probability $2q_1q/2^n$.

(2) Inputs ϕ_1 and C_2 such that C_2 is chosen after seeing random R and: (2.1) $\mathsf{P}_1(\phi_1(K) \oplus K_1) \oplus K_2 = \mathsf{P}_2^-(C_2 \oplus K_3)$. This amounts to guessing K_2 with probability $2q_1q/2^n$. (2.2) $C_2 \oplus K_3 = R \oplus K_3$. This event violates the KDM rule that an output ciphertext is not decrypted.

This concludes the proof of theorem. □

6.3 Two-round Even–Mansour with a single permutation

We now consider KDM security of two-round EM with permutation reuse.

Offset-xor-freeness. We define the offset-xor-freeness advantage of $\mathcal{A}$ against a KDM set Φ consisting of functions $\phi : \{0,1\}^{3n} \longrightarrow \{0,1\}^n$ as

$$\mathbf{Adv}^{\mathrm{ox}}_{\Phi}(\mathcal{A}) := \Pr[\phi(K_1, K_2, K_3) = K_1 \oplus K_2 \oplus X : (K_1, K_2, K_3) \twoheadleftarrow \{0,1\}^{3n}; (\phi, X) \twoheadleftarrow \mathcal{A}] .$$

Offset-xor-freeness and claw-freeness are sufficient for the KDM security of two-round EM with a single permutation.

Theorem 6. *Let Φ be a KDM set that is claw-free and offset-xor-free. Then $\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1, K_2, K_3]$ (with a single permutation) is Φ-KDM-CCA secure. More precisely, for any adversary $\mathcal{A}$ against the Φ-KDM-CCA security of $\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1, K_2, K_3]$, there is an adversary $\mathcal{C}_1$ against the claw-free property of Φ and an adversary $\mathcal{C}_2$ against the offset-xor-free property of Φ such that*

$$\mathbf{Adv}^{\text{kdm-cca}}_{\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1,K_2,K_3],\Phi}(\mathcal{A}) \leq 9q_1q/2^n + q^2(2 \cdot \mathbf{Adv}^{\mathrm{cf}}_{\Phi}(\mathcal{C}_1) + \mathbf{Adv}^{\mathrm{ox}}_{\Phi}(\mathcal{C}_2) + 9/2^n) ,$$

where q_1 is the number of queries of $\mathcal{A}$ to $\mathsf{P}^{\pm}$ and q is the number of challenge queries of $\mathcal{A}$ in either direction.

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (+, \phi))$
    $(K_1, K_2, K_3) \leftarrow K$
    $X_1 \leftarrow \phi(K_1, K_2) \oplus K_1$
    $X_2 \leftarrow \mathrm{O}(+, X_1)$
    $X_3 \leftarrow \textsc{Chal}(+, X_2 \oplus K_2)$
    $C \leftarrow X_3 \oplus K_3$
    Return $C$
```

```
Algo. $\mathcal{B}^{\mathrm{O},\textsc{Chal}}(K, (-, zC))$
    $(K_1, K_2, K_3) \leftarrow K$
    $X_3 \leftarrow C \oplus K_3$
    $X_2 \leftarrow \mathrm{O}(-, X_3)$
    $X_1 \leftarrow \textsc{Chal}(-, X_2 \oplus K_2)$
    $M \leftarrow X_1 \oplus K_1$
    Return $M$
```

Figure 11: Algorithm $\mathcal{B}$ used in the KDM analysis of two-round EM with a single permutation.

Proof. The proof structure is analogous to that previous case and we only present the associated algorithm $\mathcal{B}$ in Figure 11 below.
The adversary can trigger Sp1 in a number of ways as described below.

(1) Two forward queries (ϕ, X) are such that $\mathsf{P}(\phi_1(K) \oplus K_1) \oplus K_2 = X$. This amounts to guessing K_2 with probability $2q_1q/2^n$ (Recall that in the one-round construction we used offset-freeness at this stage.)

(2) Two backward queries (C, X) are such that $\mathsf{P}^-(C \oplus K_3) \oplus K_2 = X$. Again, this amounts to guessing K_2 with probability $2q_1q/2^n$.

(3) A backward query C and a forward direct query X are such that: (3.1) $\mathsf{P}^-(C \oplus K_3) \oplus K_2 = \mathsf{P}(X)$. This amounts to guessing K_2 with probability $2q_1q/2^n$. (3.2) For some random and known R we have $R \oplus K_1 = X$. This amounts to guessing K_3.

(4) A forward query ϕ and a backward direct query X are such that: (4.1) $\mathsf{P}(\phi(K) \oplus K_1) \oplus K_2 = \mathsf{P}^-(X)$. This amounts to guessing K_2 with probability $2q_1q/2^n$. (4.2) $R \oplus K_3 = X$. This reduces to guessing K_3.

The adversary can trigger Fg in one of the ways described below.

(1) Two forward queries $\phi_1 \neq \phi_2$ are such that $\mathsf{P}(\phi_1(K) \oplus K_1) \oplus K_2 = \mathsf{P}(\phi_2(K) \oplus K_1) \oplus K_2$. This violates claw-freeness.

(2) Two backward queries $C_1 \neq C_2$ are such that $\mathsf{P}^-(C_1 \oplus K_3) \oplus K_2 = \mathsf{P}^-(C_2 \oplus K_3) \oplus K_2$. This is not possible.

(3) A forward ϕ and a backward C are such that C is chosen second and: (3.1) $\mathsf{P}(\phi(K) \oplus K_1) \oplus K_2 = R \oplus K_1$. Here R is a random value chosen after ϕ corresponding to the output of $\mathcal{B}$ on C. This happens with probability $1/2^n$. (3.2) $\mathsf{P}^-(C \oplus K_3) \oplus K_2 = R \oplus K_1$. Here R is a random value corresponding to the output of $\mathcal{B}$ on ϕ. Here C can depend on R. This happens with probability $1/2^n$: Even if K_3 is known, this event amounts to guessing $K_1 \oplus K_2$. This event happens with probability $2q^2/2^n$. (Note that here we rely on round keys being different.)

(4) A forward ϕ and a backward C are such that ϕ is chosen second and: (4.1) $\mathsf{P}(\phi(K) \oplus K_1) \oplus K_2 = R \oplus K_1$. Here ϕ can depend on R. This happens with probability $1/2^n$: Even if we allow the value $\mathsf{P}(\phi(K) \oplus K_1)$ to be chosen, this amounts to guessing $K_1 \oplus K_2$. (4.2) $\mathsf{P}^-(C \oplus K_3) \oplus K_2 = R \oplus K_1$. Here R is chosen after C. This happens with probability $2q^2/2^n$.

We also need to analyze event Sp2 here as $\mathcal{B}$ depends on the oracle. This event can be triggered as a result of a collision between two queries to $\mathcal{B}$ one of which is to the oracle and the other to the challenge. This can happen in one of the following ways.

(1) Inputs ϕ_1 and ϕ_2 such that: $\phi_1(K) \oplus K_1 = \mathsf{P}(\phi_2(K) \oplus K_1) \oplus K_2$. If the input to the permutation cannot be guessed, this happens with low probability. Else it violates xor-offset-freeness.

(2) Inputs C_1 and C_2 such that: $C_1 \oplus K_3 = \mathsf{P}^-(C_2 \oplus K_3) \oplus K_2$. Even given K_3, this amounts to guessing K_2.

(3) Inputs ϕ_1 and C_2 such that: (3.1) $\phi_1(K) \oplus K_1 = R \oplus K_1$. If R is chosen after ϕ_1 this happens with probability $1/2^n$. Else, a constant ϕ_1 is not allowed and a non-constant ϕ_1 violates claw-freeness with the constant function mapping to R. (3.2) $C_2 \oplus K_3 = R \oplus K_3$. This violates the rules of the KDM game for a C_2 chosen after R and otherwise happens with probability $1/2^n$. (3.3) $\mathsf{P}(\phi_1(K) \oplus K_1) = \mathsf{P}^-(C_2 \oplus K_3) \oplus K_2$. Even given K_3, this amounts to guessing K_2. This happens with probability $2q^2/2^n$.

□

Remark. The adversary $\mathcal{A}^{\mathsf{P},\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K,K,K]}$ described in Figure 8 triggers the following event that does not respect $\mathbf{fg}_{\text{rp}}$ for $\mathcal{B}^{\mathsf{P},\textsc{Chal}}$: $(\sigma_1, x_1, y_1) = (\sigma_1, x_1, y_1)$ with $\sigma_1 = +$, $x_1 = \mathsf{P}(\Delta \oplus K) \oplus K$, $y_1 = \mathsf{P}(\mathsf{P}(\Delta \oplus K) \oplus K) \oplus K$. Similarly, the adversary against the scheme $\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1, K_2, K_2]$ uses KDM functions without xor-offset-freeness. It is however not clear if xor-offset-freeness is indeed necessary for the general key schedule $\mathsf{EM}^{\mathsf{P},\mathsf{P}}[K_1, K_2, K_3]$.

References

[App14] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. *Journal of Cryptology*, 27(3):429–451, July 2014.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knežević, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, Heidelberg, December 2012.

[BF15] Manuel Barbosa and Pooya Farshim. The related-key analysis of Feistel constructions. In Carlos Cid and Christian Rechberger, editors, *Fast Software Encryption – FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*, 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, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, Heidelberg, August 2008.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems – CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, Heidelberg, September 2007.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, 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: 9th Annual International Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 62–75. Springer, Heidelberg, August 2003.

[BW99] Alex Biryukov and David Wagner. Slide attacks. In Lars R. Knudsen, editor, *Fast Software Encryption – FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, Heidelberg, March 1999.

[CLL$^+$14] Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John P. Steinberger. Minimizing the two-round Even-Mansour cipher. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 39–56. 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, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, 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 *Lecture Notes in Computer Science*, pages 222–238. Springer, Heidelberg, December 2001.

[DS14] Gareth T. Davies and Martijn Stam. KDM security in the hybrid framework. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *Lecture Notes in Computer Science*, pages 461–480. Springer, Heidelberg, February 2014.

[DSST17] Yuanxi Dai, Yannick Seurin, John P. Steinberger, and Aishwarya Thiruvengadam. Five rounds are sufficient and necessary for the indifferentiability of iterated even-mansour. *IACR Cryptology ePrint Archive*, 2017:42, 2017.

[EM93] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *Advances in Cryptology – ASIACRYPT'91*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, Heidelberg, November 1993.

[FP15] Pooya Farshim and Gordon Procter. The related-key security of iterated Even-Mansour ciphers. In Gregor Leander, editor, *Fast Software Encryption – FSE 2015*, volume 9054 of *Lecture Notes in Computer Science*, 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.

[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: 14th Conference on Computer and Communications Security*, pages 466–475. ACM Press, October 2007.

[HT16] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 3–32. Springer, Heidelberg, August 2016.

[HU08] Dennis Hofheinz and Dominique Unruh. Towards key-dependent message security in the standard model. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 108–126. Springer, Heidelberg, April 2008.

[LLJ15] Xianhui Lu, Bao Li, and Dingding Jia. KDM-CCA security from RKA secure authenticated encryption. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 559–583. Springer, Heidelberg, April 2015.

[ML15] Nicky Mouha and Atul Luykx. Multi-key security: The Even-Mansour construction revisited. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 209–223. Springer, Heidelberg, August 2015.

[MTY11] Tal Malkin, Isamu Teranishi, and Moti Yung. Efficient circuit-size independent public key encryption with KDM security. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 507–526. Springer, Heidelberg, May 2011.