Multi-key Security: The Even-Mansour Construction Revisited

Nicky Mouha[1,2] and Atul Luykx[1(✉)]

[1] Department of Electrical Engineering-ESAT/COSIC, KU Leuven, Leuven and iMinds, Ghent, Belgium
{Nicky.Mouha,Atul.Luykx}@esat.kuleuven.be
[2] Project-team SECRET, Inria, Paris, France

Abstract. At ASIACRYPT 1991, Even and Mansour introduced a block cipher construction based on a single permutation. Their construction has since been lauded for its simplicity, yet also criticized for not providing the same security as other block ciphers against generic attacks. In this paper, we prove that if a small number of plaintexts are encrypted under multiple independent keys, the Even-Mansour construction surprisingly offers similar security as an ideal block cipher with the same block and key size. Note that this *multi-key setting* is of high practical relevance, as real-world implementations often allow frequent rekeying. We hope that the results in this paper will further encourage the use of the Even-Mansour construction, especially when a secure and efficient implementation of a key schedule would result in significant overhead.

Keywords: Even-Mansour · Multi-key setting · Broadcast attack · Related-key setting

1 Introduction

Modern block cipher design is based on the concept of iterating a round function [44]. This round function typically consists of a subkey addition followed by an unkeyed invertible function. All commonly-used block ciphers, including the DES [46] and AES [23] standards, follow this design strategy.

As such, the design of an iterated block cipher consists of two parts: the design of a round function, and a key schedule to generate the subkeys for every round. Although round function design seems to be a relatively well-understood problem, this is much less the case for the key schedule. For example, Rijmen and Daemen already stated in the AES design book that: "*There is no consensus on the criteria that a key schedule must satisfy*" [23, p. 77]. This fact has been repeated many times since, for example in the SHA-3 finalist Grøstl design document [32, p. 5]. In particular, there seems to be no consensus on whether a "strong" or a "simple" key schedule is required, a choice which appears to depend on the application.

An argument for a "simple" key schedule is that keys should be chosen uniformly at random from the entire key space anyway, in order to avoid a speed-up

R. Gennaro and M. Robshaw (Eds.): CRYPTO 2015, Part I, LNCS 9215, pp. 209–223, 2015.
DOI: 10.1007/978-3-662-47989-6_10

of brute-force attacks due to low key entropy. As a result, attacks based on weak keys [25] and known keys [42] are no longer applicable. Similarly, when multiple keys are used, they should be chosen independently to prevent a compromise of one key helping the recovery of other keys. This avoids attacks based on related keys [9–11].

Proponents of a "strong" key schedule point out that in practice, keys may not always be chosen independently from a uniformly random distribution. The cause of this could be a weak protocol, a programming error or an insecure implementation.

Complexity, however, always comes at a cost. It makes cryptosystems more difficult to design, to implement and to analyze. Hence, we argue for a block cipher with a simple key schedule, combined with the use of a secure key derivation function (KDF) [20] for secret-key applications. A KDF can be as simple as using the same block cipher to encrypt a counter, in which case the implementation overhead is minimal. For a theoretical treatment of KDFs, we refer to [1].

Although the use of a KDF avoids attacks on weak keys, known keys and related keys, it cannot prevent multi-key attacks [12,14,24,38]: a plaintext may be encrypted under multiple independent keys. Often overlooked by block cipher designers, multi-key attacks are highly relevant in practice (cf. Sect. 2).

This leads us to the following open problem, formulated by Daemen and Rijmen in 2012 [24]. In their paper, they point out: "A scenario where the adversary can query the block cipher under related keys, or even multiple keys, inevitably leads to security erosion". Paraphrased, concerning the multi-key setting they ask: "Can we design a secure block cipher with a lighter key schedule and higher key agility if related-key security is not required?"

In this paper, we give a positive answer to their question. Surprisingly, one of the simplest block ciphers, the single-key Even-Mansour construction[1] [26,29,30], shown in Fig. 1, offers similar security to an ideal block cipher when a small number of plaintexts can be queried under many keys.

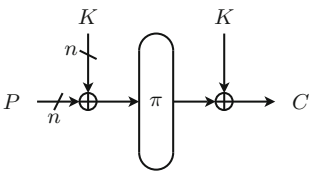

Fig. 1. The Even-Mansour construction.

Outline. The single-key, related-key and multi-key attack settings are discussed in Sect. 2, from a theoretical as well as a practical perspective. In Sect. 3, we prove tight bounds for the security of an Even-Mansour block cipher and of an ideal

[1] Throughout this paper, we will always refer to the single-key variant of the Even-Mansour construction, that is, using only a single n-bit key K.

block cipher in the multi-key setting. The relevance of our observations for the security and efficiency of block cipher implementations is discussed in Sect. 4. We conclude the paper in Sect. 5.

2 Attack Settings

2.1 Three Attack Settings

Single-Key Setting. One key K is chosen uniformly at random from the key space in the *single-key setting*, also referred to as the *fixed-key setting*. The adversary can then make encryption and decryption queries to block cipher E, all under the same key K.

Related-Key Setting. In the *related-key setting*, the adversary can perform encryption and decryption queries to block cipher E under keys K_i. Each key i satisfies the relationship $K_i = \Phi_i(K)$, where K is secret, but the functions Φ_i are chosen by the adversary. To avoid that not every block cipher E is vulnerable to a related-key attack, restrictions are necessary on the functions Φ_i as explained in [9].

Security against related-key attacks is often considered in the design of a block cipher. For example, it was a stated design goal for the AES block cipher [23], although it was shown that AES is not secure against related-key attacks [15,16].

Furthermore, it should be noted that certain commonly-used algorithms, including DES and Triple-DES, are trivially insecure against related-key attacks. For DES [46] and Triple-DES [6], this is an immediate result of its complementation property [36]: $\overline{E_K(P)} = E_{\overline{K}}(\overline{P})$, where $\overline{x}$ represents the bitwise complement of x.

It is difficult to say whether related-key security should be a requirement, as this depends on the protocol in which the cryptosystem is used. Nevertheless, it seems fair to point out that protocol designers should not assume related-key security, given that several commonly-used designs are (sometimes trivially) insecure in this setting.

Multi-key Setting. In the *multi-key setting*, the adversary can query encryption and decryption queries under keys K_i, where all K_i are independently chosen, uniformly at random. The multi-key setting can be seen as a generalization of the *multi-user setting* of Chatterjee et al. [19], where encryption queries of only one plaintext P are allowed under keys K_i.[2] This multi-user setting is then

[2] In the model of Chatterjee et al. [19], an adversary can also *corrupt* any user of its choosing, meaning that their key is given to the adversary. The goal of the adversary is then to win the game for any uncorrupted user. Although it is straightforward to take this refinement into account, we decided not to do this for the clarity of our exposition.

again a further generalization of the *broadcast setting* of Mantin and Shamir [45], where the plaintext P is unknown to the attacker.

Every attack in the broadcast setting also leads to an attack in the multi-user setting, and every multi-user attack is also a multi-key attack. We will therefore use the multi-key setting throughout this paper, in order to evaluate the security against the most powerful adversaries.

2.2 Practical Relevance of the Multi-key Setting

The terms "broadcast" and "multi-user" imply a setting where one message is sent to many users, encrypted under independent keys. Note, however, that this setting does not actually require a large amount of users, and also applies to one user that rekeys frequently.

Frequent rekeying is often a result of the common implementation practice to use *session keys*. As explained in [48, Sect. 12.2.2], session keys limit the available ciphertext under the same key for cryptanalytic attacks, and limit the exposure in case a session key is compromised.

Furthermore, rekeying is necessary in certain scenarios in order to avoid cryptanalytical attacks or to comply with existing standards. It should be noted that several cryptanalytical attacks have a higher success probability when more plaintext-ciphertext pairs under a specific key are available [8].

For example, NIST limits the amount of plaintext that can be processed under the same key to 2^{32} blocks (32 GB) for three-key Triple-DES, and to 2^{20} blocks (16 MB) for two-key Triple-DES [6]. In the case of MAC functions, NIST not only recommends to limit the number of message blocks under the same key, but also to limit the number of MAC failures before rekeying is required [27,28]. In the case of TLS, rekeying is required after only one MAC failure [34].

AlFardan et al. [3] showed that it is a realistic attack vector in the case of TLS to obtain the encryption of one secret (a cookie or password) under multiple independent keys. They explained that this can be done either by using JavaScript malware to generate multiple sessions, or by causing the session to be terminated, after which some applications automatically reconnect and retransmit the cookie or password. As shown by Paterson et al. [54], the same attack setting also applies to WPA-TKIP because of its use of per-packet keys.

2.3 Security in the Multi-key Setting

As noted by Biham [12,13], there exists a faster generic key-recovery attack on any block cipher in the multi-key setting compared to the single-key setting. This can be seen as follows. To keep our explanation simple, let us assume in this section that key size k equals the block size n.

In the single-key setting, an adversary with $D = 1$ plaintext-ciphertext pair will need on average $T = 2^{k-1}$ encryptions to recover the key by exhaustive search with a success probability of about 50 %. More plaintext-ciphertext pairs will increase the success probability of the attack, as probability decreases that

a random key will be found instead of the correct key, but will not reduce the time complexity of the attack.

Recovering one key can be done with a lower time complexity in the multi-key setting. To see this, let an attacker have D encryptions

$$E_{K_1}(P), E_{K_2}(P), \ldots, E_{K_D}(P) \tag{1}$$

of the same plaintext P under multiple independent keys $K_1, K_2, \ldots, K_D$. Then, after on average $T = 2^{k-1}/D$ encryptions of the plaintext P, one of the keys $K_1, K_2, \ldots, K_D$ will be recovered with a success probability of about 50 %.

Besides this observation, Biham also remarked in [12,13] that key collisions become likely in this multi-key setting after about $D = 2^{k/2}$ plaintext-ciphertext pairs. Consequently, the key size k should be chosen to be sufficiently large by design to avoid key collision attacks.

In Sect. 3, we will prove that the Even-Mansour construction has similar security to an ideal block cipher in the multi-key setting, assuming the number of plaintexts queried per key is small. Or put differently, when multi-key attacks with a small amount of plaintext per key are taken into account, there is little advantage in choosing a block cipher with a more complicated key schedule than the Even-Mansour construction.

2.4 Related Work

The time-memory tradeoff of Hellman [37] is not a concern for block ciphers with a reasonably long key size, because its precomputation time is the same as that of exhaustive key search. This is different from the time-memory-data tradeoffs for stream ciphers of Babbage-Golić [5,33] and Biryukov-Shamir [17], where the time complexity can be far below exhaustive search.

As shown by Hong and Sarkar [38], and independently by Birykov [14], the stream cipher time-memory-data tradeoffs can be applied to the block cipher setting as well, assuming that a plaintext is encrypted under multiple keys. Their work generalizes the findings of Biham that we presented in Sect. 2.3.

Chatterjee, Koblitz, Menezes and Sarkar critiqued the security proofs of symmetric-key encryption and authentication modes [19,43,47], pointing out that security is often reduced when a multi-user setting is considered.

Their findings inspired Fouque et al. [31] to look at collision search algorithms in the multi-user setting. One of their results is the first analysis of the Even-Mansour construction in the multi-user setting. We will use an entirely different approach in this paper, by considering information-theoretic adversaries that are only limited by the number of queries to the Even-Mansour block cipher and to the underlying permutation. As we will show, the attack by Fouque et al. reaches the security bound that we will prove for the Even-Mansour construction in the multi-key setting.

A recent paper by Andreeva et al. [4] considers the security of keyed sponge constructions in the single-target and multi-target scenarios, which are similar to our single-key and multi-key settings. The approach that we follow in this

paper is different, as we introduce these concepts in a more general way. Furthermore, we do not express the attacks and security bounds in terms of the "total maximum multiplicity μ", a parameter that is specific to the analysis of sponge constructions.

Note that the multi-user setting is not only relevant for symmetric-key cryptography, but also for public-key cryptography. For a theoretical treatment of public-key encryption in the multi-user setting, we refer to Bellare et al. [7].

3 Security Proofs in the Multi-key Setting

Block cipher security in the multi-key setting is formalized with a distinguisher comparing two worlds, one in which the distinguisher is given access to a block cipher instantiated with ℓ keys, and one in which it is given access to ℓ independent permutations. Our focus is on constructions in the ideal model, meaning they make use of an ideal primitive.

Definition 1. *An ideal primitive is a uniformly distributed random variable over a set of functions F.*

These primitives model basic components from which cryptographic algorithms are constructed. In line with Kerckhoffs's principle, ideal primitives are public and can be accessed by adversaries in security definitions. The adversaries themselves are information-theoretic and are only bounded in the number of queries they make to each oracle.

Let $\mathsf{perm}(n)$ denote the set of all permutations on n bits, and $\mathsf{block}(k, n)$ denote the set of all block ciphers with k-bit key and n-bit block size. Let ℓ denote number of keys K_i under which the adversary performs queries, that is, there is at least one query for every key K_i for $1 \leq i \leq \ell$.

Definition 2 (Multi-key Security). *Let Π be a primitive and E^{Π} a random variable over $\mathsf{block}(k, n)$. Given an adversary $\mathcal{A}$, its multi-key advantage with respect to ℓ keys is*

$$\mathbf{Adv}_{E}^{\mathsf{mk}}(\mathcal{A}) = \left| \Pr\left(\mathcal{A}^{E^{\Pi}_{K_1}, E^{\Pi}_{K_2}, \ldots, E^{\Pi}_{K_\ell}, \Pi} \to 1\right) - \Pr\left(\mathcal{A}^{p_1, p_2, \ldots, p_\ell, \Pi} \to 1\right) \right|, \quad (2)$$

where the keys $K_1, \ldots, K_\ell$ are independently and uniformly drawn from $\{0,1\}^k$, and $p_1, \ldots, p_\ell$ are independently and uniformly drawn from $\mathsf{perm}(n)$. The adversary $\mathcal{A}$ has access to both forward and inverse oracles.

In the case of the Even-Mansour block cipher, the primitive Π is a permutation, whereas for an ideal block cipher, the primitive Π is the block cipher itself.

Note that our definition is similar to the "3PRP" notion in the security analysis of Chaskey [49], however we consider ℓ independent keys instead of three keys that are related to each other. Our definition also closely follows the "Joint Distinguishing Advantage" of Andreeva et al. [4, Definition 2], except that the total maximum multiplicity μ is not a parameter in our security definition.

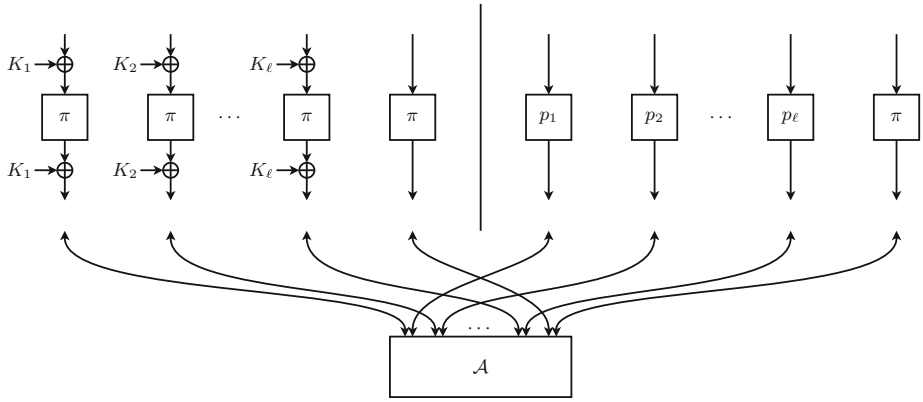

Fig. 2. An Even-Mansour block cipher $E_K(P) = \pi(P \oplus K) \oplus K$ in the multi-key setting. Although only one direction is shown, inverse oracles can be accessed as well. The number of queries by the adversary $\mathcal{A}$ to any of the first ℓ oracles is denoted by D, the number of queries to the last oracle by T.

Theorem 1 (Even-Mansour Multi-key Security). *Let* EM *be the Even-Mansour block cipher $E_K(P) = \pi(P \oplus K) \oplus K$, then for all $\mathcal{A}$ making at most D queries to $E_{K_1}, \ldots, E_{K_\ell}$ (resp. $p_1, \ldots, p_\ell$) or their inverses and at most T queries to π or π^{-1},*

$$\mathbf{Adv}^{\mathsf{mk}}_{\mathrm{EM}}(\mathcal{A}) \leq \frac{D^2 + 2DT}{2^n}. \tag{3}$$

Our proof is similar to the security proof of the MAC function Chaskey [49], except that we now consider that ℓ keys are drawn independently and uniformly at random. The proof uses Patarin's H-coefficient technique [53]. For a detailed explanation of this technique, we refer to Chen and Steinberger [21]. The proof can be seen as a generalization of the security analysis of the Even-Mansour block cipher [29,30].

Proof. As shown in Fig. 2, we consider an adversary $\mathcal{A}$ that has bidirectional access to $\ell + 1$ oracles $(\mathcal{O}_1, \ldots, \mathcal{O}_{\ell+1})$. In the real world, these are $(E_{K_1}, \ldots, E_{K_\ell}, \pi)$ (where $E_K(P) = \pi(P \oplus K) \oplus K$) with $K_i \xleftarrow{\$} \{0,1\}^n$ for $i = 1, \ldots, \ell$, $\pi \xleftarrow{\$} \mathsf{perm}(n)$, and in the ideal world these are $(p_1, \ldots, p_\ell, \pi) \xleftarrow{\$} \mathsf{perm}(n)^{\ell+1}$. Without loss of generality we assume that $\mathcal{A}$ is deterministic. It makes D_i queries to oracle $\mathcal{O}_i$ for $i = 1, \ldots, \ell$, and T queries to $\mathcal{O}_{\ell+1}$. Let $D = \sum_{i=1}^{\ell} D_i$. To be overly generous to the adversary $\mathcal{A}$, after it has made all of its $D + T$ queries, but before it outputs its decision, we will reveal the keys $K_1, \ldots, K_\ell$ (in the real world) or randomly generated dummy keys $K_1, \ldots, K_\ell$ (in the ideal world).

The interaction of $\mathcal{A}$ with the oracles can be summarized by a transcript $\tau = (K_1, \ldots, K_\ell, \tau_1, \ldots, \tau_{\ell+1})$. Here, the directionless list of queries to $\mathcal{O}_j$ for $i = 1, \ldots, \ell$ is denoted by $\tau_i = \{(P_i^{(1)}, C_i^{(1)}), \ldots, (P_i^{(D_i)}, C_i^{(D_i)})\}$, and to $\mathcal{O}_{\ell+1}$

by $\tau_{\ell+1} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(T)}, y^{(T)})\}$. We assume the adversary never makes duplicate queries, so that $P_i^{(j)} \neq P_i^{(j')}$, $C_i^{(j)} \neq C_i^{(j')}$, $x^{(j)} \neq x^{(j')}$, and $y^{(j)} \neq y^{(j')}$ for all i, j, j' where $j \neq j'$.

Given the fixed deterministic adversary $\mathcal{A}$, we denote the probability distribution of transcripts in the real world by X, and in the ideal world by Y. We say that a transcript τ is attainable if it can be obtained from interacting with $(p_1, \ldots, p_\ell, \pi)$, hence if $\Pr(Y = \tau) > 0$. According to the H-coefficient technique, we have (see [21] for a proof):

Lemma 1 (H-coefficient Technique). *Let us consider a fixed deterministic adversary $\mathcal{A}$, and let $\mathcal{T} = \mathcal{T}_{\text{good}} \cup \mathcal{T}_{\text{bad}}$ be a partition of the set of attainable transcripts. Let ε be such that for all $\tau \in \mathcal{T}_{\text{good}}$*

$$\frac{\Pr(X = \tau)}{\Pr(Y = \tau)} \geq 1 - \varepsilon. \tag{4}$$

Then, $\mathbf{Adv}^{\mathsf{mk}}_{\text{EM}}(\mathcal{A}) \leq \varepsilon + \Pr(Y \in \mathcal{T}_{\text{bad}})$.

We say that a transcript τ is *bad* if two different queries would result in the same input or output to π, were $\mathcal{A}$ interacting with the real world. Put formally, τ is bad if one of the following conditions is set:

$$\exists i, i', j, j' : i \neq i' : \; P_i^{(j)} \oplus P_{i'}^{(j')} = K_i \oplus K_{i'} \;\vee\; C_i^{(j)} \oplus C_{i'}^{(j')} = K_i \oplus K_{i'}, \tag{5}$$

$$\exists i, j, j' : \qquad P_i^{(j)} \oplus x^{(j')} = K_i \qquad \vee\; C_i^{(j)} \oplus x^{(j')} = K_i. \tag{6}$$

A transcript that is not a *bad* transcript, is referred to as a *good* transcript.

Upper Bounding $\Pr(Y \in \mathcal{T}_{\text{bad}})$. We want to upper bound the event that a transcript τ in the ideal world satisfies (5)–(6). Note that $K_i \xleftarrow{\$} \{0,1\}^n$ for $i = 1, \ldots, \ell$ are dummy keys generated independently of $\tau_1, \ldots, \tau_\ell$. Therefore, there are at most $2D_iD_{i'}$ possible keys that satisfy (5) for any fixed $i \neq i'$. Analogously, there are at most $2D_iT$ possible keys that satisfy (6) for any fixed i. Therefore,

$$\Pr(Y \in \mathcal{T}_{\text{bad}}) \leq \frac{\sum_i \sum_{i'<i} 2D_iD_{i'} + \sum_i 2D_iT}{2^n}, \tag{7}$$

$$\leq \frac{D^2 + 2DT}{2^n}. \tag{8}$$

Lower Bounding Ratio $\Pr(X = \tau) \,/\, \Pr(Y = \tau)$. Let us consider a good and attainable transcript $\tau \in \mathcal{T}_{\text{good}}$. Then denote by $\Omega_X = 2^{n\ell} \cdot 2^n!$ the set of all possible oracles in the real world and by $\mathsf{comp}_X(\tau) \subseteq \Omega_X$ the set of oracles in Ω_X compatible with transcript τ. Define $\Omega_Y = 2^{n\ell} \cdot (2^n!)^{\ell+1}$ and $\mathsf{comp}_Y(\tau)$ similarly. According to the H-coefficient technique:

$$\Pr(X = \tau) = \frac{|\mathsf{comp}_X(\tau)|}{|\Omega_X|}, \qquad \text{and} \qquad \Pr(Y = \tau) = \frac{|\mathsf{comp}_Y(\tau)|}{|\Omega_Y|}. \tag{9}$$

First, we calculate $|\mathsf{comp}_X(\tau)|$. As $\tau \in \mathcal{T}_{\text{good}}$, there are no two queries in τ with the same input to or output of the underlying permutation. Any query tuple in τ therefore fixes exactly one input-output pair of the underlying oracle. Because τ consists of $D+T$ query tuples, the number of possible oracles in the real world equals $(2^n - D - T)!$. By a similar reasoning, the number of possible oracles in the ideal world equals $\prod_{i=1}^{\ell}(2^n - D_i)! \cdot (2^n - T)!$. Therefore,

$$\Pr(X = \tau) = \frac{(2^n - D - T)!}{2^{n\ell} \cdot 2^n!}, \tag{10}$$

$$\Pr(Y = \tau) = \frac{\prod_{i=1}^{\ell}(2^n - D_i)! \cdot (2^n - T)!}{2^{n\ell} \cdot (2^n!)^{\ell+1}} \leq \frac{(2^n - D - T)! \cdot (2^n!)^{\ell}}{2^{n\ell} \cdot (2^n!)^{\ell+1}}. \tag{11}$$

It then follows that $\Pr(X = \tau) / \Pr(Y = \tau) \geq 1$. □

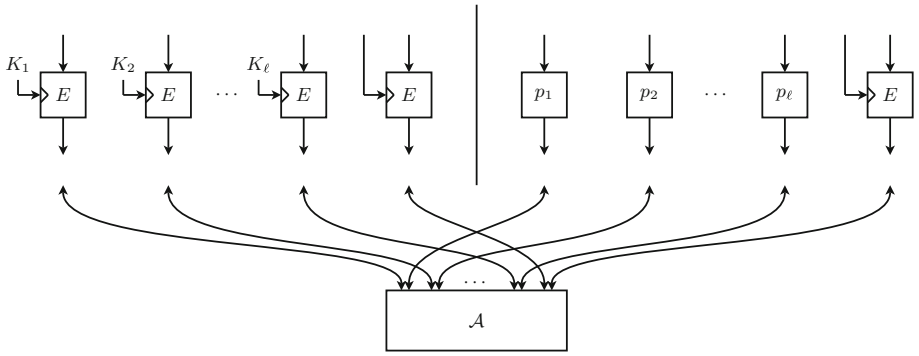

Fig. 3. An ideal block cipher E_K in the multi-key setting. Although only one direction is shown, all oracles are assumed to be bidirectional.

Theorem 2 (Ideal Block Cipher Multi-key Security). *Let the ideal block cipher* IBC *be uniformly distributed random variable over* $\mathsf{block}(k, n)$*, then for all* $\mathcal{A}$ *making at most* D *queries to* $E_{K_1}, \ldots, E_{K_\ell}$ *(resp.* $p_1, \ldots, p_\ell$*) or their inverses and at most* T *queries to* E_K *or its inverse under adversary-chosen keys,*

$$\mathbf{Adv}^{\mathsf{mk}}_{\text{IBC}}(\mathcal{A}) \leq \frac{\ell^2 + 2\ell T}{2^{k+1}}. \tag{12}$$

Proof. We consider the adversary $\mathcal{A}$ shown in Fig. 3. Define $\mathbf{E}$ to be the event where either

1. there exists $i \neq j$ such that $K_i = K_j$ or
2. there exists a query $E(K, X)$ or $E^{-1}(K, X)$ such that $K = K_i$ for some i.

Given that $\mathbf{E}$ does not happen, $E_{K_1}, \ldots, E_{K_\ell}$ are drawn independently and uniformly at random from $\mathsf{perm}(n)$, and all queries made to E are independent of

$E_{K_1}, \ldots, E_{K_\ell}$. Therefore $(E_{K_1}, \ldots, E_{K_\ell}, E)$ and $(p_1, \ldots, p_\ell, E)$ are indistinguishable given the negation of **E**, and by the fundamental lemma of game playing,

$$\mathbf{Adv}^{\mathsf{mk}}_{\mathrm{IBC}}(\mathcal{A}) \leq \Pr(\mathbf{E}). \tag{13}$$

The probability that there exists an adversary query $E(K, X)$ or $E^{-1}(K, X)$ such that $K = K_i$ is at most $\frac{T\ell}{2^k}$. The probability that two keys collide is bounded above by $\frac{\ell^2}{2^{k+1}}$, hence

$$\Pr(\mathbf{E}) \leq \frac{\ell^2 + 2\ell T}{2^{k+1}}. \tag{14}$$

□

By our definition, there must be at least one query for every key. Therefore $D \geq \ell$, so that the following corollary can be derived from Theorem 2:

Corollary 1 (Corollary of Theorem 2).

$$\mathbf{Adv}^{\mathsf{mk}}_{\mathrm{IBC}}(\mathcal{A}) \leq \frac{D^2 + 2DT}{2^{k+1}}. \tag{15}$$

Observe that when the amount of plaintext per key is small, this bound is close to that of Theorem 2.

3.1 Tightness of the Security Bounds

Several attacks have been published that match the security bound of the Even-Mansour block cipher in the single-key setting. The first attacks were published by Daemen [22]: a known-plaintext attack for $D = 2$ and a chosen-plaintext attack for any value of D. Biruykov and Wagner [18] presented a known-plaintext attack for $D \geq 2^{n/2}$. A known-plaintext attack for any value of D was given by Dunkelman et al. [26].

The single-key setting is a special case of the multi-key setting where $\ell = 1$. We proved in Sect. 3 (see Theorem 1) that the security bound of Even-Mansour in multi-key setting is a straightforward extension of the single-key setting. Therefore, the bound that we derived for Even-Mansour in the multi-key setting is also tight.

An attack matching our Even-Mansour security bound in the multi-key setting was recently given by Fouque et al. [31] for $\ell = 2^{n/3}$, $D_i = 2^{n/3}$ for $i = 1, \ldots, \ell$ and $T = 2^{n/3}$.

In the case of an ideal block cipher in the multi-key setting with $D = \ell$, the key collision and time-memory trade-off attacks of Biham [12,13] show that the security bound of Corollary 1 is also tight. For a discussion of these attacks and their subsequent improvements, we refer to Sect. 2.3. Evidently, these attacks are also applicable to the Even-Mansour block cipher in the multi-key setting.

4 Discussion

As we proved in Sect. 3, the Even-Mansour block cipher has similar security in the multi-key setting as an ideal block cipher with the same block and key size, assuming $D \approx \ell$. In Sect. 2.2, we pointed out the relevance of this multi-key setting in practice.

The Even-Mansour block cipher is interesting from a design point of view because of its simplicity. As Dunkelman et al. [26] argued, it also achieves minimalism, in the sense that removing any component (one of the two key additions or the permutation) results in an insecure construction. But the Even-Mansour construction also has many implementation advantages.

From an efficiency point of view, the Even-Mansour block cipher avoids that round keys need to be precalculated and stored in memory, or that they need to be calculated on-the-fly. Avoiding precalculation of the key schedule results in a higher key agility, because of the lower cost to rekey. If round keys do not need to be calculated on-the-fly, the efficiency of every block cipher call increases. For software implementations, avoiding a key schedule reduces register pressure and decreases RAM requirements. The amount of RAM is very critical on certain microcontrollers, as shown for example in [39].

From a security point of view, the Even-Mansour block cipher avoids the need to store round keys securely. Note that in the case of AES-128, recovery of any round key leads to recovery of the encryption key. In the case of AES with 192-bit or 256-bit keys, any two consecutive round keys can be used to recover the encryption key.

Secure key storage is not only a problem for smart cards and RFID tags, but is also difficult to ensure on general purpose CPUs. The virtual memory system may move cryptographic keys into swap storage, which necessitates error-prone techniques to avoid swapping, or to use swap encryption [57].

But even in the presence of these countermeasures, cold boot attacks [35] may be used to exploit the fact that DRAM retains a large part of its memory for several seconds after removing power. Cooling techniques may be used to increase this time to several hours or even days.

If the round keys that are recovered by a cold boot attack contain errors (due to memory bit decay), it may still be possible to recover the encryption key. For AES, Halderman et al. [35] describe a simple algorithm to recover the encryption key in this case. Improved attacks were later given by Albrecht et al. [2] using integer programming, and by Tsow [58], and Kamal and Youssef [41] using a SAT solver.

The Even-Mansour block cipher avoids all attacks that recover the encryption key from multiple noisy round keys, as it avoids the calculation of round keys altogether. The leakage of encryption keys or round keys cannot be avoided without additional security measures, for example by ensuring they are only stored in the processor registers and not in RAM [50,51]. However, this problem becomes much more manageable for the Even-Mansour construction, as only one n-bit key needs to be protected, instead of multiple round keys.

5 Conclusion and Future Work

Rekeying occurs frequently in real-world implementations, meaning that a plaintext may be encrypted under different keys. This setting is used, for example, in the attacks by AlFardan et al. [3] on TLS, and by Paterson et al. [54] on WPA-TKIP.

This setting is often referred to as the *broadcast setting* or the *multi-user setting*. In this paper, we introduced the *multi-key setting* to generalize the aforementioned settings. In the multi-key setting, the adversary can perform chosen-plaintext and chosen-ciphertext attacks under a set of unknown keys.

In the multi-key setting, we proved that the Even-Mansour block cipher is secure up to $(D^2 + 2DT)/2^n$ queries. We proved a similar bound for an ideal block cipher with $k = n$, and showed that both bounds are tight. We used our proofs to argue in favor of the Even-Mansour construction: not only because of the simplicity of its design, but also because the lack of a key schedule makes it easier to generate fast and secure implementations.

Modes of operation for encryption and/or authentication may be designed more efficiently, if it is known that the underlying block cipher follows the Even-Mansour construction. It also seems that an Even-Mansour block cipher may still be secure if the underlying permutation is far from ideal, an idea that was pioneered by the design of the Chaskey MAC function [49]. We leave the further exploration of these research questions to future work.

Acknowledgments. The authors would like to thank the anonymous reviewers and Bart Mennink for their useful comments and suggestions. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007), by Research Fund KU Leuven, OT/13/071, and by the French Agence Nationale de la Recherche through the BLOC project under Contract ANR-11-INS-011. Nicky Mouha is supported by a Postdoctoral Fellowship from the Flemish Research Foundation (FWO-Vlaanderen). Atul Luykx is supported by a Ph.D. Fellowship from the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. Abdalla, M., Bellare, M.: Increasing the lifetime of a key: a comparative analysis of the security of re-keying techniques. In: Okamoto [52], pp. 546–559
2. Albrecht, M., Cid, C.: Cold boot key recovery by solving polynomial systems with noise. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 57–72. Springer, Heidelberg (2011)
3. AlFardan, N.J., Bernstein, D.J., Paterson, K.G., Poettering, B., Schuldt, J.C.: On the security of RC4 in TLS and WPA. In: USENIX Security Symposium (2013)
4. Andreeva, E., Daemen, J., Mennink, B., Assche, G.V.: Security of keyed sponge constructions using a modular proof approach. In: Demirci, H., Leander, G. (eds.) FSE 2015. LNCS, Springer (2015, to appear). https://www.cosic.esat.kuleuven.be/publications/article-2502.pdf

5. Babbage, S.H.: Improved "exhaustive search" attacks on stream ciphers. In: ECOS 95 (European Convention on Security and Detection), Conference publication No. 408, pp. 161–166, May 1995
6. Barker, W.C., Barker, E.: SP 800–67 Revision 1: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, January 2012. http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf
7. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: security proofs and improvements. In: Preneel [56], pp. 259–274
8. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: 38th Annual Symposium on Foundations of Computer Science, FOCS 1997, Miami Beach, Florida, USA, October 19–22, 1997, pp. 394–403. IEEE Computer Society (1997)
9. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003)
10. Biham, E.: New types of cryptanalytic attacks using related keys (extended abstract). In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 398–409. Springer, Heidelberg (1994)
11. Biham, E.: New types of cryptanalytic attacks using related keys. J. Cryptology **7**(4), 229–246 (1994)
12. Biham, E.: How to Forge DES-Encrypted Messages in 2^{28} Steps. Technical report CS0884, Technion Computer Science Department, Israel (1996)
13. Biham, E.: How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. Inf. Process. Lett. **84**(3), 117–124 (2002)
14. Biryukov, A.: Some Thoughts on Time-Memory-Data Tradeoffs. Cryptology ePrint Archive, Report 2005/207 (2005). http://eprint.iacr.org/
15. Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
16. Biryukov, A., Khovratovich, D., Nikolić, I.: Distinguisher and related-key attack on the full AES-256. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 231–249. Springer, Heidelberg (2009)
17. Biryukov, A., Shamir, A.: Cryptanalytic time/memory/data tradeoffs for stream ciphers. In: Okamoto [52], pp. 1–13
18. Biryukov, A., Wagner, D.: Advanced slide attacks. In: Preneel [56], pp. 589–606
19. Chatterjee, S., Menezes, A., Sarkar, P.: Another look at tightness. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 293–319. Springer, Heidelberg (2012)
20. Chen, L.: Recommendation for Key Derivation Using Pseudorandom Functions (Revised). NIST special publication 800–108, National Institute of Standards and Technology (NIST), October 2009. http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf
21. Chen, S., Steinberger, J.: Tight security bounds for key-alternating ciphers. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 327–350. Springer, Heidelberg (2014)
22. Daemen, J.: Limitations of the even-mansour construction. In: Imai et al. [40], pp. 495–498
23. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
24. Daemen, J., Rijmen, V.: On the related-key attacks against AES. In: Proceedings of the Romanian Academy, Series A, vol. 13(4), pp. 395–400 (2012)

25. Davies, D.W.: Some regular properties of the 'data encryption standard' algorithm. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO, pp. 89–96. Plenum Press, New York (1982)
26. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in cryptography: the even-mansour scheme revisited. In: Pointcheval and Johansson [55], pp. 336–354
27. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST special publication 800–38b, National Institute of Standards and Technology (NIST), May 2005. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf
28. Dworkin, M.: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf
29. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. In: Imai et al. [40], pp. 210–224
30. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptology **10**(3), 151–162 (1997)
31. Fouque, P.-A., Joux, A., Mavromati, C.: Multi-user collisions: applications to discrete logarithm, even-mansour and PRINCE. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 420–438. Springer, Heidelberg (2014)
32. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl - a SHA-3 candidate. Submission to the NIST SHA-3 Competition (Round 3) (2011). http://www.groestl.info/Groestl.pdf
33. Golić, J.D.: Cryptanalysis of alleged A5 stream cipher. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 239–255. Springer, Heidelberg (1997)
34. Group, I.N.W.: The Transport Layer Security (TLS) Protocol (2006). http://tools.ietf.org/html/rfc4346
35. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA, pp. 45–60. USENIX Association (2008)
36. Hellman, M.E., Merkle, R., Schroeppel, R., Diffie, W., Pohlig, S., Schweitzer, P.: Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard. Technical report, Stanford University, USA (1976)
37. Hellman, M.E.: A cryptanalytic time-memory trade-off. IEEE Trans. Inf. Theory **26**(4), 401–406 (1980)
38. Hong, J., Sarkar, P.: New applications of time memory data tradeoffs. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 353–372. Springer, Heidelberg (2005)
39. Ideguchi, K., Owada, T., Yoshida, H.: A Study on RAM Requirements of Various SHA-3 Candidates on Low-cost 8-bit CPUs. Cryptology ePrint Archive, Report 2009/260 (2009). http://eprint.iacr.org/
40. Matsumoto, T., Imai, H., Rivest, R.L. (eds.): ASIACRYPT 1991. LNCS, vol. 739. Springer, Heidelberg (1993)
41. Kamal, A.A., Youssef, A.M.: Applications of SAT solvers to AES key recovery from decayed key schedule images. In: 2010 Fourth International Conference on Emerging Security Information Systems and Technologies (SECURWARE), pp. 216–220. IEEE (2010)
42. Knudsen, L.R., Rijmen, V.: Known-key distinguishers for some block ciphers. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 315–324. Springer, Heidelberg (2007)

43. Koblitz, N., Menezes, A.: Another look at HMAC. J. Math. Cryptology **7**(3), 225–251 (2013)
44. Lai, X., Massey, J.L.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
45. Mantin, I., Shamir, A.: A practical attack on broadcast RC4. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 152–164. Springer, Heidelberg (2002)
46. Mehuron, W.: Data Encryption Standard (DES), October 1999. http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
47. Menezes, A.: Another look at provable security. In: Pointcheval and Johansson [55], p. 8
48. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, USA (1997)
49. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In: Joux, A., Youssef, A. (eds.) SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer, Heidelberg (2014)
50. Müller, T., Dewald, A., Freiling, F.C.: AESSE: a cold-boot resistant implementation of AES. In: Costa, M., Kirda, E. (eds.) Proceedings of the Third European Workshop on System Security, EUROSEC 2010, Paris, France, April 13, 2010, pp. 42–47. ACM (2010)
51. Müller, T., Freiling, F.C., Dewald, A.: TRESOR runs encryption securely outside RAM. In: Proceedings of 20th USENIX Security Symposium, San Francisco, CA, USA, August 8–12, 2011. USENIX Association (2011)
52. Okamoto, T. (ed.): ASIACRYPT 2000. LNCS, vol. 1976. Springer, Heidelberg (2000)
53. Patarin, J.: The "coefficients H" technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer, Heidelberg (2009)
54. Paterson, K.G., Poettering, B., Schuldt, J.C.N.: Plaintext recovery attacks against WPA/TKIP. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 325–349. Springer, Heidelberg (2015)
55. Pointcheval, D., Johansson, T. (eds.): EUROCRYPT 2012. LNCS, vol. 7237. Springer, Heidelberg (2012)
56. Preneel, B. (ed.): EUROCRYPT 2000. LNCS, vol. 1807. Springer, Heidelberg (2000)
57. Provos, N.: Encrypting virtual memory. In: Bellovin, S.M., Rose, G. (eds.) 9th USENIX Security Symposium, Denver, Colorado, USA, August 14–17, 2000. USENIX Association (2000)
58. Tsow, A.: An improved recovery algorithm for decayed AES key schedule images. In: Jacobson Jr, M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 215–230. Springer, Heidelberg (2009)