The Related-Key Security of Iterated Even–Mansour Ciphers

Pooya Farshim[1] and Gordon Procter[2]

[1] Queen's University Belfast, Northern Ireland, UK
[2] Royal Holloway, University of London, UK
pooya.farshim@gmail.com gordon.procter.2011@live.rhul.ac.uk

Abstract. The simplicity and widespread use of blockciphers based on the iterated Even–Mansour (EM) construction has sparked recent interest in the theoretical study of their security. Previous work has established their strong pseudorandom permutation and indifferentiability properties, with some matching lower bounds presented to demonstrate tightness. In this work we initiate the study of the EM ciphers under related-key attacks which, despite extensive prior work, has received little attention. We show that the simplest one-round EM cipher is strong enough to achieve non-trivial levels of RKA security even under chosen-ciphertext attacks. This class, however, does not include the practically relevant case of offsetting keys by constants. We show that two rounds suffice to reach this level under chosen-plaintext attacks and that three rounds can boost security to resist chosen-ciphertext attacks. We also formalize how indifferentiability relates to RKA security, showing strong positive results despite counterexamples presented for indifferentiability in multi-stage games.

Keywords. Even–Mansour, related-key attack, public permutation, ideal cipher, indifferentiability.

1 Introduction

1.1 Background

Formal analyses of cryptographic protocols often assume that cryptosystems are run on keys that are independently generated and bear no relation to each other. Implicit in this assumption is the premise that user keys are stored in protected areas that are hard to tamper with. Security under *related-key attacks* (RKAs), first identified by Biham and Knudsen [9,10,35], considers a setting where an adversary might be able to disturb user keys by injecting faults [2], and consequently run a cryptosystem on *related* keys. Resilience against RKAs has become a desirable security goal, particularly for blockciphers.

The need for RKA security is further highlighted by the fact that through (improper) design, a higher-level protocol might run a lower-level one on related keys. Prominent examples are the key derivation procedures in standardized protocols such as EMV [23] and the 3GPP integrity and confidentiality algorithms [31], where efficiency considerations have led the designers to use a blockcipher under related keys. Similar considerations can arise in the construction of tweakable blockciphers [38], if a blockcipher is called on keys that are offset by xoring tweak values. An RKA-secure primitive can offer security safeguards against such protocol misuse.

Bellare and Kohno (BK) [7] initiated the theoretical treatment of security under related-key attacks and propose definitions for RKA-secure pseudorandom functions (PRFs) and pseudorandom permutations (PRPs). The BK model were subsequently extended by Albrecht et al. [1] to idealized models of computation to account for the possibility that key might be derived in ways that depend on the ideal primitive. Both works prove that the ideal cipher is RKA secure against wide sets of related-key deriving (RKD) functions. Bellare and Cash [5] present an RKA-secure pseudorandom function from standard intractability assumptions and Bellare, Cash, and Miller [6] give a comprehensive treatment of RKA security for various cryptographic primitives, leveraging the RKA resilience of PRGs to construct RKA-secure instances of various other primitives. In this work we are interested in the RKA security of blockciphers.

1.2 The Even–Mansour ciphers

Key-alternating ciphers were introduced by Daemen and Rijmen [21] with the aim of facilitating a theoretical discussion of the design of AES. The key-alternating cipher has since become a popular paradigm for blockcipher design, with notable examples including AES [20,42], Present [14], LED [29], PRINCE [16], KLEIN [28], and

Zorro [27]. Key-alternating ciphers originate in the work of Even and Mansour [24,25], who considered a single round of the construction show in Figure 1; their motivation was to design the simplest blockcipher possible. This design is closely related to Rivest's DES-X construction, proposed as a means to protect DES against brute-force attacks [33], which itself builds on principles dating back to Shannon [46, p. 713]. In this work, we use the terms 'key-alternating cipher' and 'iterated Even–Mansour cipher' interchangeably.

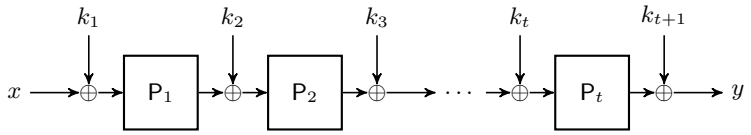

Fig. 1. The t-round iterated Even–Mansour scheme: $\mathsf{E}((k_1, \ldots, k_{t+1}), x) := \mathsf{P}_t(\cdots \mathsf{P}_2(\mathsf{P}_1(x \oplus k_1) \oplus k_2) \cdots) \oplus k_{t+1}$.

Provable security. Even and Mansour's original analysis [24,25] considers 'cracking' and 'forging' attacks in the random-permutation model and shows that no adversary can predict x given $\mathsf{E}(k, x)$ or $\mathsf{E}(k, x)$ given x with reasonable probability, without making q_1 queries to the permutation and q_{em} to the encryption/decryption oracle, where $q_1 q_{em} \approx 2^n$. The indistinguishability of the Even–Mansour scheme from a random permutation is shown by Kilian and Rogaway [33,34, Theorem 3.1 with $\kappa = 0$] and Lampe, Patarin and Seurin [36, App. B of the full version]. Both works show that an adversary making q_1 and q_{em} queries to the permutation oracle and the encryption/decryption oracles respectively, has a success probability of approximately $q_1 q_{em}/2^{n-1}$. Gentry and Ramzan [26] show that the permutation oracle can be instantiated by a Feistel network with a random oracle without loss of security.

At Eurocrypt 2012, Dunkelman, Keller, and Shamir [22] showed that the Even–Mansour scheme retains the same level of security using only a single key, that is $\mathsf{E}(k, x) = \mathsf{P}(x \oplus k) \oplus k$. Bogdanov et al. [15] show that the t-round Even–Mansour cipher with independent keys and permutations and at least two rounds ($t \geq 2$) provides security up to approximately $2^{2n/3}$ queries but can be broken in $t \cdot 2^{tn/(t+1)}$ queries. Following this work, several papers have moved towards proving a bound that meets this attack [47,36], with Chen and Steinberger [18] able to prove optimal bounds using Patarin's H-coefficient technique [44]. Chen et al. [17] consider two variants of the two-round Even–Mansour scheme: one with independent permutations and identical round keys, the other with identical permutations but a more complex key schedule. In both cases (under certain assumptions about the key schedule), security is maintained up to roughly $2^{2n/3}$ queries.

Maurer, Renner, and Holenstein (MRH) [40] introduce a framework which formalizes what it means for a non-monolithic object to be able to replace another in arbitrary cryptosystems. This framework, know as indifferentiability, has been used to validate the design principle behind many cryptographic constructions, and in particular that of the iterated Even–Mansour constructions. Lampe and Seurin [37] show that the 12-round Even–Mansour cipher using a single key is indifferentiable from the ideal cipher. Andreeva et al. [3] show that a modification of the single-key, 5-round Even–Mansour cipher, where the key is first processed through a random oracle, is indifferentiable from the ideal cipher.

Cryptanalysis. Daemen [19] describes a chosen-plaintext attack that recovers the key of Even–Mansour in approximately $q_1 \approx q_{em} \approx 2^{n/2}$ queries. Biryukov and Wagner [13] are able to give a known-plaintext attack against the Even–Mansour scheme with the same complexity as Daemen's chosen-plaintext attack. Dunkelman, Keller, and Shamir [22] introduce the slidex attack that uses only known plaintexts and can be carried out with any number of queries as long as $q_1 \cdot q_{em} \approx 2^n$.

Mendel et al. [41] describe how to extend Daemen's attack [19] to a related-key version, and are able to recover the keys when all round keys are independent. Bogdanov et al. [15] remark that related-key distinguishing attacks against the iterated Even–Mansour scheme with *independent* round keys "exist trivially," and describe a key-recovery attack, requiring roughly $2^{n/2}$ queries against the two-round Even–Mansour scheme with identical round keys, assuming that an adversary can xor constants into the round key.

Many key-alternating ciphers such as AES [12,11], Present [43], LED [41], and Prince [32] have been analyzed in the related-key model. One of the security claims of the LED blockcipher [29] is a high resistance to related-key attacks, which is justified by giving a lower bound on the number of active S-boxes.

1.3 Contributions

Despite extensive literature on the provable security of iterated Even–Mansour ciphers and (RKA) cryptanalysis of schemes using this design strategy, their formal related-key analysis has received little attention. In this work we initiate the provable RKA security analysis of such key-alternating ciphers. Our results build on the work of Barbosa and Farshim [4] who study the RKA of security of Feistel constructions. They show that by appropriate reuse of keys across the rounds, the 3-round Feistel construction achieves RKA security under chosen-plaintext attacks. With four rounds the authors are able to prove RKA security for chosen-ciphertext attacks. The authors also formalize a random-oracle model transform by Lucks [39] which processes the key via the random oracle before application. Our results are similar and we show that key reuse is also a viable strategy to protect against related-key attacks in key-alternating ciphers. In contrast to the Feistel constructions, key-alternating ciphers operate *intrinsically* in an idealized model of computation, and our analyses draw on techniques used in the formalization of Lucks's heuristic in [4].

We start with the simplest of the key-alternating ciphers, namely the (one-round) EM cipher. We recall that for xor related-key attacks, where an adversary can offset keys by values of its choice, this construction does not provide RKA security [16,15,37,3]. Indeed, it is easy to check that $\mathsf{E}((k_1, k_2), x) = \mathsf{E}((k_1 \oplus \Delta, k_2), x \oplus \Delta)$, which only holds with negligible probability for the ideal cipher. We term this pattern of adversarial behaviour *offset switching*. One idea to thwart the above attack here would be to enforce key reuse in the construction; although the above equality no longer holds, a close variant still applies:

$$\mathsf{E}(k, x) = \mathsf{E}(k \oplus \Delta, x \oplus \Delta) \oplus \Delta \ .$$

Despite this negative result, we show that the minimal EM cipher with key-reuse enjoys a non-trivial level of RKA security (even in the chosen-ciphertext setting). For a set of allowed relate-key queries Φ, we identify a set of sufficient conditions that allow us to argue that $\mathsf{E}(\phi(k), x)$ and $\mathsf{E}(\phi'(k), x')$ for $\phi, \phi' \in \Phi$ look random and independent from an adversary's point of view. As usual, our conditions impose that the RKD functions have *unpredictable* outputs, as otherwise RKA security is trivially unachievable. (For $\phi(k) = c$, a predictable value, consider an adversary which computes $\mathsf{E}(c, 0)$ and compares it $\mathsf{E}(\phi(k), 0)$.) Our second condition looks at the generalization of the offset-switching attack above and requires it to be infeasible to find offset claws, i.e., for any pair of functions (ϕ_1, ϕ_2) and any value Δ of adversary's choice, over a random choice of k

$$\phi_1(k) \oplus \phi_2(k) \neq \Delta \ .$$

This strengthens the standard claw-freeness condition [7,1,4], which corresponds to the $\Delta = 0$ case. In our work, we also consider RKD functions that *depend* on the underlying permutations by placing queries to them. As mentioned above, this is particularly relevant for the Even–Mansour ciphers as they inherently operate in the random-permutation model. We build on previous work in the analysis of such functions [1,4] and formulate adequate restrictions on oracle queries that allow a security proof to be established. Informally, our condition requires that the queries made by ϕ's have empty intersection with the outputs of ϕ's, even with offsets.

The search for xor-RKA security leads us to consider the two-round EM constructions. The first attack discussed above, where the key is offset by a constant, still applies in this setting and once again we consider key reuse. (The two permutations are still independent.) For this cipher, the offset-switching attack no longer applies, which raises the possibility that the two-round Even–Mansour might provide xor-RKA security. We start with chosen-plaintext attacks, formulate three new conditions (analogous to those given for the basic scheme), and prove security under them. These conditions, as before, decouple the queries made to the permutation oracle and allow us to simulate the outer P_2 oracle *forgetfully* in a reduction. We then show that this new set of restrictions are *weak* enough to follow from the standard output-unpredictability and claw-freeness properties. Since xoring with constants is output unpredictable and claw-free [7], the xor-RKA security of the single-key, two-round EM construction follows. Under chosen-ciphertext attacks, however, this construction falls prey to an attack of Andreeva et al. [3] on the indifferentiability of two-round EM (adapted to the RKA setting). For CCA security, we turn to three-round constructions, where we show of the 14 possible way to reuse keys, all but one fall prey to either offset switching attacks or Andreeva et al.'s attack [3]. On the other hand, the three-round construction which uses a single key meets the desired xor-RKA security in the CCA setting.

Dunkelman, Keller, and Shamir [22] consider several variants of the Even–Mansour scheme, such as *addition* Even–Mansour where the xors are replaced with modular additions, and *involution* Even–Mansour, where

random permutations are replaced with random involutions. It is reasonable to expect that our results can be modified to also apply to these schemes. Another possible variant of the Even–Mansour scheme is one where the same permutation is used across the rounds [17]; we briefly argue that our proof techniques carry over to this *permutation reuse* setting.

As mentioned above, Lampe and Seurin [37] show that the 12-round EM construction is indifferentiable from the ideal cipher when a single key is used throughout the rounds. Ristenpart, Shacham and Shrimpton [45], on the other hand, point out that indifferentiability does not necessarily guarantee composition in *multi-stage* settings and go on to note that the RKA game is multi-staged. This leaves open the question of whether indifferentiability provides any form of RKA security. We show that if RKD functions query the underlying primitive indirectly *via the construction only*, then composition holds. This level of RKA security is fairly strong as, in our opinion, it is unclear what it menas to *syntactically* changing the RKD functions from those in the ideal setting which have access to the ideal cipher to those which (suddenly) get access to permutations. Our result, in particular, implies that Lampe and Seurin's constructions [37] and Holenstein, Künzler, and Tessaro's 14-round Feistel construction [30] are RKA secure against key offsets in the CCA setting.

2 Preliminaries

NOTATION. We write $x \leftarrow y$ for assigning value y to variable x. We write $x \xleftarrow{\$} \mathsf{X}$ for the action of sampling x from a finite set X uniformly at random. If $\mathcal{A}$ is a probabilistic algorithm we write $y \xleftarrow{\$} \mathcal{A}(x_1, \ldots, x_n)$ for the action of running $\mathcal{A}$ on inputs $x_1, \ldots, x_n$ with randomly chosen coins, and assigning the results to y. We let $[n] := \{1, \ldots, n\}$, and we denote the bitwise complement of a bit string x by $\overline{x}$.

BLOCKCIPHERS. A (block)cipher is a function $\mathsf{E} \colon \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{M}$ such that for every $k \in \mathcal{K}$ the map $\mathsf{E}(k, \cdot)$ is a permutation on $\mathcal{M}$. Such an E uniquely defines its inverse map $\mathsf{D}(k, \cdot)$ for each key k. We write $\mathsf{BC} := (\mathsf{E}, \mathsf{D})$ to denote a blockcipher, which also implicitly defines the cipher's key space $\mathcal{K}$ and message space or domain $\mathcal{M}$. We denote the set of all blockciphers with key space $\mathcal{K}$ and domain $\mathcal{M}$ by $\mathrm{Block}(\mathcal{K}, \mathcal{M})$. The ideal cipher with key space $\mathcal{K}$ and message space $\mathcal{M}$ corresponds to a model of computation where all parties have oracle access to a uniformly chosen random element of $\mathrm{Block}(\mathcal{K}, \mathcal{M})$ in both the forward and backward directions. For a blockcipher $\mathsf{BC} := (\mathsf{E}, \mathsf{D})$, notation $\mathcal{A}^{\mathsf{BC}}$ denotes oracle access to both E and D for $\mathcal{A}$.

PERMUTATIONS. An ideal permutation can be viewed as a blockcipher whose key space contains a single key. In this work, we are interested in building blockciphers with large key spaces from a small number of ideal permutations $\mathsf{P}_1, \ldots, \mathsf{P}_t$ and their inverses. This is equivalent to access to a blockcipher with key space $[t]$, where $\mathsf{P}_i(x) := \mathsf{P}(i, x)$. In order to ease notation, we define a single oracle π, which provides access to all t ideal permutations in both directions. This oracle takes as input (i, x, σ), where $i \in [t]$, $x \in \mathcal{M}$, and $\sigma \in \{+, -\}$ and returns $\mathsf{P}_i(x)$ if $\sigma = +$ and $\mathsf{P}_i^{-1}(x)$ if $\sigma = -$. Slightly abusing notation, we define $\mathsf{P}_i^{\sigma}(x) := \mathsf{P}^{\sigma}(i, x) := \pi(i, x, \sigma)$, and assume $\sigma = +$ whenever it is omitted from the superscript. A blockcipher constructed from t ideal permutations π is written $\mathsf{BC}^{\pi} := (\mathsf{E}^{\pi}, \mathsf{D}^{\pi})$.

RKD FUNCTIONS. A related-key deriving (RKD) function maps keys to keys in some key space $\mathcal{K}$. In this paper, we view RKD functions as circuits that may contain special oracles gates π. An RKD set Φ is a set of RKD functions $\phi^{\pi} : \mathcal{K} \longrightarrow \mathcal{K}$, where π is an oracle. (The oracle will be instantiated with π as defined above.) Throughout the paper we assume that membership in RKD sets can be efficiently decided.

RKA SECURITY. Following [7,1], we formalize the RKA security of a blockcipher $\mathsf{BC}^{\pi} := (\mathsf{E}^{\pi}, \mathsf{D}^{\pi})$ in the (multiple) ideal-permutation model via the game shown in Figure 2. The RKA game is parametrized by an RKD set Φ which specifies the RKD functions that an adversary is permitted to query during its attack. This game also includes a procedure for oracle π defined above. We define the RKCCA advantage of an adversary $\mathcal{A}$ via

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{BC}^{\pi},\Phi,t}(\mathcal{A}) := 2 \cdot \Pr\left[\mathsf{RKCCA}_{\mathsf{BC}^{\pi},\mathcal{A},\Phi,t}\right] - 1 \ .$$

The RKCPA game and advantage are defined similarly by considering adversaries that do not make any RKDEC queries (backwards queries to the permutations are still permitted).

RKA SECURITY OF THE IDEAL CIPHER. Following [7] we define the RKA security of the ideal cipher by augmenting the procedures of the above game with those for computing the ideal cipher $\mathsf{IC} := (\mathsf{iE}, \mathsf{iD})$ in both

$\underline{\mathsf{RKCCA}_{\mathsf{BC}^\pi,\mathcal{A},\Phi,t}}$:	$\underline{\text{RKEnc}(\phi^\pi, x)}$:
$b \leftarrow\!\!\$\ \{0,1\}$; $k \leftarrow\!\!\$\ \mathcal{K}$	$k' \leftarrow \phi^\pi(k)$
$(\mathsf{P}, \mathsf{P}^{-1}) \leftarrow\!\!\$\ \text{Block}([t], \mathcal{M})$	If $b = 0$ Return $\mathsf{iE}(k', x)$
$(\mathsf{iE}, \mathsf{iD}) \leftarrow\!\!\$\ \text{Block}(\mathcal{K}, \mathcal{M})$	Return $\mathsf{E}^\pi(k', x)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\text{RKEnc},\text{RKDec},\pi}$	
Return $(b' = b)$	$\underline{\text{RKDec}(\phi^\pi, x)}$:
	$k' \leftarrow \phi^\pi(k)$
$\underline{\pi(i, x, \sigma)}$:	If $b = 0$ Return $\mathsf{iD}(k', x)$
Return $\mathsf{P}^\sigma(i, x)$	Return $\mathsf{D}^\pi(k', x)$

Fig. 2. Game defining the Φ-RKCCA security of a blockcipher $\mathsf{BC}^\pi := (\mathsf{E}^\pi, \mathsf{D}^\pi)$ with access to t ideal permutations. An adversary can query the RKEnc and RKDec oracles with a $\phi^\pi \in \Phi$ only. In the RKCPA game the adversary cannot query the RKDec oracle.

directions. When working with the ideal cipher, t is often 0, but we consider RKD functions which have oracle access to the ideal procedures iE and iD as in [1].

Even–Mansour ciphers. The t-round Even–Mansour (EM) cipher $\mathsf{EM}^\pi := (\mathsf{E}^\pi, \mathsf{D}^\pi)$ with respect to t permutations $\mathsf{P}_1,\ldots,\mathsf{P}_t$ on domain $\{0,1\}^n$ has key space $\mathcal{K} = \{0,1\}^{n(t+1)}$, domain $\mathcal{M} = \{0,1\}^n$, and is defined via

$$\mathsf{E}^\pi((k_1,\ldots,k_{t+1}),x) := \mathsf{P}_t(\cdots \mathsf{P}_2(\mathsf{P}_1(x \oplus k_1) \oplus k_2)\cdots) \oplus k_{t+1}\ ,$$
$$\mathsf{D}^\pi((k_1,\ldots,k_{t+1}),x) := \mathsf{P}_1^{-1}(\cdots \mathsf{P}_{t-1}^{-1}(\mathsf{P}_t^{-1}(x \oplus k_{t+1}) \oplus k_t)\cdots) \oplus k_1\ .$$

In this work we are interested in EM ciphers where keys are reused in various rounds. Following notation adopted in [4], we denote the EM construction where key k_{i_j} is used before round j by $\mathsf{EM}^\pi[i_1, i_2, \ldots, i_{t+1}]$. We call such key schedules *simple*. Note that $\mathcal{K} = \{0,1\}^{n\cdot|\{i_1,i_2,\ldots,i_{t+1}\}|}$ in these constructions. Of particular interest to us are the $\mathsf{EM}^\pi[1,1]$, $\mathsf{EM}^\pi[1,1,1]$ and $\mathsf{EM}^\pi[1,1,1,1]$ constructions, where a single key is used in all rounds. We emphasize that the round permutations in all these constructions are independently chosen, unless stated otherwise.

3 Indifferentiability and RKA Security

Given the indifferentiability results for the EM and Feistel constructions discussed in the introduction, in this section we study to what extent (if any) an indifferentiable construction can provide resilience against related-key attacks. We start by recalling what it means for a blockcipher construction to be indifferentiable from the ideal cipher [40].

Indifferentiability. Let $\mathsf{BC}^\pi := (\mathsf{E}^\pi, \mathsf{D}^\pi)$ be a blockcipher and let $\mathcal{S}^{\mathsf{IC}}$ be a simulator with oracle access to the ideal cipher having the same key and message spaces as those of BC^π. We define the indifferentiability advantage of a distinguished $\mathcal{D}$ with respect to $\mathcal{S}$ against BC^π via

$$\mathbf{Adv}^{\mathsf{indiff}}_{\mathsf{BC}^\pi,t}(\mathcal{S},\mathcal{D}) := \Pr\left[\mathcal{D}^{\mathsf{BC}^\pi,\pi}\right] - \Pr\left[\mathcal{D}^{\mathsf{IC},\mathcal{S}^{\mathsf{IC}}}\right]\ ,$$

where the first probability is taken over a random choice of π (as defined in Figure 2), and the second probability is taken over a random choice of a blockcipher $\mathsf{IC} := (\mathsf{iE}, \mathsf{iD})$. Note that in this definition we require a *universal* simulator.

Theorem 1. *Let Φ be an RKD set consisting of function ϕ^{OC} having access to a blockcipher oracle OC. Let π be as before, BC^π be a blockcipher construction, and $\mathcal{S}$ be an indifferentiability simulator. Then for any adversary $\mathcal{A}$ against the Φ-RKCCA security of BC^π, where the oracles in the RKD functions are instantiated with BC^π, there are adversaries $\mathcal{D}_1$ and $\mathcal{D}_2$ against the indifferentiability of BC^π, and an adversary $\mathcal{B}$ against the Φ-RKCCA of the ideal cipher, where the oracles in the RKD functions are instantiated with the ideal cipher, such that*

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{BC}^\pi,\Phi,t}(\mathcal{A}) \le \mathbf{Adv}^{\mathsf{indiff}}_{\mathsf{BC}^\pi,t}(\mathcal{S},\mathcal{D}_1) + \mathbf{Adv}^{\mathsf{indiff}}_{\mathsf{BC}^\pi,t}(\mathcal{S},\mathcal{D}_2) + \mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{IC},\Phi,t}(\mathcal{B})\ .$$

Proof. The proof structure is as follows. Given an adversary $\mathcal{A}$ against the Φ-RKCCA of BC^π we construct an adversary $\mathcal{B}$ against the Φ-RKCCA of the ideal cipher IC using the simulator $\mathcal{S}$. We then use the indifferentiability of the construction to argue that $\mathcal{B}$'s advantage is negligibility different from that of $\mathcal{A}$ for each value of the challenge bit.

Let $\mathcal{S}$ be a (universal) indifferentiability simulator as in the theorem statement. Let $\mathcal{A}$ be an adversary as above with access to π and related-key oracles RKEnc and RKDec, which use either BC^π or the ideal cipher $\mathsf{IC}' = (\mathsf{iE}', \mathsf{iD}')$ to handle the queries. Let $\mathcal{B}$ be an adversary with oracle access to IC and the RKEnc and RKDec oracles that use either IC or an independent ideal cipher IC' as follows. Algorithm $\mathcal{B}$ runs $\mathcal{A}$ and answers its π queries via $\mathcal{S}^{\mathsf{IC}}$, where $\mathcal{S}$'s oracles are answered using access to IC given to $\mathcal{B}$. To answer RKEnc (resp. RKDec) queries (ϕ^{OC}, x), algorithm $\mathcal{B}$ queries its own RKEnc (resp. RKDec) oracle on (ϕ^{OC}, x) and returns the response to $\mathcal{A}$. When $\mathcal{A}$ terminates with a bit b', algorithm $\mathcal{B}$ also returns b'.

When $\mathcal{B}$ is run with respect to related-key oracles that use $\mathsf{IC} = (\mathsf{iE}, \mathsf{iD})$, adversary $\mathcal{A}$ is run with respect to the oracles

$$\mathcal{S}^{\mathsf{IC}}(x), \quad \mathsf{iE}(\phi^{\mathsf{IC}}(k), x), \quad \mathsf{iD}(\phi^{\mathsf{IC}}(k), x) \ .$$

We use indifferentiability to show that the distribution of $\mathcal{A}$'s output is close to that of $\mathcal{A}$ when run with the oracles

$$\pi(x), \quad \mathsf{E}^\pi(\phi^{\mathsf{BC}^\pi}(k), x), \quad \mathsf{D}^\pi(\phi^{\mathsf{BC}^\pi}(k), x) \ . \tag{1}$$

Consider a distinguisher $\mathcal{D}_1$ which operates as follows. It chooses a random key k, runs $\mathcal{A}$ and answers its permutation queries using its own permutation oracle which implements either $\mathcal{S}^{\mathsf{IC}}$ or π. When an RKD query (ϕ^{OC}, x) in either the forward or backward direction is placed, algorithm $\mathcal{D}_1$ first computes $\phi^{\mathsf{OC}}(k)$ by answering ϕ's oracle queries using its second set of oracles. These implement either IC or BC^π. It then obtains a related key k', queries its forward or backward blockcipher oracle on (k', x) as needed, and returns the answer to $\mathcal{A}$. It is easy to see that $\mathcal{D}_1$ runs $\mathcal{A}$ in one of the environments above according to the oracles that $\mathcal{D}_1$ gets, and since $\mathcal{S}$ is an indifferentiability simulator for BC^π, the difference in $\mathcal{A}$'s outputs is bounded by the indifferentiability advantage.

When $\mathcal{B}$ is run with respect to related-key oracles that use $\mathsf{IC}' = (\mathsf{iE}', \mathsf{iD}')$, adversary $\mathcal{A}$ is run with respect to the oracles

$$\mathcal{S}^{\mathsf{IC}}, \quad \mathsf{iE}'(\phi^{\mathsf{IC}}(k), x), \quad \mathsf{iD}'(\phi^{\mathsf{IC}}(k), x) \ .$$

We show that the distribution of $\mathcal{A}$'s output is close to that of $\mathcal{A}$ when run with the oracles

$$\pi(x), \quad \mathsf{iE}'(\phi^{\mathsf{BC}^\pi}(k), x), \quad \mathsf{iD}'(\phi^{\mathsf{BC}^\pi}(k), x) \ . \tag{2}$$

Once again, this follows from the indifferentiability of the construction. Consider a distinguisher $\mathcal{D}_2$ which choses a key k, runs $\mathcal{A}$, and answers its permutation queries using its own permutation oracle, which implements either $\mathcal{S}^{\mathsf{IC}}$ or π as before. When an RKD query (ϕ^{OC}, x) in either the forward or backward direction is placed, algorithm $\mathcal{D}_2$ first computes $\phi^{\mathsf{OC}}(k)$ by answering ϕ's oracle queries using its own second set of oracles. Once again, these implement either IC or BC^π. It then obtains a related key k'. Algorithm $\mathcal{D}_2$ now simulates an independent ideal cipher IC' via lazy sampling, queries its forward or the backward procedure on (k', x) as needed, and returns the result to $\mathcal{A}$. It is easy to see that $\mathcal{D}_1$ runs $\mathcal{B}$ according to one of the environments above depending on the oracles that it gets. Since $\mathcal{S}$ is an indifferentiability simulator for BC^π, the difference in $\mathcal{A}$'s outputs is bounded by the indifferentiability advantage.

Finally, observe that the difference in $\mathcal{A}$'s output with respect to environments (1) and (2) is, by definition, $\mathcal{A}$'s advantage in the Φ-RKCCA game against BC^π. This concludes the proof. □

Care with composition. Ristenpart, Shacham, and Shrimpton [45] show that indifferentiability does *not* always guarantee secure composition in *multi-stage* settings where multiple adversaries communicate in restricted ways only. The authors then remark that RKA security is multi-staged. To see this note that the RKA game can be seen as consisting of two adversaries $\mathcal{A}_1^\pi$ and $\mathcal{A}_2^\pi$ where $\mathcal{A}_1^\pi$ corresponds to the standard RKA adversary $\mathcal{A}^\pi$ and $\mathcal{A}_2^\pi$ is an adversary which has access to the key k, receives a state value from $\mathcal{A}_1^\pi$ containing the description of an RKD function ϕ^π and a value x, computes $\phi^\pi(k)$ using its access to π to get k', and returns $\mathsf{E}^\pi(k', x)$ or $\mathsf{D}^\pi(k', x)$ to $\mathcal{A}_1^\pi$ as needed. Hence adversary $\mathcal{A}_2^\pi$ does not freely communicate with $\mathcal{A}_1^\pi$. The above theorem

essentially shows that in settings where $\mathcal{A}_2^\pi$ takes the form $\mathcal{A}_2^{\mathsf{BC}^\pi}$, indifferentiability suffices. In our opinion, this restricted access particularly suits well the RKA security model. Indeed, when starting in the ideal setting where the RKD functions have access to the ideal cipher, one needs to address how the oracles are instantiated when moved to the construction. A natural way to do this is to simply instantiate the oracles with those of the construction, and in this setting, as we show, indifferentiability suffices. On the other hand, giving the RKD functions direct access to π would constitute a *syntactic* change in the two RKD sets, and it is unclear how one should compare the two RKA settings.

Lampe and Seurin [37, Theorem 2] show that the 12-round $\mathsf{EM}^\pi[1,\cdots,1]$ construction is indifferentiable from the ideal cipher (with a universal simulator). Bellare and Kohno [7], on the other hand, show that the ideal cipher is $\Phi^{\oplus}$-RKCCA secure, where

$$\Phi^{\oplus} := \{k \mapsto k \oplus \Delta : \Delta \in \mathcal{K}\} \ .$$

We therefore obtain as a corollary of the above theorem that the 12-round construction $\mathsf{EM}^\pi[1,\cdots,1]$ is $\Phi^{\oplus}$-RKCCA secure. The same conclusion applies to the 14-round Feistel construction of Holenstein, Künzler, and Tessaro [30]. These construction, however, are suboptimal in terms rounds with respect to RKA security. Barbosa and Farshim [4] show that 4 rounds with key reuse suffices for Feistel networks. In the following sections, we study the Even–Mansour ciphers with smaller number of rounds while maintaining RKA security.

4 The RKA Security of $\mathsf{EM}^\pi[1,1]$

In this section we study RKD sets Φ for which the single-key Even–Mansour construction provides Φ-RKCCA security. Our results are similar to those of Bellare and Kohno [7], Albrecht et al. [1], and Barbosa and Farshim [4] in that we identify a set of restrictions on the RKD set Φ that allow us to establish a security proof. For the one-round construction there are two simple key schedules up to relabeling: $\mathsf{EM}^\pi[1,1]$ and $\mathsf{EM}^\pi[1,2]$. Neither of these constructions can provide $\Phi^{\oplus}$-RKCPA security due to the offset-switching attacks discussed in the introduction. Despite this, we show that the most simple of the EM constructions, $\mathsf{EM}^\pi[1,1]$, provides a non-trivial level of RKA security. The results of this section will also serve as a warm up to the end goal of achieving strong forms of RKA security, which will encompass key offsets as a special case.

4.1 Restricting RKD sets

Bellare and Kohno [7] observe that if an adversary is able to choose a $\phi \in \Phi$ that has *predictable* outputs on a randomly chosen key, then Φ-RKCCA security is not achievable. To see this, let ϕ be the constant zero (or any predictable) function. An adversary can simply test if it is interacting with the real or the ideal cipher by enciphering x under the zero key and comparing it to the value it receives from its RKEnc oracle on (ϕ, x). This motivates the following definition of unpredictability, adapted to the ideal-permutation model.

Output unpredictability (OUP). The advantage of an adversary $\mathcal{A}$ against the *output unpredictability* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{oup}}_{\Phi,t}(\mathcal{A}) := \Pr\left[\exists\,(\phi^\pi, c) \in \mathsf{List} : \phi^\pi(k) = c : \mathsf{List} \ \leftarrow\!\!\$\ \mathcal{A}^\pi\right] \ .$$

Here List contains pairs of the form (ϕ^π, c) for $\phi^\pi \in \Phi$ and $c \in \mathcal{K}$, and π is the oracle containing t ideal permutations. The probability is taken over a random choice of $k \leftarrow\!\!\$\ \mathcal{K}$, the t random permutations implicit in π, and the coins of the adversary. Note that via a simple guessing argument, this definition can be shown to be equivalent to one where the adversary is required to output a single pair, with a loss of $1/|\mathsf{List}|$ in the reduction.

A second condition that Bellare and Kohno [7] introduce is *claw-freeness*. Roughly speaking, a set Φ has claws if there are two distinct $\phi_1, \phi_2 \in \Phi$ such that $\phi_1(k) = \phi_2(k)$. Although this condition is not in general necessary—given an arbitrary claw there isn't necessarily an attack—it turns out that existence of claws prevent natural approaches to proofs of security. We lift claw-freeness to the ideal-permutation model below.

Claw-freeness (CF). The advantage of an adversary $\mathcal{A}$ against the *claw-freeness* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{cf}}_{\Phi,t}(\mathcal{A}) := \Pr\left[\exists\,(\phi_1^\pi,\phi_2^\pi) \in \mathsf{List} : \phi_1^\pi(k) = \phi_2^\pi(k) \wedge \phi_1^\pi \neq \phi_2^\pi : \mathsf{List} \;\leftarrow\!\!\$\; \mathcal{A}^\pi\right] .$$

Here List contains pairs of RKD functions, π is as before, and the probability space is defined similarly to that for output unpredictability. Once again this definition is equivalent to one where List is restricted to be of size one.

Claw-freeness is not a strong enough condition for the one-round EM construction to be RKA secure. Indeed, consider an adversary that queries its encryption oracle with two pairs (ϕ_1, x_1) and (ϕ_2, x_2), possibly with $x_1 \neq x_2$, such that

$$x_1 \oplus \phi_1(k) = x_2 \oplus \phi_2(k) .$$

Then the permutation underlying the construction will be queried at the same point and the resulting ciphertexts will differ by $\phi_1(k) \oplus \phi_2(k) = x_1 \oplus x_2$, a predictable value. This observation motivates a strengthening of the claw-freeness property.

Xor claw-freeness (XCF). The advantage of an adversary $\mathcal{A}$ against the *xor claw-freeness* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{xcf}}_{\Phi,t}(\mathcal{A}) := \Pr\left[\exists\,(\phi_1^\pi,\phi_2^\pi,c) \in \mathsf{List} : \phi_1^\pi(k) \oplus \phi_2^\pi(k) = c \wedge \phi_1^\pi \neq \phi_2^\pi : \mathsf{List} \;\leftarrow\!\!\$\; \mathcal{A}^\pi\right] .$$

The variables and probability space are defined similarly to those for claw-freeness.

Xor claw-freeness implies claw-freeness as the latter is a special case with $c = 0$. That claw-freeness is weaker than xor claw-freeness can be seen by considering the set $\Phi^{\oplus}$ corresponding to xoring with constants. This set can be easily shown to be output unpredictable and claw-free [7], but is not xor claw-free as

$$\phi_{\Delta_1}(k) \oplus \phi_{\Delta_2}(k) = \Delta_1 \oplus \Delta_2 \quad \text{where} \quad \phi_\Delta(k) := k \oplus \Delta .$$

We also observe that xor claw-freeness of Φ implies that there is at most one $\phi \in \Phi$ which is predictable as any *two* predictable RKD functions can be used to break xor claw-freeness.

Let us now consider oracle access in the RKD functions. Following the attacks identified in [1,4], we consider the oracle-dependent RKD set

$$\Phi := \left\{id : k \mapsto k,\, \phi^{\mathsf{P}} : k \mapsto \mathsf{P}(k)\right\} .$$

Consider the following Φ-RKCPA adversary against $\mathsf{EM}^\pi[1,1]$. Query $(id, 0)$ and get $y = \mathsf{P}(k) \oplus k$. Query (ϕ^{P}, y) and get z. Return $(z = 0)$. When interacting with $\mathsf{EM}^\pi[1,1]$ we have that

$$z = \mathsf{E}^{\mathsf{P}}(\mathsf{P}(k), \mathsf{P}(k) \oplus k) = \mathsf{P}(\mathsf{P}(k) \oplus k \oplus \mathsf{P}(k)) \oplus \mathsf{P}(k) = \mathsf{P}(k) \oplus \mathsf{P}(k) = 0 .$$

On the other hand, this identity is true with probability at most $1/(2^n - 1)$ with respect to the ideal cipher. This attack stems from the fact that when answering an RKEnc query, π is evaluated at a point already queried by an RKD function; this motivates our final restriction.

Xor query independence (XQI). The advantage of an adversary $\mathcal{A}$ against the *xor query independence* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{xqi}}_{\Phi,t}(\mathcal{A}) := \Pr\left[\exists\,(i,\sigma,\phi_1^\pi,\phi_2^\pi,c) \in \mathsf{List} : (i, \phi_1^\pi(k) \oplus c, \sigma) \in \overline{\mathsf{Qry}}[\phi_2^\pi(k)];\; \mathsf{List} \;\leftarrow\!\!\$\; \mathcal{A}^\pi\right]$$

where

$$\mathsf{Qry}[\phi^\pi(k)] := \{(i,x,\sigma) : (i,x,\sigma) \text{ queried to } \pi \text{ by } \phi^\pi(k)\} ,$$
$$\overline{\mathsf{Qry}}[\phi^\pi(k)] := \mathsf{Qry}[\phi^\pi(k)] \cup \{(i, \pi(i,x,\sigma), -\sigma) : (i,x,\sigma) \in \mathsf{Qry}[\phi^\pi(k)]\} .$$

Note that for the EM cipher, restricting the above definition to $i = 1$ suffices. We also define *query independence* [1] as above but demand that $c = 0$.

EXAMPLES. The OUP, XCF, and XQI conditions introduced above do not lead to vacuous RKD sets. As an example of an RKD set which is independent of the permutations consider

$$\Phi^{\mathsf{xu}} := \{k \mapsto H(k,x) : x \in \mathcal{K}'\} \ ,$$

where H is an xor-universal hash function from $\mathcal{K}$ to $\mathcal{K}$ with key space $\mathcal{K}'$. As a simple instantiation, let $\mathcal{K}' = \{0,1\}^k \setminus 0^k$ and for $k \in \mathcal{K}'$ define $H(k,x) := k \cdot x$, where $\{0,1\}^k$ is interpreted as $\mathrm{GF}(2^k)$ with respect to a fixed irreducible polynomial, and multiplication is defined over $\mathrm{GF}(2^k)$.

As an example of an oracle-dependent RKD set, one can take

$$\Phi := \{k \mapsto \mathsf{P}(k \oplus \Delta) : \Delta \in \mathcal{K}\} \ .$$

4.2 Sufficiency of the conditions

We now show that if an RKD set Φ meets the output unpredictability, xor claw-freeness and xor query independence properties defined above, then $\mathsf{EM}^{\pi}[1,1]$ provides Φ-RKCCA security. Throughput the paper we denote the number of queries to various oracles in an attack as follows:

q_i : the number of direct, distinct queries to π with index i made by the adversary $\mathcal{A}$.
q_{em} : the number of distinct queries to the RKENC and (if present) RKDEC oracles by $\mathcal{A}$.
q_i^{ϕ} : the number of distinct queries to π with index i made by the RKD function ϕ^{π}.

We call an RKA adversary repeat-free if it does not query its RKENC or RKDEC oracle on a pair (ϕ, x) twice. We call an RKA adversary redundancy-free if it does not query RKENC on (ϕ, x) to get y and then RKDEC on (ϕ, y) to get x, or vice versa. Without loss of generality, we assume that all adversaries in this paper are repeat-free and redundancy-free.

Theorem 2 (Φ-RKCCA security of $\mathsf{EM}^{\pi}[1,1]$). *Let Φ be an RKD set. Then for any adversary $\mathcal{A}$ against the Φ-RKCCA security of $\mathsf{EM}^{\pi}[1,1]$ with parameters as defined above, there are adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ and $\mathcal{B}_4$ such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^{\pi}[1,1],\Phi,1}(\mathcal{A}) \leq & \mathbf{Adv}^{\mathsf{oup}}_{\Phi,1}(\mathcal{B}_1) + \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,1}(\mathcal{B}_2) + \mathbf{Adv}^{\mathsf{xcf}}_{\Phi,1}(\mathcal{B}_3) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi}(\mathcal{B}_4) \\ & + \frac{q_{em}(q_1 + \sum_{\phi} q_1^{\phi})}{2^n - (q_1 + \sum_{\phi} q_1^{\phi})} + \frac{2q_{em}^2}{2^n} \ ,\end{aligned}$$

where $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$ and $\mathcal{B}_4$ output lists of sizes $2q_1q_{em}$, $2q_{em}^2$, q_{em}^2, and q_{em}^2 respectively and they all make q_1 queries to π.

We give the intuition behind the proof here and leave the details to Appendix A. The adversary $\mathcal{A}$ in the Φ-RKCCA game is run with respect to the oracles

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{P}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k), \quad \mathsf{P}^{-1}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ .$$

Our goal is to make a transition to an environment with the oracles

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{iE}(\phi^{\pi}(k), x), \quad \mathsf{iD}(\phi^{\pi}(k), x) \ ,$$

where $(\mathsf{iE}, \mathsf{iD})$ denotes the ideal cipher. To this end, we consider two intermediate environments where the last two oracles corresponding to RKENC and RKDEC are handled via a *forgetful* oracle \$ that returns uniform strings on each invocation, irrespectively of its inputs. Applying this change to the first environment above gives

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \$(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k), \quad \$(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ ,$$

while the second gives

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \$(\phi^{\pi}(k), x), \quad \$(\phi^{\pi}(k), x) \ ,$$

both of which are identical to the environment $(\mathsf{P}(x), \mathsf{P}^{-1}(x), \$(), \$())$. We will now argue that the above changes alter $\mathcal{A}$'s winning probabilities negligibly, down to the conditions on Φ that we introduced in the previous section.

Let us first look at the change where we replace $\mathsf{iE}(\phi^\pi(k), x)$ and $\mathsf{iD}(\phi^\pi(k), x)$ with $\$(\phi^\pi(k), x)$. We introduce another game and replace the random keyed permutations iE and iD by random keyed *functions* iF and iC:

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{iF}(\phi^\pi(k), x), \quad \mathsf{iC}(\phi^\pi(k), x) \ .$$

Via (a keyed extension of) the random permutation/random function (RP/RF) switching lemma [8], the environments containing $(\mathsf{iF}, \mathsf{iC})$ and $(\mathsf{iE}, \mathsf{iD})$ can be shown to be indistinguishable up to the birthday bound $q_{em}^2/2^n$. The environments containing $\mathsf{iF}(\phi^\pi(k), x)$ and $\mathsf{iC}(\phi^\pi(k), x)$ and two copies of $\$(\phi^\pi(k), x)$ and can be shown to be identical down to the CF property. Indeed, an inconsistency could arise whenever $(\phi_1^\pi, x_1) \neq (\phi_2^\pi, x_2)$ but $(\phi_1^\pi(k), x_1) = (\phi_2^\pi(k), x_2)$. This means $x_1 = x_2$ and hence we must have that $\phi_1^\pi \neq \phi_2^\pi$. But $\phi_1^\pi(k) = \phi_2^\pi(k)$ and this leads to a break of the claw-freeness.

Let us now look at the changes made when we replace $\mathsf{P}^\pm(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)$ with $\$(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)$. We need to consider the points where a forgetful simulation of P or P^{-1} via $\$$ in the last two oracles leads to inconsistencies. Let us define the following six lists.

$$\begin{aligned}
&\mathsf{List}_\mathsf{P}^+ := [(a, \mathsf{P}(a)) : \mathcal{A} \text{ queries } a \text{ to } \mathsf{P}], \qquad \mathsf{List}_\mathsf{P}^- := [(\mathsf{P}^{-1}(b), b) : \mathcal{A} \text{ queries } b \text{ to } \mathsf{P}^{-1}] \ , \\
&\mathsf{List}_\phi^+ := [(a, \mathsf{P}(a)) : \phi^\pi(k) \text{ queries } a \text{ to } \mathsf{P}], \quad \mathsf{List}_\phi^- := [(\mathsf{P}^{-1}(b), b) : \phi^\pi(k) \text{ queries } b \text{ to } \mathsf{P}^{-1}] \ , \\
&\mathsf{List}_\$^+ := [(x \oplus \phi^\pi(k), \$(x \oplus \phi^\pi(k))) : \mathcal{A} \text{ queries } (\phi^\pi, x) \text{ to RKEnc}] \ , \\
&\mathsf{List}_\$^- := [(\$(\phi^\pi(k) \oplus y), \phi^\pi(k) \oplus y) : \mathcal{A} \text{ queries } (\phi^\pi, y) \text{ to RKDec}] \ .
\end{aligned}$$

Let $\mathsf{List}_\star$ be the union of the above lists over all ϕ queried to RKEnc or RKDec. This list encodes the trace of the attack, as in the forgetful environment no queries to P or P^{-1} are made while handling RKEnc and RKDec queries. This trace is consistent with one coming from a permutation unless $\mathsf{List}_\star$ does not respect the permutivity properties, i.e., there are two entries $(a, b), (a', b') \in \mathsf{List}_\star$ such that it is not the case that $(a = a' \iff b = b')$. Note that one of these pairs must be in $\mathsf{List}_\$:= \mathsf{List}_\$^+ \cup \mathsf{List}_\$^-$ as the other oracles are faithfully implemented. There is an inconsistency on $\mathsf{List}_\star$ if and only if there is an inconsistency among two lists (one of which is either $\mathsf{List}_\$^+$ or $\mathsf{List}_\$^-$). There are 20 possibilities to consider, including the order that queries are made. We consider first query of a pair being on $\mathsf{List}_\$^+$; the other cases are dealt with symmetrically.

$\mathsf{List}_\$^+$ and $\mathsf{List}_\mathsf{P}^+$: (1) The first component of a pair on $\mathsf{List}_\$^+$—we call this a first entry on $\mathsf{List}_\$^+$—matches a first entry a on $\mathsf{List}_\mathsf{P}^+$. This means that for some query (ϕ^π, x) to RKEnc we have that $a = \phi^\pi(k) \oplus x$. This leads to a break of output unpredictability. (2) The second entry on these lists match. More explicitly, we are looking at the probability that $\mathsf{P}(a) = R$, for R the output of $\$$ on a forward query. Here we can assume that R is known and this addresses the adaptivity of choice of a. But even in this case the probability of this event is small as P is a random permutation.

$\mathsf{List}_\$^+$ and $\mathsf{List}_\mathsf{P}^-$: (1) A second entry on $\mathsf{List}_\$^+$ matches a second entry b' on $\mathsf{List}_\mathsf{P}^-$. This means that for some query (ϕ^π, x) to RKEnc with output y we have that $b' = \phi^\pi(k) \oplus y$. This leads to a break of output unpredictability. (2) The first entries match on these lists. The argument is similar to case (2) above, but now is for P^{-1}.

$\mathsf{List}_\$^+$ and List_ϕ^+ : (1) A first entry on $\mathsf{List}_\$^+$ matches a first entry List_ϕ^+. This means that for some query (ϕ_1^π, x) to RKEnc we have that $a = \phi_1^\pi(k) \oplus x$ for a query a of some other ϕ_2^π. This leads to a break of xor query independence. (2) The second entries match on these lists. The argument is as in case (2) of first pair of lists.

$\mathsf{List}_\$^+$ and List_ϕ^- : (1) A second entry on $\mathsf{List}_\$^+$ matches a second entry b' on List_ϕ^-. This means that for some query (ϕ_1^π, x) to RKEnc with output y we have that $b' = \phi_1^\pi(k) \oplus y$ for a query b' of some other ϕ_2^π. This leads to a break of xor query independence. (2) The first entries match on these lists. The argument is as in case (2) of the second pair of lists.

$\mathsf{List}_\$^+$ and $\mathsf{List}_\$^+$: Two first entries on $\mathsf{List}_\$^+$ match. This means that for two queries (ϕ_1^π, x_1) and (ϕ_2^π, x_2) to RKEnc we have that $\phi_1^\pi(k) \oplus x_1 = \phi_2^\pi(k) \oplus x_2$. Repeat-freeness ensures that $\phi_1 \neq \phi_2$ as otherwise $x_1 = x_2$ as well. This leads to a break of xor claw-freeness. (2) The second entries match on these lists. Since the oracle returns independent random values, this probability can be bounded by the birthday bound.

$\mathsf{List}_\$^+$ and $\mathsf{List}_\$^-$: A second entry on $\mathsf{List}_\$^+$ matches a second entry on $\mathsf{List}_\$^-$. This means that for a queries (ϕ_1^π, x_1) to RKEnc with outputs y_1 and (ϕ_2^π, x_2) to RKDec, we have that $\phi_1^\pi(k) \oplus y_1 = \phi_2^\pi(k) \oplus x_2$. Redundancy-freeness ensures that $\phi_1 \neq \phi_2$ as otherwise x_2 would be an encryption of x_1. This leads to a

break of xor claw-freeness. (2) The first entries match on these lists. The probability of this event can be also bounded by the birthday bound.

Hence inconsistencies among any two pairs of lists happen with small probability, and this shows that $\mathsf{List}_\star$ is also inconsistent with small probability.

5 The Φ-RKCPA Security of $\mathsf{EM}^\pi[1,1,1]$

The theorem established in the previous section does not encompass $\Phi^\oplus$-RKA security as this set is not xor claw-free. In this section, we investigate whether an extra round of iteration can extend RKA security to the $\Phi^\oplus$ set. For the two-round EM constructions, up to relabelling, there are 5 simple key schedules: $[1,1,1]$, $[1,1,2]$, $[1,2,1]$, $[1,2,2]$, and $[1,2,3]$. It is easy to see that offset-switching attacks can be used to attack the $\Phi^\oplus$-RKCPA security of all but the first of these. In the following subsections we study the RKA security of the only remaining construction, $\mathsf{EM}^\pi[1,1,1]$.

5.1 Weakening the conditions

We start by following a similar proof strategy to that given for $\mathsf{EM}^\pi[1,1]$ and identify a set of restrictions which are strong enough to enable a security proof, yet weak enough to encompass the $\Phi^\oplus$ set. Starting from the CPA environment

$$\pi(i,x,\sigma), \quad \mathsf{P}_2(\mathsf{P}_1(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k) \ ,$$

we simulate the P_2 oracle forgetfully and move to a setting with oracles

$$\pi(i,x,\sigma), \quad \$(\mathsf{P}_1(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k) \qquad \equiv \qquad \pi(i,x,\sigma), \quad \$() \ .$$

This game can be also be reached from the ideal game $\pi(i,x,\sigma), \mathsf{iE}(\phi^\pi(k),x)$ via an application of the RP/RF switching lemma [8] and the claw-freeness property as in the analysis of $\mathsf{EM}^\pi[1,1]$.

We now analyze the probability that the second environment simulates the first one in an inconsistent way. We look at inconsistencies which arise due to oracles being queried on the same inputs. The first place such an inconsistency might arise is when $\mathcal{A}$ makes an explicit π query $(2,a,+)$ that matches a query made to \$, i.e., $\mathsf{P}_1(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k) = a$ for some (ϕ^π, x). Our first condition below addresses this event; we give a slight strengthening of the condition as we will be using it later on.

First-order output unpredictability. Let $t \geq 1$. The advantage of an adversary $\mathcal{A}$ against the *first-order output unpredictability* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{oup1}}_{\Phi,t}(\mathcal{A}) := \Pr\left[\exists\,(i,\sigma,\phi^\pi,x,c) \in \mathsf{List} \text{ s.t. } \mathsf{P}_i^\sigma(\phi^\pi(k) \oplus x) \oplus \phi^\pi(k) = c : \mathsf{List} \ \leftarrow\!\!\$\ \mathcal{A}^\pi\right] \ .$$

Oracle π, the probability space, and List are defined analogously to the previous definitions. Note that in the RKCPA setting we do not need to consider inconsistencies resulting from inputs to P_1^{-1} or P_2^{-1} arising through RKDec queries, and only need to consider $(i,\sigma) = (1,+)$ above.

Inconsistencies arising as a result of two RKEnc queries (this oracle places queries to \$) lead to the following modification of claw-freeness.

First-order claw-freeness. Let $t \geq 1$. The advantage of an adversary $\mathcal{A}$ against the *first-order claw-freeness* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,t}(\mathcal{A}) := \Pr[\exists\,(i,\sigma,\phi_1^\pi,x_1,\phi_2^\pi,x_2) \in \mathsf{List} \text{ s.t. } \mathsf{P}_i^\sigma(\phi_1^\pi(k) \oplus x_1) \oplus \phi_1^\pi(k) = \mathsf{P}_i^\sigma(\phi_2^\pi(k) \oplus x_2) \oplus \phi_2^\pi(k) \wedge \phi_1^\pi \neq \phi_2^\pi : \mathsf{List} \ \leftarrow\!\!\$\ \mathcal{A}^\pi] \ .$$

We now look at inconsistencies in the simulation due to a mismatch in an RKD query to π and a query to \$ made via the RKEnc oracle. Since only the second function is forgetfully simulated, we require independence of queries for P_2 only. One again, in the RKCPA setting, restricting the definition to $(i,\sigma) = (1,+)$ suffices.

FIRST-ORDER QUERY INDEPENDENCE. Let $t \geq 2$. The advantage of an adversary $\mathcal{A}$ against the *first-order query independence* of an RKD set Φ with access to t ideal permutations is defined via

$$\mathbf{Adv}^{\mathsf{qi1}}_{\Phi,t}(\mathcal{A}) := \Pr[\exists(i,\sigma,\phi_1^\pi,x_1,\phi_2^\pi) \in \mathsf{List} : (2,\mathsf{P}_i^\sigma(\phi_1^\pi(k)\oplus x_1)\oplus\phi_1^\pi(k),\pm) \in \overline{\mathsf{Qry}}[\phi_2^\pi(k)]; \mathsf{List} \leftarrow\!\!\$\ \mathcal{A}^\pi] \ ,$$

where, as before,

$$\mathsf{Qry}[\phi^\pi(k)] := \{(i,x,\sigma) : (i,x,\sigma) \text{ queried to } \pi \text{ by } \phi^\pi(k)\} \ ,$$
$$\overline{\mathsf{Qry}}[\phi^\pi(k)] := \mathsf{Qry}[\phi^\pi(k)] \cup \{(i,\pi(i,x,\sigma),-\sigma) : (i,x,\sigma) \in \mathsf{Qry}[\phi^\pi(k)]\} \ .$$

The new set of conditions identified above allow us to carry out a similar proof strategy to that of Theorem 2 and establish the following result. (See Appendix B for the details of the proof.)

Theorem 3 (Φ-RKCPA security of $\mathsf{EM}^\pi[1,1,1]$). *Let Φ be an RKD set. Then for any adversary $\mathcal{A}$ against the Φ-RKCPA security of $\mathsf{EM}^\pi[1,1,1]$ with parameters as defined before there are $\mathcal{B}_{1a}$ against $\mathsf{OUP1}$, $\mathcal{B}_{1b}$ against OUP , $\mathcal{B}_{2a}$ against $\mathsf{QI1}$, $\mathcal{B}_{2b}$ against XQI, $\mathcal{B}_3$ against $\mathsf{CF1}$, and $\mathcal{B}_4$ against CF such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcpa}}_{\mathsf{EM}^\pi[1,1,1],\Phi,2}(\mathcal{A}) \leq & \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,2}(\mathcal{B}_{1a}) + \mathbf{Adv}^{\mathsf{oup}}_{\Phi,2}(\mathcal{B}_{1b}) + \mathbf{Adv}^{\mathsf{qi1}}_{\Phi,2}(\mathcal{B}_{2a}) + \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,2}(\mathcal{B}_{2b}) \\ & + 2\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,2}(\mathcal{B}_3) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,2}(\mathcal{B}_4) + \frac{q_{em}(q_2+\sum_\phi q_2^\phi)}{2^n-(q_2+\sum_\phi q_2^\phi)} + \frac{2q_{em}^2}{2^n} \ ,\end{aligned}$$

where $\mathcal{B}_{1a}$ and $\mathcal{B}_{1b}$ output lists of length $q_2 q_{em}$, $\mathcal{B}_{2a}$ and $\mathcal{B}_{2b}$ lists of length q_{em}^2, $\mathcal{B}_3$ a list of length q_{em}^2, and $\mathcal{B}_4$ a list of length at most q_{em}^2.

Proof (Outline). Following the high-level proof of Theorem 2, we give a brief overview of the proof. We define the lists $\mathsf{List}^\pm_{\mathsf{P}}$ and $\mathsf{List}^\pm_\phi$ (for P_2) as in the proof of Theorem 2, but now consider

$$\mathsf{List}^+_\$:= [(\mathsf{P}_1(x\oplus\phi^\pi(k))\oplus\phi^\pi(k), \$(\mathsf{P}_1(x\oplus\phi^\pi(k)))\oplus\phi^\pi(k)) : \mathcal{A} \text{ queries } (\phi^\pi,x) \text{ to } \text{RKE}\textsc{nc}] \ .$$

We need to bound the probabilities of collisions among these lists. one of which must be on $\mathsf{List}^+_\$$. There are 9 possibilities to consider, including the order that queries are made. We consider first query of a pair being on $\mathsf{List}^+_\$$; the other cases are dealt with symmetrically.

- $\mathsf{List}^+_\$$ **and** $\mathsf{List}^+_{\mathsf{P}}$: (1) A first entry on $\mathsf{List}^+_\$$ matches a first entry a on $\mathsf{List}^+_{\mathsf{P}}$. This means that for some query (ϕ^π,x) to RKE$\textsc{nc}$ we have that $a = \mathsf{P}_1(\phi^\pi(k)\oplus x)\oplus\phi^\pi(k)$. This leads to a break of first-order output unpredictability. (2) The second entry on these lists match. More explicitly, we are looking at the probability that $\mathsf{P}_2(a) = R$, for R the output of \$ on a forward query a. Here we can assume that R is known and this addresses the adaptivity of choice of a. But even in this case the probability of this event is small as P_2 is a random permutation.
- $\mathsf{List}^+_\$$ **and** $\mathsf{List}^-_{\mathsf{P}}$: (1) Two first entries match on these lists. This means that $\mathsf{P}_2^{-1}(b) = \mathsf{P}_1(\phi^\pi(k)\oplus x)\oplus\phi^\pi(k)$ for some query b to P_2^{-1}. The probability of this occurring can be bounded information theoretically as P_2 is a random permutation. (2) A second entry on $\mathsf{List}^+_\$$ matches a second entry b' on $\mathsf{List}^-_{\mathsf{P}}$. This means that for some query (ϕ^π,x) to RKE$\textsc{nc}$ with output y we have that $b' = y\oplus\phi^\pi(k)$. This leads to a break of (standard) output unpredictability.
- $\mathsf{List}^+_\$$ **and** List^+_ϕ : (1) A first entry on $\mathsf{List}^+_\$$ matches a first entry List^+_ϕ. This means that for some query (ϕ_1^π,x) to RKE$\textsc{nc}$ we have that $a = \mathsf{P}_1(\phi_1^\pi(k)\oplus x)\oplus\phi_1^\pi(k)$ for a query a of some other ϕ_2^π. This leads to a break of first-order query independence. (2) The second entries on these lists match. In this case $\mathsf{P}_2(a) = y\oplus\phi_1^\pi(k)$ for a query a of an RKD function ϕ_2^π, with y the output of the RKE$\textsc{nc}$ oracle. This probability can be bounded information theoretically as P_2 is a random permutation.
- $\mathsf{List}^+_\$$ **and** List^-_ϕ : (1) A first entry on $\mathsf{List}^+_\$$ matches a first entry List^-_ϕ. This means that for some query (ϕ_1^π,x) to RKE$\textsc{nc}$ we have that $\mathsf{P}_1^{-1}(b) = \mathsf{P}_1(\phi_1^\pi(k)\oplus x)\oplus\phi_1^\pi(k)$ for a query b of some other ϕ_2^π. The probability of this occurring can be bounded information theoretically as P_2 is a random permutation. (2) The second entries match on these lists. In this case $b = y\oplus\phi_1^\pi(k)$ for a query b of an RKD function ϕ_2^π, with y the output of the RKE$\textsc{nc}$ oracle. This probability can be bounded down to xor query independence.

$\mathsf{List}_\$^+$ **and** $\mathsf{List}_\$^+$: (1) Two first entries on $\mathsf{List}_\$^+$ match. This means that for two queries (ϕ_1^π, x_1) and (ϕ_2^π, x_2) we have that $\mathsf{P}_1(\phi_1^\pi(k) \oplus x_1) \oplus \phi_1^\pi(k) = \mathsf{P}_1(\phi_2^\pi(k) \oplus x_2) \oplus \phi_2^\pi(k)$. This leads to a break of first-order claw-freeness. (2) Two second entries on $\mathsf{List}_\$^+$ match. Since the outputs are chosen randomly, the probability of this event can be bounded information theoretically.

The above cover all possibilities that could lead to inconsistencies and the theorem follows. □

5.2 $\Phi^\oplus$-RKCPA security

We show that the restrictions identified above are weak enough so that the offset RKD set $\Phi^\oplus$ can be shown to satisfy them. We start by showing that for oracle-independent sets, Φ is output unpredictable and claw-free if and only if it is first-order output unpredictable and first-order claw-free.

Proposition 1 ($\mathsf{OUP} \wedge \mathsf{CF} \iff \mathsf{OUP1} \wedge \mathsf{CF1}$)**.** *Let Φ be an oracle-independent RKD set and let $t \geq 1$. Then for any adversary $\mathcal{A}$ against the* OUP *(resp.* CF*) game outputting a list of size ℓ and placing q_i permutation queries with index i, there is an adversary $\mathcal{B}_1$ (resp. $\mathcal{B}_2$) outputting a list of size ℓ (resp. ℓ) and placing $q_i + \delta_{1i}\ell$ (resp. q_i) permutation queries with index i such that*

$$\mathbf{Adv}^{\mathsf{oup}}_{\Phi,t}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,t}(\mathcal{B}_1) \quad \textit{and} \quad \mathbf{Adv}^{\mathsf{cf}}_{\Phi,t}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{cf1}}_{\Phi,t}(\mathcal{B}_2) \ .$$

Moreover, for any adversary $\mathcal{A}$ against OUP1 *with parameters as before, there is an adversary $\mathcal{B}_1$ against* OUP *outputting a list of size $\ell \cdot q_\pi := \ell \cdot \sum_i q_i$, where it places q_i permutation queries with index i such that*

$$\mathbf{Adv}^{\mathsf{oup1}}_{\Phi,t}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{oup}}_{\Phi,t}(\mathcal{B}_1) + \frac{\ell(q_\pi + 1)}{2^n - \ell} \ .$$

Finally, for any adversary $\mathcal{A}$ against CF1 *with parameters as before, there are adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$, where $\mathcal{B}_1$ is as in the previous case, and $\mathcal{B}_2$ outputs a list of size ℓ and makes q_i permutation queries with index i such that*

$$\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,t}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{oup}}_{\Phi,t}(\mathcal{B}_1) + 2 \cdot \mathbf{Adv}^{\mathsf{cf}}_{\Phi,t}(\mathcal{B}_2) + \frac{\ell}{2^n - \ell} + \frac{\ell}{2^n - 2\ell} \ .$$

Proof. For the first inequality, given $\mathcal{A}$ against OUP outputting List of size ℓ, algorithm $\mathcal{B}_1$ against OUP1 runs $\mathcal{A}$, simulates its π queries using its own π oracle, and constructs a new list List' consisting of tuples $(1, +, \phi, 0, \mathsf{P}_1(c) \oplus c)$ for each $(\phi, c) \in \mathsf{List}$. Now if List contains an entry (ϕ, c) such that $\phi(k) = c$, then the corresponding entry $(1, +, \phi, 0, c')$ on List' would satisfy $\mathsf{P}_1(\phi(k) \oplus 0) \oplus \phi(k) = c'$. Note that List' is also of size ℓ, but $\mathcal{B}_1$ places ℓ extra queries to P_1.

For the second inequality, given $\mathcal{A}$'s output List of size ℓ, algorithm $\mathcal{B}_2$ runs $\mathcal{A}$, simulates its π queries using its own π oracle, and constructs a new list List' consisting of tuples of the form $(1, +, \phi_1, 0, \phi_2, 0)$ for each $(\phi_1, \phi_2) \in \mathsf{List}$. Now if List contains an entry (ϕ_1, ϕ_2) such that $\phi_1(k) = \phi_2(k)$, then the corresponding entry $(1, +, \phi_1, 0, \phi_2, 0)$ on List' would satisfy $\mathsf{P}_1(\phi_1(k) \oplus 0) \oplus \phi_1(k) = \mathsf{P}_1(\phi_2(k) \oplus 0) \oplus \phi_2(k)$. Note that the size of List' is also ℓ and $\mathcal{B}_2$ also places the same number of queries as $\mathcal{A}$ to $\mathsf{P}_i^\pm$.

For the third inequality, let us consider a modified OUP1 game where the π oracle used in the winning condition is replaced with an independent random permutation π'. Since the outputs of π' are independent of $\mathcal{A}$'s view, each entry in $\mathcal{A}$'s list wins the game with probability at most $1/(2^n - \ell + 1)$, and hence $\mathcal{A}$'s advantage is at most $\ell/(2^n - \ell)$. Furthermore, these two games are identical unless $\mathcal{A}$'s list of π queries has an entry which appears on the list of π' queries. We form the lists

$$\mathsf{List}_i^+ := [(a, \mathsf{P}_i(a)) : \mathcal{A} \text{ queries } a \text{ to } \mathsf{P}_i] \quad \text{and} \quad \mathsf{List}_i^- := [(\mathsf{P}_i^{-1}(b), b) : \mathcal{A} \text{ queries } b \text{ to } \mathsf{P}_i^{-1}] \ ,$$

and analogous lists $\mathsf{List}'^\pm_i$ for π'. We consider inconsistencies with List_i^+. (The case of List_i^- is dealt with similarly.) There are two possibilities:

List_i^+ **and** List'^+_i : (1) The first entries match. Then $a = \phi(k) \oplus x$ for some (ϕ, x) and a, and we can win the output unpredictability game by outputting $(\phi, x \oplus a)$. (2) The second entries match. Then $\mathsf{P}_i(a) = \mathsf{P}'_i(a')$. This happens with probability at most $\ell_i^+ \cdot q_i^+ / (2^n - \min(\ell_i^+, q_i^+))$, where q_i^+ is the size of List_i^+ and ℓ_i^+ is the size of List'^+_i.

List_i^+ **and** $\mathsf{List'}_i^-$: (1) The second entries match. Then $b = \phi(k) \oplus x$ for some (ϕ, x) and b, and we can win the output unpredictability game by outputting $(\phi, x \oplus b)$. (2) The first entries match. Then $\mathsf{P'}_i^{-1}(b') = a$. This happens with probability at most $\ell_i^- \cdot q_i^+/(2^n - \ell_i^-)$, where ℓ_i^- is the size of $\mathsf{List'}_i^-$.

Considering all cases leads to an upper bound of $\mathbf{Adv}^{\mathsf{oup}}_{\Phi,t}(\mathcal{B}_1) + \ell \cdot q_\pi/(2^n - \ell)$.

To prove the final inequality, again we consider a modified game where the winning condition is performed with respect to an independent permutation π'. The change in $\mathcal{A}$'s success probability can be bounded as in the previous case down to output unpredictability. We modify this game further, by considering a third game whose winning requirement is changed to that of the CF game: given a list of entries $(i, \sigma, \phi_1, x_1, \phi_2, x_2)$ check if $\phi_1(k) = \phi_2(k)$ for some entry on the list. The outputs of these two games are identical unless one of the following takes place. (1) The second game outputs false and the third outputs true. In this, case we can construct an adversary which wins the CF game by simply outputs all pairs (ϕ_1, ϕ_2) in $\mathcal{A}$'s list. (2) The second game outputs true and the third outputs false. In this case, there are two sub-possibilities: (2.1) The adversary wins with a pair $(i, \sigma, \phi_1, x_1, \phi_2, x_2)$ such that $\phi_1(x_1) \oplus x_1 = \phi_2(x_2) \oplus x_2$ (but of course $\phi_1(k) \neq \phi_2(k)$). This cannot be the case as π' is a permutation. (2.2) Adversary $\mathcal{A}$ wins with a pair $(i, \sigma, \phi_1, x_1, \phi_2, x_2)$ such that $\phi_1(x_1) \oplus x_1 \neq \phi_2(x_2) \oplus x_1$. As before since π' are independent of $\mathcal{A}$'s view, the probability of this event is at most $\ell/(2^n - 2\ell)$, since each entry places 2 queries to π. Finally note that the final game is identical to the CF game (and oracle π' is not used by the game). □

Bellare and Kohno [7] show that the RKD set $\Phi^\oplus$ is output unpredictable with advantage $\ell/2^n$ for any adversary outputting a list of size ℓ, and claw-free with advantage 0. The above proposition allow us to conclude that this set is also first-order output unpredictable and first-order claw-free.

Corollary 1. *Let $t \geq 1$ and suppose $\Phi^\oplus$ is defined with respect to a key space of size 2^n. Then for any $\mathcal{A}$ outputting a list of at most $\ell \leq 2^n/4$ and making at most q_1 queries to its P_1 oracle,*

$$\mathbf{Adv}^{\mathsf{oup1}}_{\Phi^\oplus,t}(\mathcal{A}) \leq \frac{\ell \cdot (q_1 + 1)}{2^{n-1}} \quad \textit{and} \quad \mathbf{Adv}^{\mathsf{cf1}}_{\Phi^\oplus,t}(\mathcal{A}) \leq \frac{\ell \cdot (q_1 + 2)}{2^{n-1}} \ .$$

This corollary together with Theorem 3 allow us to establish that $\mathsf{EM}^\pi[1,1,1]$ is $\Phi^\oplus$-RKCPA secure.

Corollary 2. *For any adversary $\mathcal{A}$ against the $\Phi^\oplus$-RKCPA security of $\mathsf{EM}^\pi[1,1,1]$ that makes at most q_π queries to its π oracle (of which q_i are to $\pi(i,\cdot,\cdot)$) and at most q_{em} queries to its* RKEnc *oracle, with $q_2 q_{em}$, $q_{em}^2 \leq 2^n/4$, we have*

$$\mathbf{Adv}^{\mathsf{rkcpa}}_{\mathsf{EM}^\pi[1,1,1],\Phi^\oplus,2}(\mathcal{A}) \leq \frac{q_{em}(q_2 + q_{em})(2q_1 + 5)}{2^n} + \frac{q_2 q_{em}}{2^n - q_2} \ .$$

We remark that via a direct analysis (but at the expense of modularity) the cubic bound above can be tightened to a quadratic one.

5.3 A $\Phi^\oplus$-RKCCA attack on $\mathsf{EM}^\pi[1,1,1]$

The above result raises the question if the security proof can be extended to the CCA setting. Adapting an attack due to Andreeva et al. [3] on the indifferentiability of the two-round EM construction to the RKA setting, we show that $\mathsf{EM}^\pi[1,1,1]$ is $\Phi^\oplus$-RKCCA insecure. The corresponding adversary is shown in Figure 3 where $\overline{x}$ denotes $x \oplus 1^n$, and $\Delta \in \{0,1\}^n$ denotes the function $k \mapsto k \oplus \Delta$.

Analysis. When interacting with oracles implementing the EM construction, we show that $\mathcal{A}$ returns true with probability 1. We have that $y_0 = \mathsf{P}_2(\mathsf{P}_1(k) \oplus k) \oplus k$ and $y_1 = \mathsf{P}_2(\mathsf{P}_1(k) \oplus \overline{k}) \oplus \overline{k}$. Now x is calculated as

$$\overline{y_0} = \mathsf{P}_2(\mathsf{P}_1(k) \oplus k) \oplus \overline{k} \xrightarrow{\oplus \overline{k}} \mathsf{P}_2(\mathsf{P}_1(k) \oplus k) \xrightarrow{\mathsf{P}_2^{-1}} \mathsf{P}_1(k) \oplus k \xrightarrow{\oplus \overline{k}} \overline{\mathsf{P}_1(k)} \xrightarrow{\mathsf{P}_1^{-1}} \mathsf{P}_1^{-1}(\overline{\mathsf{P}_1(k)}) \xrightarrow{\oplus \overline{k}} \mathsf{P}_1^{-1}(\overline{\mathsf{P}_1(k)}) \oplus \overline{k} \ .$$

Variable y_2 is calculated as

$$\overline{x} = \mathsf{P}_1^{-1}(\overline{\mathsf{P}_1(k)}) \oplus k \xrightarrow{\oplus k} \mathsf{P}_1^{-1}(\overline{\mathsf{P}_1(k)}) \xrightarrow{\mathsf{P}_1} \overline{\mathsf{P}_1(k)} \xrightarrow{\oplus k} \overline{\mathsf{P}_1(k)} \oplus k \xrightarrow{\mathsf{P}_2} \mathsf{P}_2(\overline{\mathsf{P}_1(k)} \oplus k) \xrightarrow{\oplus k} \mathsf{P}_2(\overline{\mathsf{P}_1(k)} \oplus k) \oplus k \ .$$

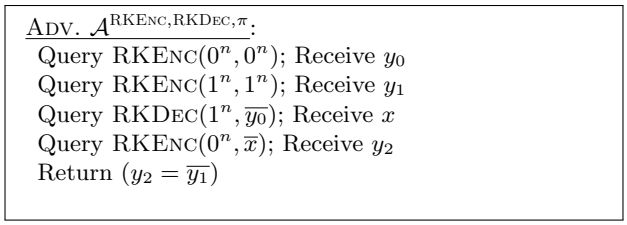

Fig. 3. Adversary $\mathcal{A}$ attacking the $\Phi^{\oplus}$-RKCCA security of $\mathsf{EM}^{\pi}[1,1,1]$.

Hence $y_2 = \mathsf{P}_2(\mathsf{P}_1(k) \oplus \overline{k}) \oplus k = \overline{y_1}$. On the other hand, when the adversary is interacting with the ideal cipher, for the equality to hold we need to have that

$$\mathsf{E}_k(\mathsf{D}_{\overline{k}}(\overline{\mathsf{E}_k(0)})) = \overline{\mathsf{E}_{\overline{k}}(1)} \quad \text{i.e.,} \quad \mathsf{D}_{\overline{k}}(\overline{\mathsf{E}_k(0)}) = \mathsf{D}_k(\overline{\mathsf{E}_{\overline{k}}(1)}) \ .$$

The latter equality however holds with negligible probability. This attack also applies if the round permutations are identical, i.e., when $\mathsf{P}_2 = \mathsf{P}_1$.

REMARK. Note that in the CCA setting we would need to simulate *both* permutations P_1 and P_2 forgetfully as forward and backward outputs need to look random. To do this we would have to re-introduce the xor claw-free condition in order to rule out collisions on P_1, which in turn excludes the $\Phi^{\oplus}$ set. It is instructive to check where the above sequence of queries triggers collisions in the *second* permutation, irrespectively of how P_1 is simulated. Let $z := \mathsf{P}_1(k) \oplus k$. During the first and second RKENC queries, P_2 is queried on points z and $\overline{z}$, respectively. During the decryption query, P_2^{-1} is queried on $\mathsf{P}_2(z)$, which is equivalent to P_2 being queried on z. This is a P_2 collision. Note also that in the third RKENC query a second collision occurs as P_2 is queried on $\overline{z}$.

6 The Φ-RKCCA Security of $\mathsf{EM}^{\pi}[1,1,1,1]$

Building on the results of the previous sections, we set out to find a key schedule for the iterated Even–Mansour construction that provides $\Phi^{\oplus}$-RKCCA security. Our previous results show that at least three rounds are necessary. We start by showing that of the fourteen possible simple key schedules for three-round EM, all but one fall prey to $\Phi^{\oplus}$-RKCCA attacks. We then show that the remaining $\mathsf{EM}^{\pi}[1,1,1,1]$ construction does indeed provide $\Phi^{\oplus}$-RKCCA security.

6.1 Attacking $\mathsf{EM}^{\pi}[\kappa]$ for $\kappa \neq [1,1,1,1]$

Up to relabeling, then there are 14 possible key schedules for the three-round Even–Mansour schemes. Of these, 9 are susceptible to offset-switching attacks. These are key schedules where a key appears only in the first or the last round and nowhere else, e.g., $[1,2,2,2]$, $[1,2,2,3]$, or $[1,2,2,1]$. This rules out 9 key schedules. Another 4 can be attacked using Andreeva et al.'s attack [3]. These are the $[1,1,2,1]$, $[1,2,1,1]$, $[1,1,2,2]$, and $[1,2,1,2]$ schedules. We give three attacks in Figure 4 below. Here, $c_1c_2 \in \{0,1\}^2$ denotes the RKD function $(k_1, k_2) \mapsto (k_1 \oplus c_1^n, k_2 \oplus c_2^n)$. The analysis of the success probabilities of these adversaries are similar to that for the attack in Section 5.3 and hence omitted.

These attacks give a generic 4-query related-key distinguisher for reduced-round LED [29] (8 out of 32 rounds for LED-64 and 16 out of 48 for LED-128). Our results lend support to the designers' claim that LED provides good related-key attack security in spite of the simple key schedule, even though they do not apply directly to LED as the round functions are neither random permutations nor independent.

6.2 RKA security of $\mathsf{EM}^{\pi}[1,1,1,1]$

We now show that $\mathsf{EM}^{\pi}[1,1,1,1]$ achieves Φ-RKCCA security for sets Φ which include, amongst others, the $\Phi^{\oplus}$ set. As before, we motivate a number of restrictions on Φ by considering a simulation strategy and analyzing

$\underline{\text{ADV. } \mathcal{A}^{\text{RKENC},\text{RKDEC},\pi}}$:	$\underline{\text{ADV. } \mathcal{A}^{\text{RKENC},\text{RKDEC},\pi}}$:	$\underline{\text{ADV. } \mathcal{A}^{\text{RKENC},\text{RKDEC},\pi}}$:
Query $\text{RKENC}(00,0)$; Get y_0	Query $\text{RKENC}(00,0)$; Get y_0	Query $\text{RKENC}(00,0)$; Get y_0
Query $\text{RKENC}(10,1)$; Get y_1	Query $\text{RKENC}(10,1)$; Get y_1	Query $\text{RKENC}(10,1)$; Get y_1
Query $\text{RKDEC}(10,\overline{y_0})$; Get x	Query $\text{RKDEC}(10,y_0)$; Get x	Query $\text{RKDEC}(10,y_0)$; Get x
Query $\text{RKENC}(00,\overline{x})$; Get y_2	Query $\text{RKENC}(00,\overline{x})$; Get y_2	Query $\text{RKENC}(00,\overline{x})$; Get y_2
Return $(y_2 = \overline{y_1})$	Return $(y_2 = y_1)$	Return $(y_2 = y_1)$

Fig. 4. Adversaries attacking $\mathsf{EM}^{\pi}[1,1,2,1]$ (left), $\mathsf{EM}^{\pi}[1,1,2,2]$ (middle), and $\mathsf{EM}^{\pi}[1,2,1,2]$ (right).

the inconsistencies that could arise. The adversary in the Φ-RKCCA game with respect to the construction has access to π and the oracles

$$\mathsf{P}_3(\mathsf{P}_2(\mathsf{P}_1(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ ,$$

$$\mathsf{P}_1^{-1}(\mathsf{P}_2^{-1}(\mathsf{P}_3^{-1}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ .$$

Once again we aim to simulate the above two oracles by returning uniformly random values. There are at least two way to perform this:

(a) Simulate the outer permutations in RKENC and RKDEC forgetfully. That is, the P_3 oracle in RKENC and the P_1^{-1} oracle in RKDEC are forgetfully implemented.
(b) Simulate the middle oracles P_2 and P_2^{-1} forgetfully. This will ensure that the inputs to $\mathsf{P}_1^{\pm}$ and $\mathsf{P}_3^{\pm}$ are randomized, and hence their outputs will be also random.

The first approach, although in some sense the more natural one, does not work. This is due to the fact that P_1 (resp. P_3) also appear as the first-round permutation in RKENC (resp. RKDEC). An adversary which performs an offset switch can trigger collisions in these oracles without being detected. We therefore adapt the second simulation strategy and for forgetful oracle \$ consider

$$\mathsf{P}_3(\$(\mathsf{P}_1(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ ,$$

$$\mathsf{P}_1^{-1}(\$(\mathsf{P}_3^{-1}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ .$$

We now consider inconsistencies, starting with a query collision between π (from a query of $\mathcal{A}$) and \$ arising from either the forward or backwards direction. Here we rely on first-order output unpredictability, but note that $(i,\sigma) = (1,+)$ and $(i,\sigma) = (3,-)$ will be critically relied on. Collisions arising between an RKD query to π and a \$ query in either direction can be ruled out down to first-order query independence; once again $(i,\sigma) \in \{(1,+),(3,-)\}$ will be used. Finally, the probability that a collision occurs as a result of two queries to \$ and/or $\$^{-1}$ can be bounded by the first-order claw freeness property. As before, inconsistencies also arise due to collisions between the outputs of oracle queries; the probability of this occurring can be bounded information theoretically. Note that here we also rely on independence of queries to the second permutation, but both cases $(i,\sigma) \in \{(1,+),(3,-)\}$ in the definition will be used. We formally prove the following theorem in Appendix C.

Theorem 4 (Φ-RKCCA security of $\mathsf{EM}^{\pi}[1,1,1,1]$). *Let Φ be an RKD set. Then for any adversary $\mathcal{A}$ against the Φ-RKCPA security of $\mathsf{EM}^{\pi}[1,1,1,1]$ with parameters as before, we have adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ such that*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^{\pi}[1,1,1,1],\Phi,3}(\mathcal{A}) \leq & \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,3}(\mathcal{B}_1) + \mathbf{Adv}^{\mathsf{xqi1}}_{\Phi,3}(\mathcal{B}_2) + 2\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,3}(\mathcal{B}_3) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,3}(\mathcal{B}_4) \\ & + \frac{2q_{em}^2}{2^n} + \frac{2q_{em}(q_2 + \sum_{\phi} q_2^{\phi})}{2^n - (q_2 + \sum_{\phi} q_2^{\phi})} \ ,\end{aligned}$$

where $\mathcal{B}_1$ outputs a list of length $2q_2q_{em}$, $\mathcal{B}_2$ a list of length $2q_{em}^2$, $\mathcal{B}_3$ a list of length q_{em}^2, and $\mathcal{B}_4$ a list of length at most q_{em}^2.

Corollary 1 together with Theorem 4 allow us to establish that the three-round single-key Even–Manour construction with independent round permutations is $\Phi^{\oplus}$-RKCCA secure:

Corollary 3. *For any adversary $\mathcal{A}$ against the $\Phi^{\oplus}$-RKCCA security of $\mathsf{EM}^{\pi}[1,1,1,1]$ with parameters defined as before. Then*

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^{\pi}[1,1,1,1],\Phi^{\oplus},3}(\mathcal{A}) \leq \frac{2q_{em}(q_2+q_{em})(2q_1+2q_3+9)}{2^n} + \frac{2q_{em}q_2}{2^n-q_2} \ .$$

Once again, via a direct analysis (but at the expense of modularity) the cubic bound above can be tightened to a quadratic one.

Permutation reuse. It is interesting to see if the above results can be further minimized by considering *permutation re-use* across the rounds. If the same permutation is used in the first and third rounds, a similar proof strategy applies as these oracles would be faithfully simulated in the reduction. The proof, however, does not immediately apply if the same permutation is used in all rounds as a forgetful simulation of the middle oracle introduces inconsistencies across the rounds. Without going into the details, we remark that this can be made to work by introducing a new CF-type assumption which requires that it is infeasible to find $(\phi_1, x_1, \phi_2, x_2)$ such that $\phi_2(k) \oplus x_2 = \mathsf{P}^{\pm}(x_1 \oplus \phi_1(k)) \oplus \phi_1(k)$. This condition would ensure that inconsistencies resulting from a P query in the first or thirds rounds and a \$ query happen with low probability. Following the proof of Proposition 1, we can also reduce this new property to standard OUP and CF notions: starting from the above winning condition, first consider the game where the winning condition uses an independent permutation (this change reduces to OUP), then consider the winning condition $\phi_1(k) = \phi_2(k)$ (an adversary winning this game wins the CF game), finally if an adversary wins the second game but not the third, then they have found a solution to $\phi_1(k) \oplus \phi_2(k) = x \oplus R$ where R is the random output of the independent permutation, which happens with probability at most $\ell/(2^n - 2\ell)$ as x, $\phi_1(k)$, and $\phi_2(k)$ are independent of R, where ℓ is the size of the list output by the adversary.

References

1. Martin R. Albrecht, Pooya Farshim, Kenny 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, Lyngby, Denmark, February 13–16, 2011. Springer, Berlin, Germany.
2. Ross J. Anderson and Markus G. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *Security Protocols, 5th International Workshop, Paris, France, April 7-9, 1997, Proceedings*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 1997.
3. Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger. On the indifferentiability of key-alternating ciphers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 531–550, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Berlin, Germany.
4. Manuel Barbosa and Pooya Farshim. The related-key analysis of feistel constructions. In Carlos Cid and Christian Rechberger, editors, *FSE 2014*, LNCS. Springer, 2014. (to appear).
5. Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 666–684, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.
6. 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, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany.
7. 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, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany.
8. 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, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany.
9. Eli Biham. New types of cryptoanalytic attacks using related keys (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 398–409, Lofthus, Norway, May 23–27, 1993. Springer, Berlin, Germany.
10. Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.
11. 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, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

12. 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, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.
13. Alex Biryukov and David Wagner. Advanced slide attacks. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 589–606, Bruges, Belgium, May 14–18, 2000. Springer, Berlin, Germany.
14. 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, *CHES 2007*, volume 4727 of *LNCS*, pages 450–466, Vienna, Austria, September 10–13, 2007. Springer, Berlin, Germany.
15. 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, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.
16. 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, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 208–225, Beijing, China, December 2–6, 2012. Springer, Berlin, Germany.
17. 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, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 39–56, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Berlin, Germany.
18. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 327–350, Copenhagen, Denmark, May 11–15, 2014. Springer, Berlin, Germany.
19. Joan Daemen. Limitations of the Even-Mansour construction (rump session). In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 495–498, Fujiyoshida, Japan, November 11–14, 1991. Springer, Berlin, Germany.
20. Joan Daemen and Vincent Rijmen. The block cipher Rijndael. In Jean-Jacques Quisquater and Bruce Schneier, editors, *Smart Card Research and Applications*, volume 1820 of *Lecture Notes in Computer Science*, pages 277–284. Springer Berlin Heidelberg, 2000.
21. 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, Cirencester, UK, December 17–19, 2001. Springer, Berlin, Germany.
22. Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The Even-Mansour scheme revisited. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 336–354, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.
23. EMVCo. *EMV Integrated Circuit Card Specifications for Payment Systems, Book 2, Security and Key Management*, June 2008. Version 4.2.
24. 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, *ASIACRYPT'91*, volume 739 of *LNCS*, pages 210–224, Fujiyoshida, Japan, November 11–14, 1991. Springer, Berlin, Germany.
25. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–162, 1997.
26. Craig Gentry and Zulfikar Ramzan. Eliminating random permutation oracles in the Even-Mansour cipher. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 32–47, Jeju Island, Korea, December 5–9, 2004. Springer, Berlin, Germany.
27. Benoît Gérard, Vincent Grosso, María Naya-Plasencia, and François-Xavier Standaert. Block ciphers that are easier to mask: How far can we go? In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *LNCS*, pages 383–399, Santa Barbara, California, US, August 20–23, 2013. Springer, Berlin, Germany.
28. Zheng Gong, Svetla Nikova, and Yee-Wei Law. KLEIN: a new family of lightweight block ciphers. In *RFID. Security and Privacy*, pages 1–18. Springer Berlin Heidelberg, 2011.
29. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 326–341, Nara, Japan, September 28 – October 1, 2011. Springer, Berlin, Germany.
30. Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 89–98, San Jose, California, USA, June 6–8, 2011. ACM Press.
31. 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, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany.

32. Jérémy Jean, Ivica Nikolic, Thomas Peyrin, Lei Wang, and Shuang Wu. Security analysis of PRINCE. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 92–111, Singapore, March 11–13, 2013. Springer, Berlin, Germany.
33. Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 252–267, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Berlin, Germany.
34. Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *Journal of Cryptology*, 14(1):17–35, 2001.
35. Lars Ramkilde Knudsen. Cryptanalysis of LOKI 91. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology – AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 196–208. Springer Berlin Heidelberg, 1993.
36. Rodolphe Lampe, Jacques Patarin, and Yannick Seurin. An asymptotically tight security analysis of the iterated Even-Mansour cipher. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 278–295, Beijing, China, December 2–6, 2012. Springer, Berlin, Germany.
37. 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, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany.
38. Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany.
39. Stefan Lucks. Ciphers secure against related-key attacks. In Bimal K. Roy and Willi Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 359–370, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany.
40. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany.
41. Florian Mendel, Vincent Rijmen, Deniz Toz, and Kerem Varici. Differential analysis of the LED block cipher. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 190–207, Beijing, China, December 2–6, 2012. Springer, Berlin, Germany.
42. National Institute of Standards and Technology. FIPS Publication 197, Announcing the Advanced Encryption Standard (AES), 2001.
43. Onur Özen, Kerem Varici, Cihangir Tezcan, and Çelebi Kocair. Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 09*, volume 5594 of *LNCS*, pages 90–107, Brisbane, Australia, July 1–3, 2009. Springer, Berlin, Germany.
44. Jacques Patarin. The "coefficients H" technique (invited talk). In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *SAC 2008*, volume 5381 of *LNCS*, pages 328–345, Sackville, New Brunswick, Canada, August 14–15, 2008. Springer, Berlin, Germany.
45. Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 487–506, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany.
46. C E Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 128(4), October 1949.
47. John Steinberger. Improved security bounds for key-alternating ciphers via hellinger distance. Cryptology ePrint Archive, Report 2012/481, 2012. `http://eprint.iacr.org/2012/481`.

A Proof of Theorem 2: The Φ-RKCCA Security of $\mathsf{EM}^{\pi}[1,1]$

Proof. The proof proceeds through four stages. In the first, $\mathcal{A}$ interacts with a public permutation and its inverse, plus the forward and backward directions of the Even–Mansour scheme instantiated with the same permutation:

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{P}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k), \quad \mathsf{P}^{-1}(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ .$$

We then consider an environment in which P and P^{-1} are replaced by \$, a forgetful random oracle, for queries made to the Even–Mansour scheme:

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \$(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k), \quad \$(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)$$

which is identical to

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \$(\phi^{\pi}(k), x), \quad \$(\phi^{\pi}(k), x) \ .$$

Finally, we transition to games in which \$ is replaced by keyed random functions iF and iC:

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{iF}(\phi^{\pi}(k), x), \quad \mathsf{iC}(\phi^{\pi}(k), x) \ ,$$

and then by the ideal cipher $(\mathsf{iE}, \mathsf{iD})$:

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{iE}(\phi^{\pi}(k), x), \quad \mathsf{iD}(\phi^{\pi}(k), x) \ .$$

We will now argue that the above changes alter $\mathcal{A}$'s winning probabilities negligibly and bound $\mathcal{A}$'s winning probability in terms of the conditions on Φ introduced in Section 4.

The first transition is analyzed via a series of games, given in Figure 5. These games include two intermediate transitions: in the first, P is replaced with Q (a random permutation, chosen independently of P) for queries arising through RKEnc (or RKDec); in the second, Q is replaced with \$ (a forgetful random oracle). We identify the points at which these two intermediate transitions lead to inconsistencies, by setting Bad flags. In contrast to how the intuition behind this proof is described in Section 4, we defer the bounding of the probability bad events occurring during the first intermediate transition until after the second intermediate transition. The specification of $\mathsf{DS}_1{}^{-1}$ and $\mathsf{IS}_1{}^{-1}$ are omitted for conciseness; they are defined analogously to their respective forward oracles. Without loss of generality, we will assume that no adversary makes repeat or redundant queries – this assumption is needed in the transitions to and from the forgetful random oracles. Let S_i denote the event where the adversary outputs 1 in $\mathsf{Game}\ i$.

$\mathsf{Game}\ 0$ is the RKA game augmented with a public permutation oracle (as described in Section 2), conditioned on $b = 1$. In this game, the adversary interacts with oracles realizing the public permutation P and the Even–Mansour construction instantiated with P.

$\mathsf{Game}\ 1$ is only syntactically different from $\mathsf{Game}\ 0$. The queries to π are split into two groups: those made directly to π, either by the adversary or by an RKD function, which are answered by the sampling algorithm DS_1 (or $\mathsf{DS}_1{}^{-1}$); and those made indirectly, through queries made to RKEnc (or RKDec), which are answered by IS_1 (or $\mathsf{IS}_1{}^{-1}$). The oracles DS_1 and IS_1 maintain consistent lists $\mathtt{I}_1$ and $\mathtt{D}_1$; the lists used by inverse oracles are identical to the lists used by the corresponding forward oracles. As this is a purely syntactic change, $\Pr[S_0] = \Pr[S_1]$.

$\mathsf{Game}\ 1a$ introduces syntactic changes to DS_1 and IS_1 (and the corresponding inverse oracles) in order for the code in the games that follow to be identical until specified bad events occur. Introducing this step will allow us to remove the statement $b \xleftarrow{\$} \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$; this is necessary as we wish to completely decouple DS_1 (and $\mathsf{DS}_1{}^{-1}$) from IS_1 (and $\mathsf{IS}_1{}^{-1}$).

$\mathsf{Game}\ 2$ sets Bad_1 either if DS_1 is queried on a point already defined in $\mathtt{I}_1$ or if IS_1 is queried on a point already defined in $\mathtt{D}_1$ (and similarly for the inverse oracles). This occurs either because $\mathcal{A}$ queries π directly at a point that is also queried through an indirect RKEnc (or RKDec) query, or because an RKD function queries π at a point that is also queried through an RKEnc (or RKDec) query. We will later bound the probability of this event in terms of the output unpredictability and query independence of Φ. $\mathsf{Game}\ 2$ sets Bad_2 if the value chosen at random for $\mathsf{DS}_1(a)$ is already defined in range of IS_1, or vice versa (and similarly for the inverse queries and the domain of IS_1 or DS_1). This is necessary because in Game 1, for both DS_1 and IS_1, b is sampled from $\{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ whereas our objective in Game 3 is to ensure that DS_1 and $\mathsf{DS}_1{}^{-1}$ are independent of IS_1 and $\mathsf{IS}_1{}^{-1}$. The outputs of DS_1 and IS_1 (and their inverses) remain consistent and $\Pr[S_1] = \Pr[S_2]$.

$\mathsf{Game}\ 3$ omits the boxed statements in $\mathsf{Game}\ 2$ and so is identical to $\mathsf{Game}\ 2$ unless one of Bad_1 or Bad_2 is set. In this game, the oracles DS_1 and IS_1 check consistency with their own lists (and the list for their corresponding inverse oracle contains all the same entries as their list), but they may become inconsistent with each other. It is possible for Bad_1 to be set in two different ways:

- E_1 is the event an adversary directly queries DS_1 at a point coinciding with a point queried to IS_1 from a query to RKEnc (or comparable conditions resulting from queries to $\mathsf{DS}_1{}^{-1}$ or RKDec).
- E_2 is the event an RKD function queries DS_1 at a point coinciding with a point queried to IS_1 from a query to RKEnc (or comparable conditions resulting from queries to $\mathsf{DS}_1{}^{-1}$ or RKDec).

We will analyze each of the ways that Bad_1 can be set below. Similarly, Bad_2 can be set either because of a query to DS_1 from $\mathcal{A}$, a query to DS_1 from ϕ^{π}, or from a query to IS_1 due to a query to RKEnc (or similarly for the corresponding inverse oracles); we consider all cases simultaneously below. In $\mathsf{Game}\ 3$, the responses to RKEnc (and RKDec) queries are completely decoupled from the responses to π queries, so we can consider that RKEnc (and RKDec) use Q to respond to queries and π uses P. We have that $\Pr[S_2] \leq \Pr[S_3] + \Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2]$.

Game 4 sets Bad_3 if a query to RKEnc (or RKDec) results in a value being queried to IS_1 (or $\mathsf{IS}_1{}^{-1}$) that is already in $\mathtt{I}_1$ (or $\mathtt{I}_1{}^{-1}$). Game 4 chooses the response to IS_1 uniformly from $\{0,1\}^n$ and sets Bad_4 if this value is already in $\mathtt{Rng}(\mathsf{IS}_1)$ (and similarly for $\mathsf{IS}_1{}^{-1}$). The flag Bad_4 can be set in four ways (as a result of two queries to either of IS_1 and $\mathsf{IS}_1{}^{-1}$, plus two 'mixed cases' with one query to each of IS_1 and $\mathsf{IS}_1{}^{-1}$); we consider each of these cases when we analyze the probability of setting bad events below. Game 4 is equivalent to Game 3 and, in particular, $\Pr[S_3] = \Pr[S_4]$.

Game 5 omits the boxed statements from Game 4 and so is identical to Game 4 unless Bad_3 or Bad_4 is set. Let $E_1', E_2', \mathsf{Bad}_2'$ represent events in Game 5 corresponding to events E_1, E_2, Bad_2 in Game 4, then $\Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2] \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4]$. In this game, calls to $\pi(1,\cdot,+)$ through RKEnc (which are answered by IS_1) are answered by a forgetful random oracle and so the ciphertexts are uniform and independent of the key and the plaintext (the same is true for calls to the inverse oracles).

In Game 5, the adversary interacts with

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \$(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k), \quad \$(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k) \ .$$

During the transitions to

$$\mathsf{P}(x), \quad \mathsf{P}^{-1}(x), \quad \mathsf{iF}(\phi^\pi(k), x), \quad \mathsf{iC}(\phi^\pi(k), x)$$

inconsistencies only arise if the adversary makes queries $(\phi_1^\pi, x_1) \neq (\phi_2^\pi, x_2)$, but where $(\phi_1^\pi(k), x_1) = (\phi_2^\pi(k), x_2)$. If an adversary $\mathcal{A}$ makes such a query, we can construct an adversary $\mathcal{B}_4$ which wins the CF game with a list of length at most $\frac{q_{em}^2}{2}$ as follows: $\mathcal{B}_4$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(\phi_i^\pi, \phi_j^\pi) : 1 \leq i < j \leq q_{em}\}$.

In the final transition, we switch from a random function to a random permutation (for each ϕ^π); the probability of an inconsistency arising in this step is bounded by $\frac{q_{em}^2}{2^n}$ [8].

Therefore we have that

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^\pi[1,1],\Phi,1}(\mathcal{A}) \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,1}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n} \ .$$

It remains to bound the probability that the bad events occur in Game 5.

Event E_1' occurs when the adversary directly queries π at a point that is also queried as a result of a query to RKEnc (or RKDec). This situation is described in Section 4 as an inconsistency between $\mathsf{List}_\mathsf{P} := \mathsf{List}_\mathsf{P}^+ \cup \mathsf{List}_\mathsf{P}^-$ and $\mathsf{List}_\$:= \mathsf{List}_\$^+ \cup \mathsf{List}_\$^-$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_1$ against the OUP game with a list of length $2q_1q_{em}$. The adversary $\mathcal{B}_1$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(\phi_i^\pi, x_i \oplus a_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_1\} \cup \{(\phi_i^\pi, y_i \oplus b_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_1\}$, where x_i is the input to RKEnc resulting in output y_i on the i^{th} query (reversed for a query to RKDec) and a_j is the input to $\pi(1,\cdot,+)$ resulting in output b_j on the j^{th} query (similarly reversed for a query to $\pi(1,\cdot,-)$). If $\mathcal{A}$ sets Bad_1 with an DS_1 or IS_1 query, then $\mathcal{B}_1$ wins the OUP game with a tuple of the form $(\phi_i^\pi, x_i \oplus a_j)$ and if $\mathcal{A}$ sets Bad_1 with a query to $\mathsf{DS}_1{}^{-1}$ or $\mathsf{IS}_1{}^{-1}$ then $\mathcal{B}$ wins the OUP game with a tuple of the form $(\phi_i^\pi, y_i \oplus b_j)$. We therefore conclude that $\Pr[E_1'] \leq \mathbf{Adv}^{\mathsf{oup}}_{\Phi,1}(\mathcal{B}_1)$, where $\mathcal{B}_1$ outputs a list of length $2q_1q_{em}$.

Event E_2' occurs when an RKD function queries π at a point that is also queried as a result of a query to RKEnc (or RKDec). This situation is described in Section 4 as an inconsistency between $\mathsf{List}_\phi := \mathsf{List}_\phi^+ \cup \mathsf{List}_\phi^-$ and $\mathsf{List}_\$$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_2$ against the XQI game with a list of length $2q_{em}^2$. The adversary $\mathcal{B}_2$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(1, +, \phi_i^\pi, \phi_j^\pi, x_i) : 1 \leq i, j \leq q_{em}\} \cup \{(1, -, \phi_i^\pi, \phi_j^\pi, y_i) : 1 \leq i, j \leq q_{em}\}$. If $\mathcal{A}$ sets Bad_1 with a query to IS_1 or to DS_1 from an RKD function that queries $\pi(1,\cdot,+)$, then $\mathcal{B}_1$ wins the OUP game with a tuple of the form $(1, +, \phi_i^\pi, \phi_j^\pi, x_i)$ and if $\mathcal{A}$ sets Bad_1 with a query to $\mathsf{IS}_1{}^{-1}$ or $\mathsf{DS}_1{}^{-1}$ from an RKD function that queries $\pi(1,\cdot,-)$, then $\mathcal{B}$ wins the XQI game with a tuple of the form $(1, -, \phi_i^\pi, \phi_j^\pi, y_i)$. Therefore, $\Pr[E_2'] \leq \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,1}(\mathcal{B}_2)$, where $\mathcal{B}_2$ outputs a list of length $2q_{em}^2$.

Flag Bad_2' is set in a sub-case of the situation described in Section 4 as an inconsistency between List_P and $\mathsf{List}_\$$ or List_ϕ and $\mathsf{List}_\$$. It can occur in one of 16 different ways. Collisions between DS_1 and IS_1, $\mathsf{DS}_1{}^{-1}$ and $\mathsf{IS}_1{}^{-1}$, DS_1 and $\mathsf{IS}_1{}^{-1}$, or $\mathsf{DS}_1{}^{-1}$ and IS_1 can all set Bad_2 and each is counted twice, depending on the order of the queries. This gives 8 ways to set Bad_2, however the query to DS_1 can arise through a query by $\mathcal{A}$ or through ϕ^π, which gives 16 ways. In each case, we use a birthday-bound style argument and note that each pair (x, a) has at most a $1/(2^n - q - \sum_\phi q_1^\phi)$ chance of setting Bad_2' (if it is set via a call to IS_1 or $\mathsf{IS}_1{}^{-1}$ then it is set with probability $1/2^n$). Applying the union bound and recalling that q_{em} is the total number

of queries made to RKEnc and RKDec and thus to IS and ${\mathsf{IS}_1}^{-1}$ by $\mathcal{A}$ (similarly for q_1 and q_1^{ϕ}) gives that Bad_2' is set with probability at most $\frac{(q_1+\sum_{\phi} q_1^{\phi})q_{em}}{2^n-(q_1+\sum_{\phi} q_1^{\phi})}$.

Flag Bad_3 is set if a query to RKEnc (or RKDec) result in a value being queried to IS_1 (or ${\mathsf{IS}_1}^{-1}$) that is already in $\mathtt{I}_1$ (or ${\mathtt{I}_1}^{-1}$). This situation is described in Section 4 as an inconsistency between $\mathsf{List}_{\$}$ and $\mathsf{List}_{\$}$. We will use $\mathcal{A}$ that sets Bad_3 to create an adversary $\mathcal{B}_3$ against the XCF game. The adversary $\mathcal{B}_3$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(\phi_i^{\pi}, \phi_j^{\pi}, x_i \oplus x_j) : 1 \leq i < j \leq q_{em}\} \cup \{(\phi_i^{\pi}, \phi_j^{\pi}, y_i \oplus y_j) : 1 \leq i < j \leq q_{em}\}$. If $\mathcal{A}$ sets Bad_3 as a result a query to IS_1, then $\mathcal{B}_3$ wins the XCF game with a tuple of the form $(\phi_i^{\pi}, \phi_j^{\pi}, x_i \oplus x_j)$ and if $\mathcal{A}$ sets Bad_3 as a result a query to ${\mathsf{IS}_1}^{-1}$, then $\mathcal{B}_3$ wins the XCF game with a tuple of the form $(\phi_i^{\pi}, \phi_j^{\pi}, y_i \oplus y_j)$. Thus $\Pr[\mathsf{Bad}_3] \leq \mathbf{Adv}^{\mathsf{xcf}}_{\Phi,1}(\mathcal{B}_3)$, where $\mathcal{B}_3$ outputs a list of length at most q_{em}^2.

Flag Bad_4 is set in a sub-case of the situation described in Section 4 as an inconsistency between $\mathsf{List}_{\$}$ and $\mathsf{List}_{\$}$. Using similar reasoning as in the setting of Bad_2, it is set with probability at most $\frac{q_{em}^2}{2}\frac{1}{2^n}$.

As we have that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^{\pi}[1,1],\Phi,1}(\mathcal{A}) \leq{}& \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] \\ &+ \mathbf{Adv}^{\mathsf{cf}}_{\Phi,1}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}\ ,\end{aligned}$$

we may conclude that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^{\pi}[1,1],\Phi,1}(\mathcal{A}) \leq{}& \mathbf{Adv}^{\mathsf{oup}}_{\Phi,1}(\mathcal{B}_1) + \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,1}(\mathcal{B}_2) + \frac{q_{em}(q_1+\sum_{\phi} q_1^{\phi})}{2^n-(q_1+\sum_{\phi} q_1^{\phi})} \\ &+ 2\left(\mathbf{Adv}^{\mathsf{xcf}}_{\Phi,1}(\mathcal{B}_3) + \frac{q_{em}^2}{2^{n+1}}\right) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,1}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}\ ,\end{aligned}$$

where $\mathcal{B}_1$ outputs a list of length $2q_1q_{em}$, $\mathcal{B}_2$ a list of length $2q_{em}^2$, $\mathcal{B}_3$ a list of length q_{em}^2, and $\mathcal{B}_4$ a list of length at most q_{em}^2.

Game i:	$\mathrm{RKEnc}(\phi^\pi, x)$:	$\mathrm{RKDec}(\phi^\pi, y)$:	$\pi(1, a, +)$:
$k \leftarrow\!\!\$\ \mathcal{K}$ $b' \leftarrow\!\!\$\ \mathcal{A}^{\mathrm{RKEnc},\mathrm{RKDec},\pi}$ Return b'	$k' \leftarrow \phi^\pi(k)$ Return $k' \oplus \mathsf{IS}_1(k' \oplus x)$	$k' \leftarrow \phi^\pi(k)$ Return $k' \oplus \mathsf{IS}_1{}^{-1}(k' \oplus y)$	Return $\mathsf{DS}_1(a)$

Game 1:	Game 1a:	Game 2 | Game 3:	Game 4 | Game 5:
$\mathsf{DS}_1(a)$ If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ If $\mathtt{I}_1[a] \neq \bot$ Return $\mathtt{I}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{D}_1[a] \leftarrow b; \mathtt{D}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{DS}_1) \leftarrow \mathtt{Rng}(\mathsf{DS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{DS}_1) \leftarrow \mathtt{Dom}(\mathsf{DS}_1) \cup \{a\}$ Return $\mathtt{D}_1[a]$	$\mathsf{DS}_1(a)$: If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ If $\mathtt{I}_1[a] \neq \bot$ Return $\mathtt{I}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1)$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{D}_1[a] \leftarrow b; \mathtt{D}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{DS}_1) \leftarrow \mathtt{Rng}(\mathsf{DS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{DS}_1) \leftarrow \mathtt{Dom}(\mathsf{DS}_1) \cup \{a\}$ Return $\mathtt{D}_1[a]$	$\mathsf{DS}_1(a)$: If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ If $\mathtt{I}_1[a] \neq \bot$ $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ Return $\mathtt{I}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1)$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $\mathsf{Bad}_2 \leftarrow \mathsf{true}$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{D}_1[a] \leftarrow b; \mathtt{D}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{DS}_1) \leftarrow \mathtt{Rng}(\mathsf{DS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{DS}_1) \leftarrow \mathtt{Dom}(\mathsf{DS}_1) \cup \{a\}$ Return $\mathtt{D}_1[a]$	$\mathsf{DS}_1(a)$: If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ If $\mathtt{I}_1[a] \neq \bot$ $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1)$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $\mathsf{Bad}_2 \leftarrow \mathsf{true}$ $\mathtt{D}_1[a] \leftarrow b; \mathtt{D}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{DS}_1) \leftarrow \mathtt{Rng}(\mathsf{DS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{DS}_1) \leftarrow \mathtt{Dom}(\mathsf{DS}_1) \cup \{a\}$ Return $\mathtt{D}_1[a]$
$\mathsf{IS}_1(a)$: If $\mathtt{I}_1[a] \neq \bot$ Return $\mathtt{I}_1[a]$ If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{I}_1[a] \leftarrow b; \mathtt{I}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{IS}_1) \leftarrow \mathtt{Rng}(\mathsf{IS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{IS}_1) \leftarrow \mathtt{Dom}(\mathsf{IS}_1) \cup \{a\}$ Return $\mathtt{I}_1[a]$	$\mathsf{IS}_1(a)$: If $\mathtt{I}_1[a] \neq \bot$ Return $\mathtt{I}_1[a]$ If $\mathtt{D}_1[a] \neq \bot$ Return $\mathtt{D}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{IS}_1)$ If $b \in \mathtt{Rng}(\mathsf{DS}_1)$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{I}_1[a] \leftarrow b; \mathtt{I}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{IS}_1) \leftarrow \mathtt{Rng}(\mathsf{IS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{IS}_1) \leftarrow \mathtt{Dom}(\mathsf{IS}_1) \cup \{a\}$ Return $\mathtt{I}_1[a]$	$\mathsf{IS}_1(a)$: If $\mathtt{I}_1[a] \neq \bot$ Return $\mathtt{I}_1[a]$ If $\mathtt{D}_1[a] \neq \bot$ $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ Return $\mathtt{D}_1[a]$ $b \leftarrow\!\!\$\ \{0,1\}^n$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{IS}_1)$ If $b \in \mathtt{Rng}(\mathsf{DS}_1)$ $\mathsf{Bad}_2 \leftarrow \mathsf{true}$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_1, \mathsf{IS}_1)$ $\mathtt{I}_1[a] \leftarrow b; \mathtt{I}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{IS}_1) \leftarrow \mathtt{Rng}(\mathsf{IS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{IS}_1) \leftarrow \mathtt{Dom}(\mathsf{IS}_1) \cup \{a\}$ Return $\mathtt{I}_1[a]$	$\mathsf{IS}_1(a)$: If $\mathtt{I}_1[a] \neq \bot$ $\mathsf{Bad}_3 \leftarrow \mathsf{true}$ Return $\mathtt{I}_1[a]$ If $\mathtt{D}_1[a] \neq \bot$ $\mathsf{Bad}_1 \leftarrow \mathsf{true}$ $b \leftarrow\!\!\$\ \{0,1\}^n$ If $b \in \mathtt{Rng}(\mathsf{IS}_1)$ $\mathsf{Bad}_4 \leftarrow \mathsf{true}$ $b \leftarrow\!\!\$\ \{0,1\}^n \setminus \mathtt{Rng}(\mathsf{IS}_1)$ If $b \in \mathtt{Rng}(\mathsf{DS}_1)$ $\mathsf{Bad}_2 \leftarrow \mathsf{true}$ $\mathtt{I}_1[a] \leftarrow b; \mathtt{I}_1{}^{-1}[b] \leftarrow a$ $\mathtt{Rng}(\mathsf{IS}_1) \leftarrow \mathtt{Rng}(\mathsf{IS}_1) \cup \{b\}$ $\mathtt{Dom}(\mathsf{IS}_1) \leftarrow \mathtt{Dom}(\mathsf{IS}_1) \cup \{a\}$ Return $\mathtt{I}_1[a]$

Fig. 5. Sequence of games for the proof of Theorem 2. Oracles $\pi(1, \cdot, -)$, $\mathsf{DS}_1{}^{-1}$, and $\mathsf{IS}_1{}^{-1}$ are defined in a similar way to their corresponding forward oracles.

B Proof of Theorem 3: The Φ-RKCPA Security of $\mathsf{EM}^{\pi}[1,1,1]$

Proof. The proof follows a similar pattern to the proof of Theorem 2 and proceeds through four stages. In the first, $\mathcal{A}$ interacts with the public permutations and their inverses, plus the forward direction of the 2-round Even–Mansour scheme instantiated with the same permutations:

$$\pi(i,x,\sigma), \quad \mathsf{P}_2(\mathsf{P}_1(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ .$$

We then consider an environment in which P_2 is replaced by \$, a forgetful random oracle, for queries made to the Even–Mansour scheme:

$$\pi(i,x,\sigma), \quad \$(\mathsf{P}_1(x \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k)) \oplus \phi^{\pi}(k) \ ,$$

which is identical to

$$\pi(i,x,\sigma), \quad \$(\phi^{\pi}(k),x) \ .$$

Finally, we transition to games in which \$ is replaced by a keyed random function iF:

$$\pi(i,x,\sigma), \quad \mathsf{iF}(\phi^{\pi}(k),x) \ ,$$

and then by the ideal cipher iE:

$$\pi(i,x,\sigma), \quad \mathsf{iE}(\phi^{\pi}(k),x) \ .$$

We will now argue that the above changes alter $\mathcal{A}$'s winning probabilities negligibly and bound $\mathcal{A}$'s winning probability in terms of the conditions on Φ introduced in Section 5.

The first transition is analyzed via a series of games, given in Figure 6. These games include two intermediate transitions: in the first, P_2 is replaced with Q (a random permutation, chosen independently of π) for queries arising through RKEnc; in the second, Q is replaced with \$ (a forgetful random oracle). We identify the points at which these two intermediate transitions lead to inconsistencies, by setting Bad flags. In contrast to how the intuition behind this proof is described in Section 5, we defer the bounding of the probability bad events occurring during the first intermediate transition until after the second intermediate transition. The specification of ${\mathsf{DS}_1}^{-1}$ and ${\mathsf{IS}_1}^{-1}$ are omitted for conciseness; they are defined analogously to their respective forward oracles. Without loss of generality, we will assume that no adversary makes repeat or redundant queries – this assumption is needed in the transitions to and from the forgetful random oracles. Let S_i denote the event where the adversary outputs 1 in game i.

Game **0** is the RKA game augmented with a public permutation oracle (as described in Section 2), conditioned on $b = 1$. In this game, the adversary interacts with an oracle π realizing the two public permutations and the forward direction of the Even–Mansour construction, instantiated with π.

Game **1** is only syntactically different from Game 0. The queries to π are split into three groups. The first group is those made to $\pi(1,\cdot,\cdot)$, either by the adversary, by an RKD function, or as the result of an RKEnc query; these are answered by the sampling algorithms S_1 and ${\mathsf{S}_1}^{-1}$. The second group of queries consists of those made directly to $\pi(2,\cdot,\cdot)$, either by the adversary or by an RKD function, which are answered by the sampling algorithms DS_2 (or ${\mathsf{DS}_2}^{-1}$). The third group of queries are those queries to $\pi(2,\cdot,\cdot)$ which are made indirectly, through queries made to RKEnc; these queries are answered by IS_2. The oracles DS_2 and IS_2 maintain consistent lists $\mathtt{I}_2$ and $\mathtt{D}_2$; the lists used by inverse oracles are identical to the lists used by the corresponding forward oracles As this is a purely syntactic change, $\Pr[S_0] = \Pr[S_1]$.

Game **2** sets Bad_1 either if DS_2 is queried on a point already defined in $\mathtt{I}_2$ or if IS_2 is queried on a point already defined in $\mathtt{D}_2$ (and similarly for ${\mathsf{DS}_2}^{-1}$). This occurs either because $\mathcal{A}$ queries $\pi(2,\cdot,\cdot)$ directly at a point that is also queried to IS_2 through an indirect RKEnc query, or because an RKD function queries $\pi(2,\cdot,\cdot)$ at a point that is also queried to IS_2 through an RKEnc query (and similarly for ${\mathsf{DS}_2}^{-1}$). We will later bound the probability of this event in terms of the output unpredictability, first-order output unpredictability, xor query independence and first-order query independence of Φ. Game 2 sets Bad_2 if the value chosen at random for $\mathsf{DS}_2(a)$ is already defined in range of IS_2, or vice versa (and similarly for ${\mathsf{DS}_2}^{-1}$ queries and the domain of IS_2). This is necessary because in Game 1, for both DS_2 and IS_2, b may be sampled from $\{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_2,\mathsf{IS}_2)$ and we wish to completely decouple DS_1 (and ${\mathsf{DS}_1}^{-1}$) from IS_1. The code of S_1 and is unchanged and will remain so throughout this proof. The outputs of DS_2 (and ${\mathsf{DS}_2}^{-1}$) and IS_2 remain consistent and $\Pr[S_1] = \Pr[S_2]$.

Game 3 omits the boxed statements in Game 2 and so is identical to Game 2 unless one of Bad_1 or Bad_2 is set. In this game, the oracles DS_2 and IS_2 check consistency with their own lists, (and the list for their corresponding inverse oracle contains all the same entries as their list) but they may become inconsistent with each other. It is possible for Bad_1 to be set in two possible ways:

- E_1 is the event an adversary directly queries DS_2 (or ${\mathsf{DS}_2}^{-1}$) at a point coinciding with a point queried to (or output from) IS_2 via a query to RKEnc.
- E_2 is the event an RKD function queries DS_2 (or ${\mathsf{DS}_2}^{-1}$) at a point coinciding with a point queried to (or output from) IS_2 from a query to RKEnc.

We will analyze each of the ways that Bad_1 can be set below. Similarly, Bad_2 can be set either because of a query to DS_2 from $\mathcal{A}$, a query to DS_2 from ϕ^π, or a query to IS_2 due to a query to RKEnc (or similarly for ${\mathsf{DS}_2}^{-1}$); we consider all cases simultaneously below. In Game 3, the responses to RKEnc queries are completely decoupled from the responses to π queries, so we can consider that RKEnc uses Q to respond to queries and π uses P. We have that $\Pr[S_2] \leq \Pr[S_3] + \Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2]$.

Game 4 sets Bad_3 if two distinct queries to RKEnc result in the same value being queried to IS_2. As $\mathcal{A}$ makes no queries to RKDec, we only need to consider the possibility that Bad_3 is set as a result of a query to RKEnc. Game 4 chooses the response to IS_2 uniformly from $\{0,1\}^n$ and sets Bad_4 if this value is already in $\mathtt{Rng}(\mathsf{IS}_2)$. Game 4 is equivalent to Game 3 and, in particular, $\Pr[S_3] = \Pr[S_4]$.

Game 5 omits the boxed statements from Game 4 and so is identical to Game 4 unless Bad_3 or Bad_4 is set. Let E_1', E_2', Bad_2' represent events in Game 5 corresponding to events E_1, E_2, Bad_2 in Game 4, then $\Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2] \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4]$ In this game, calls to $\pi(2,\cdot,+)$ through RKEnc (which are answered by IS_1) are answered by a forgetful random oracle and so the ciphertexts are uniform and independent of the key and the plaintexts.

In Game 5, the adversary interacts with

$$\pi(i,x,\sigma), \quad \$(\mathsf{P}_1(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k) \ .$$

During the transitions to

$$\pi(i,x,\sigma), \quad \mathsf{iF}(\phi^\pi(k), x) \ ,$$

inconsistencies only arise if the adversary makes queries $(\phi_1^\pi, x_1) \neq (\phi_2^\pi, x_2)$, but where $(\phi_1^\pi(k), x_1) = (\phi_2^\pi(k), x_2)$. If an adversary $\mathcal{A}$ makes such a query, we can construct an adversary $\mathcal{B}_4$ which wins the CF game with a list of length at most $\frac{q_{em}^2}{2}$ as follows: $\mathcal{B}_4$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(\phi_i^\pi, \phi_j^\pi) \colon 1 \leq i < j \leq q_{em}\}$.

In the final transition, we switch from a random function to a random permutation (for each ϕ^π); the probability of an inconsistency arising in this step is bounded by $\frac{q_{em}^2}{2^n}$ [8].

Therefore we have that

$$\mathbf{Adv}^{\mathsf{rkcpa}}_{\mathsf{EM}^\pi[1,1,1],\Phi,2}(\mathcal{A}) \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,2}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}$$

It remains to bound the probability that the bad events occur in Game 5.

Event E_1' occurs when the adversary directly queries $\pi(2,\cdot,\cdot)$ at a point that is also queried as a result of a query to RKEnc. This situation is analogous to that described in Section 4 as an inconsistency between List_P and $\mathsf{List}_\$$. Although $\mathcal{A}$ makes no ${\mathsf{IS}_2}^{-1}$ queries, it is possible to trigger E_2 with a query to ${\mathsf{DS}_2}^{-1}$. We will use $\mathcal{A}$ to create adversaries $\mathcal{B}_{1a}$ and $\mathcal{B}_{1b}$ against the OUP1 and OUP games respectively, both with lists of length $q_2 q_{em}$: The adversary $\mathcal{B}_{1a}$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(1,+,\phi_i^\pi, x_i, a_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_2\}$. The adversary $\mathcal{B}_{1b}$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(\phi_i^\pi, y_i \oplus b_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_2\}$. If $\mathcal{A}$ can set Bad by querying the permutation at a point that is also queried as a result of a query to RKEnc, then either the adversary $\mathcal{B}_{1a}$ will win the OUP1 game or $\mathcal{B}_{1b}$ will win the OUP game. We therefore conclude that $\Pr[E_1'] \leq \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,2}(\mathcal{B}_{1a}) + \mathbf{Adv}^{\mathsf{oup}}_{\Phi,2}(\mathcal{B}_{1b})$, where $\mathcal{B}_{1a}$ and $\mathcal{B}_{1b}$ both output a list of length $q_2 q_{em}$.

Event E_2' occurs when an RKD function queries $\pi(2,\cdot,\cdot)$ at a point that is also queried to (or returned from) IS_2 as a result of a query to RKEnc. This situation is analogous to that described in Section 4 as an inconsistency between List_ϕ and $\mathsf{List}_\$$. Although $\mathcal{A}$ makes no ${\mathsf{IS}_2}^{-1}$ queries, it is possible to trigger E_2 with a query to ${\mathsf{DS}_2}^{-1}$. We will use $\mathcal{A}$ to create adversaries $\mathcal{B}_{2a}$ and $\mathcal{B}_{2b}$ against the QI1 and XQI games respectively, both with lists of length q_{em}^2. The adversary $\mathcal{B}_{2a}$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(2,+,\phi_i^\pi, x_i, \phi_j^\pi) : 1 \leq i,j \leq q_{em}\}$;

the adversary $\mathcal{B}_{2b}$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(2, -, \phi_i^\pi, \phi_j^\pi, y_i) : 1 \leq i, j \leq q_{em}\}$. If $\mathcal{A}$ can set Bad by causing an RKD function to query the permutation at a point that is also queried (or returned) as a result of a query to RKEnc, then either $\mathcal{B}_{2a}$ will win the $\mathsf{QI1}$ game or $\mathcal{B}_{2b}$ will win the QI game. Although $\mathcal{A}$ makes no ${\mathsf{IS}_2}^{-1}$ queries, it is possible to trigger E_2 with a query to ${\mathsf{DS}_2}^{-1}$ and so we must include tuples of the form $(\phi_i^\pi, \phi_j^\pi, y_i)$. Therefore we can conclude that $\Pr[E_2'] \leq \mathbf{Adv}^{\mathsf{qi1}}_{\Phi,2}(\mathcal{B}_{2a}) + \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,2}(\mathcal{B}_{2b})$, where $\mathcal{B}_{2a}$ and $\mathcal{B}_{2b}$both output lists of length q_{em}^2.

Flag Bad_2' is set in a situation analogous to a sub-case of that described in Section 4 as an inconsistency between List_P and $\mathsf{List}_\$$ or List_ϕ and $\mathsf{List}_\$$. It can occur in one of 8 different ways. Collisions between DS_2 and IS_2 or ${\mathsf{DS}_2}^{-1}$ and IS_2 can both set Bad_2 and each is counted twice, depending on the order of the queries. This gives 4 ways to set Bad_2, however the query to DS_2 can arise through a query by $\mathcal{A}$ or through ϕ^π, which gives 8 ways. In each case, we use a birthday-bound argument and note that each pair (x, a) sets Bad_2' with probability at most $1/(2^n - q_2 - \sum_\phi q_2^\phi)$ (if it is set via a call to IS_2 then it is set with probability $1/2^n$). Applying the union bound and recalling that q_{em} is the total number of queries made to RKEnc (and thus to IS) by $\mathcal{A}$ (and similarly for q_2 and q_2^ϕ) gives that Bad_2' is set with probability at most $\frac{(q_2+\sum_\phi q_2^\phi)q_{em}}{2^n-(q_2+\sum_\phi q_2^\phi)}$.

Flag Bad_3 is set when two queries to RKEnc result in IS_2 being queried at the same point. This situation is analogous to that described in Section 4 as an inconsistency between $\mathsf{List}_\$$ and $\mathsf{List}_\$$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_3$ against the $\mathsf{CF1}$ property of Φ. The adversary $\mathcal{B}_3$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(1, +, \phi_i^\pi, x_i, \phi_j^\pi, x_j) : 1 \leq i < j \leq q_{em}\}$. Note that, as $\mathcal{A}$ makes no RKDec queries, it is unable to set Bad_3 with an RKDec query and so we do not need to include tuples of the form $(\phi_1^\pi, y_1, \phi_2^\pi, y_2)$ in List. Thus $\Pr[\mathsf{Bad}_3] \leq \mathbf{Adv}^{\mathsf{cf1}}_{\Phi,2}(\mathcal{B}_3)$, where $\mathcal{B}_3$ outputs a list of length $\frac{q_{em}(q_{em}-1)}{2}$.

Flag Bad_4 is set in a situation analogous to a sub-case of the that described in Section 4 as an inconsistency between $\mathsf{List}_\$$ and $\mathsf{List}_\$$. Using similar reasoning as in the setting of Bad_2, it is set with probability at most $\frac{q_{em}^2}{2}\frac{1}{2^n}$.

As we have that

$$\mathbf{Adv}^{\mathsf{rkcpa}}_{\mathsf{EM}^\pi[1,1,1],\Phi,2}(\mathcal{A}) \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,2}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}$$

we may conclude that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcpa}}_{\mathsf{EM}^\pi[1,1,1],\Phi,2} \leq & \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,2}(\mathcal{B}_{1a}) + \mathbf{Adv}^{\mathsf{oup}}_{\Phi,2}(\mathcal{B}_{1b}) + \mathbf{Adv}^{\mathsf{qi1}}_{\Phi,2}(\mathcal{B}_{2a}) + \mathbf{Adv}^{\mathsf{xqi}}_{\Phi,2}(\mathcal{B}_{2b}) \\ & + \frac{q_{em}(q_2 + \sum_\phi q_2^\phi)}{2^n - (q_2 + \sum_\phi q_2^\phi)} + 2\left(\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,2}(\mathcal{B}_3) + \frac{q_{em}^2}{2^{n+1}}\right) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,2}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}\ ,\end{aligned}$$

where $\mathcal{B}_{1a}$ and $\mathcal{B}_{1b}$ output lists of length $q_2 q_{em}$, $\mathcal{B}_{2a}$ and $\mathcal{B}_{2b}$ lists of length q_{em}^2, $\mathcal{B}_3$ a list of length q_{em}^2, and $\mathcal{B}_4$ a list of length at most q_{em}^2.

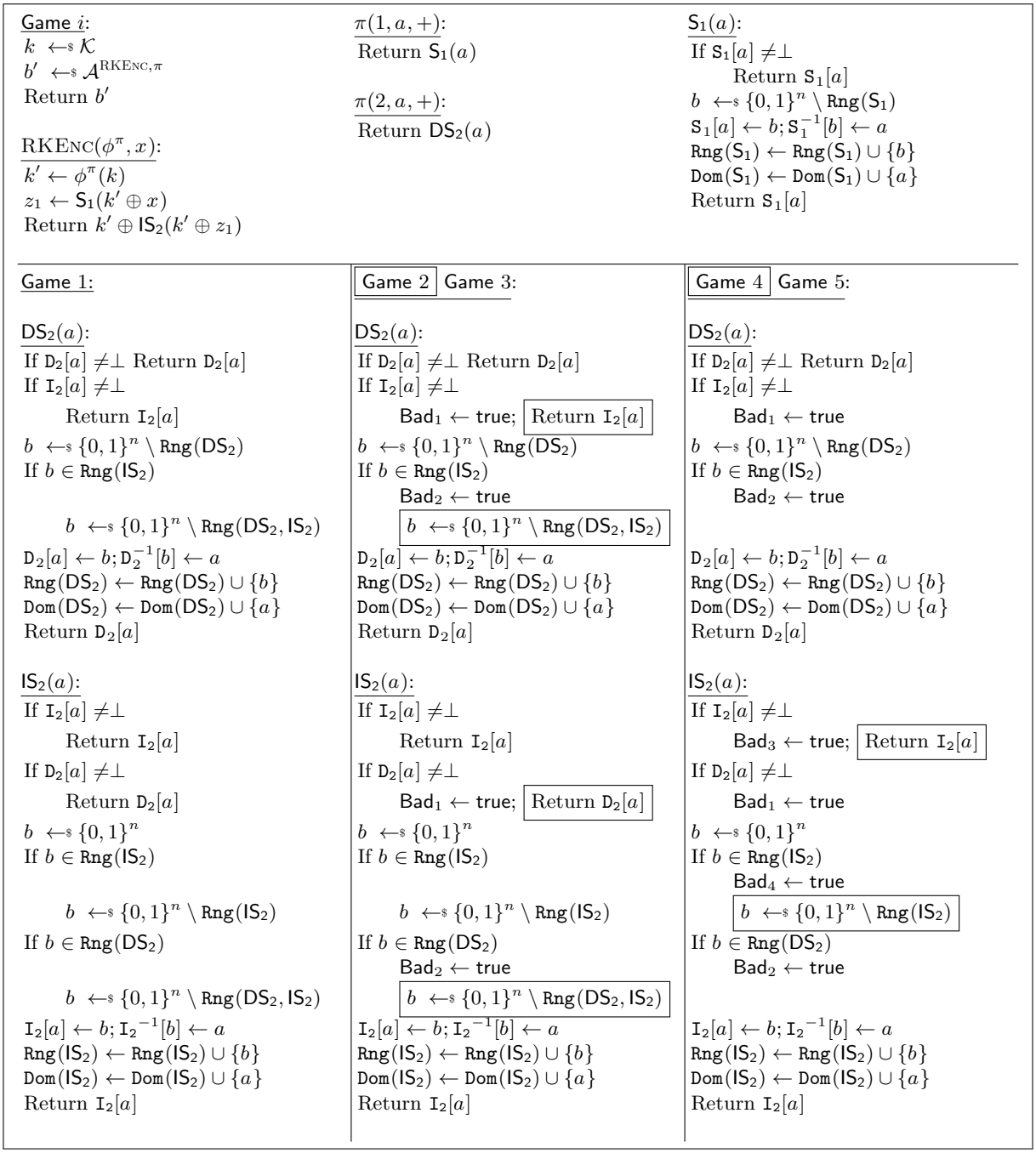

Fig. 6. Sequence of games for the proof of Theorem 3. Oracles $\pi(i, \cdot, -)$, ${\mathsf{S}_1}^{-1}$, and ${\mathsf{DS}_2}^{-1}$ are defined in a similar way to their corresponding forward oracles.

C Proof of Theorem 4: The Φ-RKCCA Security of $\mathsf{EM}^{\pi}[1,1,1,1]$

Proof. The proof follows a similar pattern to the proof of Theorems 2 and 3 and we proceed through four stages. In the first, $\mathcal{A}$ interacts with the public permutations and their inverses, plus the 3-round Even–Mansour scheme instantiated with the same permutations:

$$\pi(i,x,\sigma),\quad \mathsf{P}_3(\mathsf{P}_2(\mathsf{P}_1(x\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k),\quad \mathsf{P}_1^{-1}(\mathsf{P}_2^{-1}(\mathsf{P}_3^{-1}(x\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k)\ .$$

We then consider an environment in which P_2 and P_2^{-1} are replaced by \$, a forgetful random oracle, for queries made to the Even–Mansour scheme:

$$\pi(i,x,\sigma),\quad \mathsf{P}_3(\$(\mathsf{P}_1(x\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k),\quad \mathsf{P}_1^{-1}(\$(\mathsf{P}_3^{-1}(x\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k))\oplus\phi^\pi(k)\ ,$$

which is identical to

$$\pi(i,x,\sigma),\quad \$(\phi^\pi(k),x),\quad \$(\phi^\pi(k),x)\ .$$

Finally, we transition to games in which \$ oracles are replaced by keyed random functions iF and iC,

$$\pi(i,x,\sigma),\quad \mathsf{iF}(\phi^\pi(k),x),\quad \mathsf{iC}(\phi^\pi(k),x)\ ,$$

and then by the ideal cipher $(\mathsf{iE},\mathsf{iD})$,

$$\pi(i,x,\sigma),\quad \mathsf{iE}(\phi^\pi(k),x),\quad \mathsf{iD}(\phi^\pi(k),x)\ .$$

We will now argue that the above changes alter $\mathcal{A}$'s winning probabilities negligibly and bound $\mathcal{A}$'s winning probability in terms of the conditions on Φ introduced in Section 5.

The first transition is analyzed via a series of games, given in Figure 7. These games include two intermediate transitions: in the first, P_2 is replaced with Q (a random permutation, chosen independently of π) for queries arising through RKEnc (or RKDec); in the second, Q is replaced with \$ (a forgetful random oracle). We identify the points at which these two intermediate transitions lead to inconsistencies, by setting Bad flags. We omit a specification of the inverse oracles for conciseness; they are defined analogously to their respective forward oracles. Without loss of generality, we will assume that no adversary makes repeat or redundant queries – this assumption is needed in the transitions to and from the forgetful random oracles. Let S_i denote the event where the adversary outputs 1 in game i.

Game 0 is the RKA game augmented with a public permutation oracle (as described in Section 2), conditioned on $b = 1$. In this game, the adversary interacts with an oracle realizing the three public permutations π and the Even–Mansour construction instantiated with π.

Game 1 is only syntactically different from Game 0. Sampling algorithms S_1 and S_3 (and their inverses) are introduced to respond to queries made to $\pi(1,\cdot,\cdot)$ and $\pi(3,\cdot,\cdot)$. Queries to $\pi(2,\cdot,\cdot)$ are split into two groups: those made directly to π, either by the adversary or by an RKD function, which are answered by the sampling algorithm DS_2 (or $\mathsf{DS}_2{}^{-1}$) and those made indirectly through queries made to RKEnc (or RKDec), which are answered by IS_2 (or $\mathsf{IS}_2{}^{-1}$). The oracles DS_2 and IS_2 maintain consistent lists, $\mathtt{I}_2$ and $\mathtt{D}_2$; the lists used by inverse oracles are identical to the lists used by the corresponding forward oracles. As this is a purely syntactic change, $\Pr[S_0] = \Pr[S_1]$.

Game 2 sets Bad_1 either if DS_2 is queried on a point already defined in $\mathtt{I}_2$ or if IS_2 is queried on a point already defined in $\mathtt{D}_2$ (and similarly for the inverse oracles). This occurs either because $\mathcal{A}$ queries $\pi(2,\cdot,\cdot)$ directly at a point that is also queried to $\pi(2,\cdot,\cdot)$ through an indirect RKEnc query, or because an RKD function queries $\pi(2,\cdot,\cdot)$ at a point that is also queried to $\pi(2,\cdot,\cdot)$ through an RKEnc query (and similarly for the inverse oracles). We will later bound the probability of this event in terms of the first-order output unpredictability and first-order query independence of Φ. Game 2 sets Bad_2 if the value chosen at random for $\mathsf{IS}_2(a)$ is already defined in range of DS_2, or vice versa (and similarly for the inverse queries and the domain of IS_2 or DS_2). This is necessary because in Game 1, for both DS_2 and IS_2, b is sampled from $\{0,1\}^n \setminus \mathtt{Rng}(\mathsf{DS}_2,\mathsf{IS}_2)$ whereas our objective in Game 3 is to ensure that DS_2 (and $\mathsf{DS}_2{}^{-1}$) are independent of IS_2 (and $\mathsf{IS}_2{}^{-1}$). The code of S_1 and S_3 remains unchanged throughout this proof. The outputs of DS_2 and IS_2 remain consistent and $\Pr[S_1] = \Pr[S_2]$.

Game **3** omits the boxed statements in Game 2 and so is identical to Game 2 unless one of Bad_1 or Bad_2 is set. In this game, the oracles DS_2 and IS_2 check consistency with their own lists (and the list for their corresponding inverse oracle contains all the same entries as their list), but they may become inconsistent with each other. It is possible for Bad_1 to be set in two possible ways:

- E_1 is the event an adversary directly queries DS_2 at a point coinciding with a point queried to IS_2 from a query to RKEnc (or comparable conditions resulting from queries to inverse oracles).
- E_2 is the event an RKD function queries DS_2 at a point coinciding with a point queried to IS_2 from a query to RKEnc (or comparable conditions resulting from queries to inverse oracles).

We will analyze each of the ways that Bad_1 can be set below. Similarly, Bad_2 can be set either because of a query to DS_2 from $\mathcal{A}$, a query to DS_2 from ϕ^π, or from a query to IS_2 due to a query to RKEnc (or similarly for the corresponding inverse oracles); we consider all cases simultaneously below. In Game 3, the responses to RKEnc queries are completely decoupled from the responses to π queries, so we can consider that RKEnc uses Q to respond to queries and π uses P. We have that $\Pr[S_2] \leq \Pr[S_3] + \Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2]$.

Game **4** sets Bad_3 if a query to RKEnc (or RKDec) results in a value being queried to IS_2 (or $\mathsf{IS}_2{}^{-1}$) that is already in $\mathtt{I}_2$ (or $\mathtt{I}_2{}^{-1}$). The flag Bad_4 can be set in four ways (as a result of two queries to either of IS_2 and $\mathsf{IS}_2{}^{-1}$, plus two 'mixed cases' with one query to each of IS_2 and $\mathsf{IS}_2{}^{-1}$); we consider each of these cases when we analyze the probability of setting bad events below. Game 4 chooses the response to IS_2 uniformly from $\{0,1\}^n$ and sets Bad_4 if this value is already in $\mathtt{Rng}(\mathsf{IS}_2)$. Game 4 is equivalent to Game 3 and, in particular, $\Pr[S_3] = \Pr[S_4]$.

Game **5** omits the boxed statements from Game 4 and so is identical to Game 4 unless Bad_3 or Bad_4 is set. Let E_1', E_2', Bad_2' represent events in Game 5 corresponding to events E_1, E_2, Bad_2 in Game 4, then $\Pr[E_1 \vee E_2 \vee \mathsf{Bad}_2] \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4]$ In this game, calls to $\pi(2,\cdot,\cdot)$ through RKEnc (RKDec), which are answered by IS_2 ($\mathsf{IS}_2{}^{-1}$) are answered by a forgetful random oracle and so the ciphertexts (plaintexts) are uniform and independent of the key and the plaintext (ciphertexts).

In Game 5, the adversary interacts with

$$\pi(i,x,\sigma), \quad \mathsf{P}_3(\$(\mathsf{P}_1(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k), \quad \mathsf{P}_1^{-1}(\$(\mathsf{P}_3^{-1}(x \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k)) \oplus \phi^\pi(k) \ .$$

During the transitions to

$$\pi(i,x,\sigma), \quad \mathsf{iF}(\phi^\pi(k),x), \quad \mathsf{iC}(\phi^\pi(k),x) \ ,$$

inconsistencies only arise if the adversary makes queries $(\phi_1^\pi, x_1) \neq (\phi_2^\pi, x_2)$, but where $(\phi_1^\pi(k), x_1) = (\phi_2^\pi(k), x_2)$. If an adversary $\mathcal{A}$ makes such a query, we can construct an adversary $\mathcal{B}_4$ which wins the CF game with a list of length at most $\frac{q_{em}^2}{2}$ as follows: $\mathcal{B}_4$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(\phi_i^\pi, \phi_j^\pi) \colon 1 \leq i < j \leq q_{em}\}$.

In the final transition, we switch from a random function to a random permutation (for each ϕ^π); the probability of an inconsistency arising in this step is bounded by $\frac{q_{em}^2}{2^n}$ [8].

Therefore we have that

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^\pi[1,1,1,1,1],\Phi,3}(\mathcal{A}) \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,3}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}$$

It remains to bound the probability that the bad events occur in Game 5.

Event E_1' occurs when the adversary directly queries $\pi(2,\cdot,\cdot)$ at a point that is also queried to IS_2 (or $\mathsf{IS}_2{}^{-1}$) a result of a query to RKEnc (or RKDec). This situation is analogous to that described in Section 4 as an inconsistency between List_P and $\mathsf{List}_\$$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_1$ against the OUP1 game with a list of length $2q_2q_{em}$. The adversary $\mathcal{B}_1$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(1,+,\phi_i^\pi, x_i, a_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_2\} \cup \{(3,-,\phi_i^\pi, y_i, b_j) : 1 \leq i \leq q_{em}, 1 \leq j \leq q_2\}$. If $\mathcal{A}$ sets Bad_1 with an RKEnc or DS_2 query, then $\mathcal{B}_1$ wins the OUP1 game with a tuple of the form $(1,+,\phi_i^\pi, x_i, a_j)$ and if $\mathcal{A}$ sets Bad_1 with a query to RKDec or $\mathsf{DS}_2{}^{-1}$ then $\mathcal{B}$ wins the OUP1 game with a tuple of the form $(3,-,\phi_i^\pi, y_i, b_j)$. We therefore conclude that $\Pr[E_1'] \leq \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,3}(\mathcal{B}_1)$, where $\mathcal{B}_1$ outputs a list of length $2q_2q_{em}$.

Event E_2' occurs when an RKD function queries the $\pi(2,\cdot,\cdot)$ at a point that is also queried as a result of a query to RKEnc. This situation is analogous to that described in Section 4 as an inconsistency between List_ϕ and $\mathsf{List}_\$$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_2$ against the QI1 game with a list of length $2q_{em}^2$. The adversary $\mathcal{B}_2$ runs $\mathcal{A}$ and outputs $\mathsf{List} = \{(1,+,\phi_i^\pi, x_i, \phi_j^\pi) : 1 \leq i,j \leq q_{em}\} \cup \{(3,-,\phi_i^\pi, y_i, \phi_j^\pi) : 1 \leq i,j \leq q_{em}\}$. If

$\mathcal{A}$ can set Bad by causing an RKD function to query the permutation at a point that is also queried as a result of a query to IS_2 or DS_2, then the adversary $\mathcal{B}_2$ will win the $\mathsf{QI1}$ game with a tuple of the form $(1, +, \phi_i^\pi, x_i, \phi_j^\pi)$ and if $\mathcal{A}$ sets Bad_1 with a query to ${\mathsf{IS}_2}^{-1}$ or ${\mathsf{DS}_2}^{-1}$ then $\mathcal{B}$ wins the $\mathsf{QI1}$ game with a tuple of the form $(3, -, \phi_i^\pi, y_i, \phi_j^\pi)$. Therefore we can conclude that $\Pr[E_2'] \leq \mathbf{Adv}^{\mathsf{qi1}}_{\Phi,3}(\mathcal{B}_2)$, where $\mathcal{B}_2$ outputs a list of length $2q_{em}^2$.

Flag Bad_2' is set in a situation analogous to a sub-case of that described in Section 4 as an inconsistency between $\mathsf{List}_{\mathsf{P}}$ and $\mathsf{List}_{\$}$ or List_{ϕ} and $\mathsf{List}_{\$}$. It can occur in one of 16 different ways. Collisions between DS_2 and IS_2, ${\mathsf{DS}_2}^{-1}$ and ${\mathsf{IS}_2}^{-1}$, DS_2 and ${\mathsf{IS}_2}^{-1}$, or ${\mathsf{DS}_2}^{-1}$ and IS_2 can all set Bad_2 and each is counted twice, depending on the order of the queries. This gives 8 ways to set Bad_2, however the query to DS_2 can arise through a query by $\mathcal{A}$ or through ϕ^π, which gives 16 ways. In each case, we use a birthday-bound style argument and note that each pair (x, a) sets Bad_2' with probability at most $1/(2^n - q_2 - \sum_\phi q_2^\phi)$ (if it is set via a call to IS_2 or ${\mathsf{IS}_2}^{-1}$ then it is set with probability $1/2^n$). Applying the union bound and recalling that q_{em} is the total number of queries made to RKEnc (and thus to IS) by $\mathcal{A}$ (and similarly for q_2 and q_2^ϕ) gives that Bad_2' is set with probability at most $\frac{(q_2+\sum_\phi q_2^\phi)q_{em}}{2^n-(q_2+\sum_\phi q_2^\phi)}$.

Flag Bad_3 is set if a query to RKEnc (or RKDec) results in a value being queried to IS_2 (or ${\mathsf{IS}_2}^{-1}$) that is already in $\mathtt{I}_2$ (or ${\mathtt{I}_2}^{-1}$). This situation is analogous to that described in Section 4 as an inconsistency between $\mathsf{List}_{\$}$ and $\mathsf{List}_{\$}$. We will use $\mathcal{A}$ to create an adversary $\mathcal{B}_3$ against the $\mathsf{CF1}$ property of Φ. The adversary $\mathcal{B}_3$ runs $\mathcal{A}$ and then outputs $\mathsf{List} = \{(1, +, \phi_i^\pi, x_i, \phi_j^\pi, x_j) : 1 \leq i < j \leq q_{em}\} \cup \{(3, -, \phi_i^\pi, y_i, \phi_j^\pi, y_j) : 1 \leq i < j \leq q_{em}\}$ If $\mathcal{A}$ sets Bad_3 with query to IS_2, then $\mathcal{B}_3$ wins the $\mathsf{CF1}$ game with a tuple of the form $(1, +, \phi_i^\pi, x_i, \phi_j^\pi, x_j)$ and if $\mathcal{A}$ sets Bad_3 with query to ${\mathsf{IS}_2}^{-1}$, then $\mathcal{B}_3$ wins the $\mathsf{CF1}$ game with a tuple of the form $(3, -, \phi_i^\pi, y_i, \phi_j^\pi, y_j)$. Thus $\Pr[\mathsf{Bad}_3] \leq \mathbf{Adv}^{\mathsf{cf1}}_{\Phi,3}(\mathcal{B}_3)$, where $\mathcal{B}_3$ outputs a list of length at most q_{em}^2.

Flag Bad_4 is set in a situation analogous to a sub-case of the that described in Section 4 as an inconsistency between $\mathsf{List}_{\$}$ and $\mathsf{List}_{\$}$. Using similar reasoning as in the setting of Bad_2, it is set with probability at most $\frac{q_{em}^2}{2}\frac{1}{2^n}$.

As we have that

$$\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^\pi[1,1,1,1],\Phi,3}(\mathcal{A}) \leq \Pr[E_1' \vee E_2' \vee \mathsf{Bad}_2'] + 2\Pr[\mathsf{Bad}_3 \vee \mathsf{Bad}_4] + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,3}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}$$

we may conclude that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{rkcca}}_{\mathsf{EM}^\pi[1,1,1,1],\Phi,3}(\mathcal{A}) \leq & \mathbf{Adv}^{\mathsf{oup1}}_{\Phi,3}(\mathcal{B}_1) + \mathbf{Adv}^{\mathsf{xqi1}}_{\Phi,3}(\mathcal{B}_2) + 2\frac{q_{em}(q_2 + \sum_\phi q_2^\phi)}{2^n - (q_2 + \sum_\phi q_2^\phi)} \\ & + 2\left(\mathbf{Adv}^{\mathsf{cf1}}_{\Phi,3}(\mathcal{B}_3) + \frac{q_{em}^2}{2^{n+1}}\right) + \mathbf{Adv}^{\mathsf{cf}}_{\Phi,3}(\mathcal{B}_4) + \frac{q_{em}^2}{2^n}\end{aligned}$$

where $\mathcal{B}_1$ outputs a list of length $2q_2q_{em}$, $\mathcal{B}_2$ a list of length $2q_{em}^2$, $\mathcal{B}_3$ a list of length q_{em}^2, and $\mathcal{B}_4$ a list of length at most q_{em}^2.

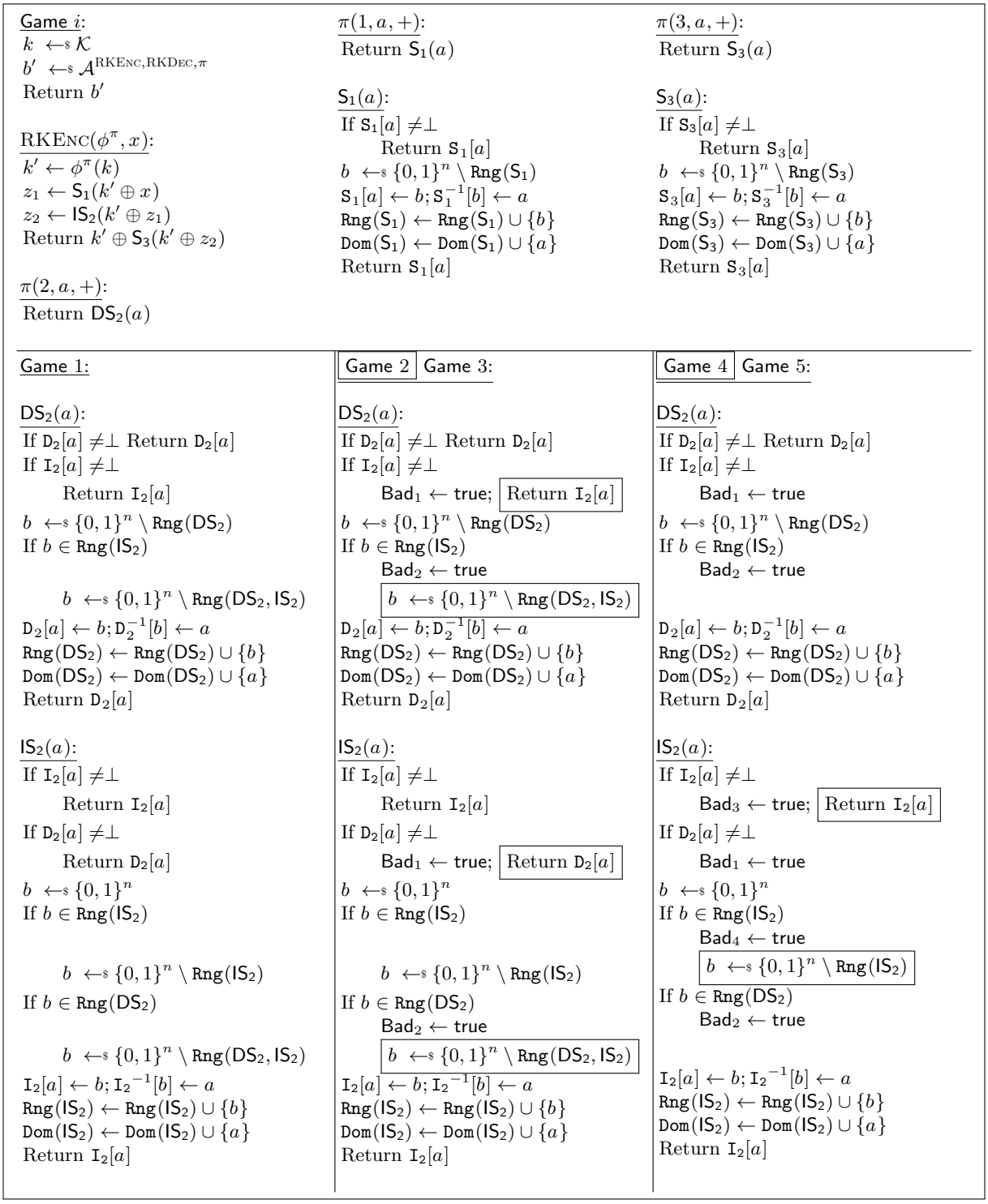

Fig. 7. Sequence of games for the proof of Theorem 4. Oracles RKDec, $\pi(i, \cdot, -)$, ${\mathsf{S}_1}^{-1}$, ${\mathsf{S}_3}^{-1}$, ${\mathsf{DS}_2}^{-1}$, and ${\mathsf{IS}_2}^{-1}$ are defined in a similar way to their corresponding forward oracles.