J. Cryptol. (2015) 28: 1–28
DOI: 10.1007/s00145-013-9164-7

Journal of
CRYPTOLOGY

Slidex Attacks on the Even–Mansour Encryption Scheme*

Orr Dunkelman
Computer Science Department, University of Haifa, Haifa 31905, Israel
orrd@cs.haifa.ac.il
and
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, P.O. Box 26, Rehovot 76100, Israel

Nathan Keller
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, P.O. Box 26, Rehovot 76100, Israel
and
Department of Mathematics, Bar-Ilan University, Ramat Gan 52900, Israel
nathan.keller@weizmann.ac.il

Adi Shamir
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, P.O. Box 26, Rehovot 76100, Israel
adi.shamir@weizmann.ac.il

Communicated by Serge Vaudenay

Received 24 March 2013
Online publication 20 November 2013

Abstract. The Even–Mansour cryptosystem was developed in 1991 in an attempt to obtain the simplest possible block cipher, using only one publicly known random permutation and two whitening keys. Its exact security remained open for more than 20 years in the sense that the lower bound proof considered known plaintexts, whereas the best published attack (which is based on differential cryptanalysis) required chosen plaintexts. In this paper, we solve this open problem by introducing the new *extended slide attack* (abbreviated as slidex) which matches the $T = \Omega(2^n/D)$ lower bound on the time T for any number of *known plaintexts* D. By using this tight security result, we show that a simplified single-key variant of the Even–Mansour scheme has exactly the same security as the original two-key scheme. We then show how to apply variants of the slidex attack to several other cryptosystems, including an Even–Mansour variant which adds rather than XORs its whitening keys, DES protected with decorrelation modules, various flavors of DESX, and a reduced-round version of GOST. In addition, we show how to apply the slidex attack in extreme scenarios in which the cryptanalyst is only given some partial information about the plaintexts, or when he can only use a tiny amount of memory.

[*] This paper is an extended version of [12], presented at EUROCRYPT 2012.

Key words. Even–Mansour block cipher, Single-key Even–Mansour, Whitening keys, Provable security, Tight security bounds, Slide attack, Slidex attack.

1. Introduction

The Even–Mansour (EM) block cipher was proposed at ASIACRYPT 1991 [13,14], and was strongly influenced by the design of the DESX scheme by Ron Rivest in 1984 [27]. It uses a single publicly known random permutation P on n-bit values and two secret n-bit keys K_1 and K_2, and defines the encryption of the n-bit plaintext m as $E(m) = P(m \oplus K_1) \oplus K_2$. The decryption of the n-bit ciphertext c is similarly defined as $D(c) = P^{-1}(c \oplus K_2) \oplus K_1$. Its extreme simplicity and suitability for rigorous security analysis had made it a very popular research topic in the last few years, with many papers related to this scheme appearing at CRYPTO, EUROCRYPT, ASIACRYPT, CHES, FSE and the IACR archive [1,7,8,11,16,20,21,23,24,29].

Unfortunately, all the bounds published so far about the security of the EM scheme are not tight in the sense that the lower bound allows known message attacks whereas the best known upper bounds require either chosen plaintexts or an extremely large number of known plaintexts. Our goal in this paper is to obtain the first tight bound, which will not only characterize the exact security of the original EM scheme, but will also make it possible to rigorously prove that a simplified variant of the original EM scheme offers exactly the same security as the original EM scheme.

One of the main tools used in previous attacks was the slide attack [5]. Originally, slide attacks were developed in order to break iterated cryptosystems with an arbitrarily large number of rounds by exploiting their self-similarity under small shifts. The attack searched the given data for a slid pair of encryptions which have identical values along their common part (see Sect. 3.2 for formal definitions). For each candidate pair, the attack uses the two known plaintexts and two known ciphertexts to analyze the two short non-common parts in order to verify the assumption that the two encryptions are indeed a slid pair, and if so to derive some key material. A different variant of this attack, called *slide with a twist* [6], tries to find a slid pair consisting of one encryption and one decryption, which have identical values along their common parts (i.e., the attack considers both shifts and reversals of the encryption rounds). In both cases, the existence of slid pairs is a random event which is expected to have a sharp threshold: Regardless of whether we use known or chosen messages, we do not expect to find any slid pairs if we are given fewer than $2^{n/2}$ encryptions where n is the size of the internal state.[1] Consequently, we cannot apply the regular or twisted slide attack unless we are given a sufficiently large number of encryptions, even if we are willing to trade off the lower amount of data with higher time and space complexities.

In this paper, we propose the *slidex attack*, which is a new extended version of the slide attack that can efficiently use any amount of given data, even when it is well below the $2^{n/2}$ threshold for the existence of slid pairs. Its main novelty is that we no longer require equality between the values along the common part, but only the existence of

[1] We note that for specific block cipher structures, e.g., Feistel networks, a specialized slide attack can require fewer than $2^{n/2}$ plaintexts. However, there is no such method that works for general structures.

some known relationship between these values. By using this new attack, we can finally close the gap between the upper and lower bounds on the security of the EM scheme.

To demonstrate the usefulness and versatility of the new slidex attack, we apply it to several additional schemes which are unrelated to the EM scheme. In particular, we show how to break 20 rounds of GOST using 2^{33} known plaintexts in 2^{77} time, and how to use the complementation property of DES in order to attack it with a slide-type attack even when it is surrounded on both sides by one of Vaudenay's proposed decorrelation modules.

The paper is organized as follows. In Sect. 2, we introduce the Even–Mansour scheme, describe its formal proof of security, and survey all the previously published attacks on the scheme. In Sect. 3, we describe the known types of slide attacks, and explain why they cannot efficiently exploit a small number of known plaintexts. We then introduce our new Slidex attack, and use it to develop a new upper bound on the security of the Even–Mansour scheme which matches the proven lower bound for any number of known plaintexts. In Sect. 4, we describe the single-key variant of the Even–Mansour scheme, which is strictly simpler but has the same level of provable security. In Sect. 5, we analyze the security of several other variants of the Even–Mansour scheme, demonstrating both the generality and the fragility of its formal proof of security. Another limitation of the proof technique is described in Sect. 6, where we show that no comparable lower bound on the memory complexity of our attacks can exist. Sections 7 and 8 describe several generalizations of the slidex attack and their applications: In Sect. 7, we describe the *mirror slidex* attack and apply it to variants of GOST and DESX, and in Sect. 8 we describe the *addition slidex* attack and apply it to attack a variant of DES surrounded by decorrelation modules. We conclude the paper with open questions and directions for future research in Sect. 9.

2. The Even–Mansour Scheme

In this section, we present the Even–Mansour (EM) scheme, review its lower bound proof given in [13], and describe previous attacks on it presented in [9] and [6].

2.1. *Definition of the EM Scheme and Its Notation*

The Even–Mansour scheme is a block cipher which consists of a single publicly known permutation $\mathcal{F}$ over n-bit strings, preceded and followed by two independent n-bit whitening keys K_1 and K_2:

$$EM^{\mathcal{F}}_{K_1,K_2}(P) = \mathcal{F}(P \oplus K_1) \oplus K_2.$$

It is assumed that the adversary is allowed to perform two types of queries:

- Queries to a full encryption/decryption oracle, called an E-oracle, that computes either $E(P) = EM^{\mathcal{F}}_{K_1,K_2}(P)$ or $D(C) = (EM^{\mathcal{F}}_{K_1,K_2})^{-1}(C)$.
- Queries to an $\mathcal{F}$-oracle that computes either $\mathcal{F}(x)$ or $\mathcal{F}^{-1}(y)$.

The designers of EM considered two types of attacks. In the first type, called *existential forgery attack*, the adversary tries to find a *new* pair (P, C) such that $E(P) = C$. The second type is the more standard security game, where the adversary tries to decrypt a message C, i.e., to find P for which $E(P)=C$.[2] The data complexity of an attack on the scheme is determined by the number D of queries to the E-oracle and their type (i.e., known/chosen/adaptively chosen etc.), and the time complexity of the attack is lower bounded by the number T of queries to the $\mathcal{F}$-oracle.[3] The success probability of an attack is the probability that the single guess it produces (either a pair (P,C) for the first type of attack, or a plaintext P for the second type) is correct.

2.2. *The Lower Bound Security Proof*

The main rigorously proven result in [13] was an upper bound of $O(DT/2^n)$ on the success probability of any cryptanalytic attack (of either type) on EM that uses at most D queries to the E-oracle and T queries to the $\mathcal{F}$-oracle. This result implies that in order to attack EM with a constant probability of success, we must have $DT=\Omega(2^n)$. Since this security proof is crucial for some of our results, we briefly describe its main steps.

The proof requires several definitions. Consider a cryptanalytic attack on EM, and assume that at some stage of the attack, the adversary already performed s queries to the E-oracle and t queries to the $\mathcal{F}$-oracle, and obtained sets $\mathcal{D}$ and $\mathcal{T}$ of E-pairs and $\mathcal{F}$-pairs, respectively, i.e.,

$$\mathcal{D}=\big\{(P_i,C_i)\big\}_{i=1,\dots,s} \quad \text{and} \quad \mathcal{T}=\big\{(X_j,Y_j)\big\}_{j=1,\dots,t}.$$

We say that the key K_1 is *bad* with respect to the sets of queries $\mathcal{D}$ and $\mathcal{T}$, if there exist i,j such that $P_i \oplus K_1 = X_j$. Otherwise, K_1 is *good* with respect to $\mathcal{D},\mathcal{T}$. Intuitively, a good key is one whose feasibility cannot be deduced from the available data, whereas a bad key is one whose feasibility has to be further analyzed (but not necessarily discarded). Similarly, K_2 is bad w.r.t. $\mathcal{D},\mathcal{T}$ if there exist i,j such that $Y_j \oplus K_2 = C_i$, and K_2 is good otherwise. The key $K=(K_1,K_2)$ is *good* with respect to $\mathcal{D},\mathcal{T}$ if both K_1 and K_2 are good. It is easy to show that the number of good keys w.r.t. $\mathcal{D}$ and $\mathcal{T}$ is at least $2^{2n}-2st\cdot 2^n$. A pair $(K=(K_1,K_2),\mathcal{F})$ is *consistent* w.r.t. $\mathcal{D}$ and $\mathcal{T}$ if for any pair $(P_i,C_i)\in\mathcal{D}$ we have $C_i=K_2\oplus\mathcal{F}(P_i\oplus K_1)$, and for any pair $(X_j,Y_j)\in\mathcal{T}$, we have $\mathcal{F}(X_j)=Y_j$.

The proof consists of two main steps.

1. The first step shows that all good keys are, in some sense, equally likely to be the correct key. Formally, if the probability over the keys and over the permutations is uniform, then for all $\mathcal{D},\mathcal{T}$, the probability

$$\Pr_{K,\mathcal{F}}\big[K=k|(K,\mathcal{F}) \text{ is consistent with } \mathcal{D},\mathcal{T}\big]$$

[2] These security notions are significantly different than the indistinguishability notions of [18] which proved similar lower bounds on the inability of the adversary to distinguish the given instance of the cipher from a random permutation. Finding the actual keys not only allows distinguishing the construction from a random permutation, but also allows winning the two security games considered in [13].

[3] In concrete implementations, this oracle is usually replaced by some publicly known program which the attacker can run on its own. In this case, the type of query (e.g., whether the inputs are adaptively chosen or not) can determine whether the attack can be parallelized on multiple processors, but we ignore such low level details in our analysis.

is the same for any key $k \in \{0,1\}^{2n}$ that is good with respect to $\mathcal{D}, \mathcal{T}$.

We present the proof of this step, since it will be crucial in the sequel. It follows from Bayes' formula that it suffices to prove that the probability

$$p = \Pr_{K,\mathcal{F}}\bigl[(K,\mathcal{F}) \text{ is consistent with } \mathcal{D}, \mathcal{T} | K = k\bigr] \tag{1}$$

is the same for all good keys. Given a good key $k = (k_1, k_2)$, it is possible to transform the set $\mathcal{D}$ of E-pairs to an equivalent set $\mathcal{D}'$ of $\mathcal{F}$-pairs by transforming the E-pair (P_i, C_i) to the $\mathcal{F}$-pair $(P_i \oplus k_1, C_i \oplus k_2)$. Since the key k is good, the pairs in $\mathcal{D}'$ and $\mathcal{T}$ do not overlap, and hence p is simply the probability of consistency of a random permutation $\mathcal{F}$ with $s + t$ given distinct input/output pairs. This probability clearly does not depend on k, which proves the assertion.

2. The second step shows that the success probability of any attack is bounded by the sum of the probability that in some step of the attack, the right key becomes a bad key, and the probability that the adversary can successfully generate a "new" consistent E-pair (P, C) if the right key is still amongst the good keys. The first probability can be bounded by $4DT/(2^n - 2DT)$, and the second probability can be bounded by $1/(2^n - D - T)$. Hence, the total success probability of the attack is bounded by $O(DT/2^n)$. We omit the proof of this step since it is not used in the sequel.

We note that obtaining non-trivial information about the key (e.g., that the least significant bit of the K_1 is zero, or the value of $K_1 \oplus K_2$), is also covered by this proof. Hence, throughout the paper we treat such leakage of information as a "problem" in the security of the construction (even if the exact keys are not found).

2.3. *Previous Attacks on the Even–Mansour Scheme*

The first proposed attack on the Even–Mansour scheme was published by Joan Daemen at ASIACRYPT 1991 [9], as an illustration of the author's doubts on the usefulness of the Even–Mansour approach. Daemen used the framework of differential cryptanalysis [3] to develop a *chosen plaintext* attack which matched the Even–Mansour lower bound for any amount of given data. The approach is to pick D pairs of chosen plaintexts whose XOR difference is some nonzero constant Δ. This plaintext difference is preserved by the XOR with the prewhitening key K_1, and similarly, the ciphertext difference is preserved by the XOR with the postwhitening key K_2. For a known permutation $\mathcal{F}$, most combinations of input and output differences suggest only a small number of possible input and output values, but it is not easy to find them. To carry out the attack, all we have to do is to sample $2^n/D$ pairs of inputs to $\mathcal{F}$ whose difference is Δ, and with constant non-negligible probability we can find an output difference which already exists among the chosen data pairs. This equality suggests actual input and output values to/from $\mathcal{F}$ for that pair, and thus recovers the two keys. We note that a similar chosen-plaintext attack was suggested in [18] for constructions where $\mathcal{F}$ is keyed (where $DT \geq 2^{n+k-1}$ for a k-bit keyed $\mathcal{F}$).

This attack matches the time/data relationship of the lower bound, but it is not tight since it requires chosen plaintexts, whereas the lower bound allows known plaintexts. This discrepancy was handled ten years later by a new attack called *slide with a twist* which was developed by Alex Biryukov and David Wagner, and presented at EUROCRYPT 2000 [6]. By taking two Even–Mansour encryptions, sliding one of them and reversing the other, they showed how to attack the scheme with known instead of chosen plaintexts.[4] However, in order to find at least one slid pair, their attack requires at least $\Omega(2^{n/2})$ known plaintext/ciphertext pairs, and thus it could not be applied with a reasonable probability of success given any smaller number of known pairs.

These two cryptanalytic attacks were thus complementary: One of them matched the full time/data tradeoff curve but required chosen plaintexts, while the other could use known plaintexts but only if at least $\Omega(2^{n/2})$ of them were given. In the next section, we present the new slidex technique that closes this gap: it allows using any number of known plaintexts with the same time/data tradeoff as in the lower bound proof, thus providing an optimal attack on the Even–Mansour scheme.

3. The Slidex Attack and a Tight Bound on the Security of the Even–Mansour Scheme

In this section, we present the new slidex attack and use it to obtain a tight bound on the security of the Even–Mansour scheme. We start with a description of the slide with a twist attack on EM [6] which serves as a basis for our attack, and then we present the slidex technique and apply it to EM. For more information on slide attacks, we refer the reader to [4–6].

3.1. *The Slide with a Twist Attack*

The main idea of the slide with a twist attack on EM is as follows. Assume that two plaintexts P, P^* satisfy

$$P \oplus P^* = K_1.$$

In such a case, we have

$$E(P) = \mathcal{F}(P \oplus K_1) \oplus K_2 = \mathcal{F}\bigl(P^*\bigr) \oplus K_2,$$

and similarly,

$$E\bigl(P^*\bigr) = \mathcal{F}\bigl(P^* \oplus K_1\bigr) \oplus K_2 = \mathcal{F}(P) \oplus K_2$$

(see Fig. 1(a)). Hence,

$$E(P) \oplus E\bigl(P^*\bigr) = \mathcal{F}(P) \oplus \mathcal{F}\bigl(P^*\bigr),$$

or equivalently,

$$E(P) \oplus \mathcal{F}(P) = E\bigl(P^*\bigr) \oplus \mathcal{F}\bigl(P^*\bigr).$$

This relation allows mounting the following attack:

[4] The slide with a twist attack on EM is described in detail in Sect. 3.1.

1. Query both the E-oracle and the $\mathcal{F}$-oracle at the same $2^{(n+1)/2}$ known values[5] $P_1, P_2, \ldots$. Store in a hash table the pairs $(E(P_i) \oplus \mathcal{F}(P_i), i)$, sorted by the first coordinate.
2. For each collision in the table, i.e., $E(P_i) \oplus \mathcal{F}(P_i) = E(P_j) \oplus \mathcal{F}(P_j)$, check the guess $K_1 = P_i \oplus P_j$ and $K_2 = E(P_i) \oplus \mathcal{F}(P_j)$.

By the birthday paradox, it is expected that the data set contains a slid pair, i.e., a pair satisfying $P_i \oplus P_j = K_1$, with a non-negligible constant probability. For a random pair (P_i, P_j), the probability that $E(P_i) \oplus \mathcal{F}(P_i) = E(P_j) \oplus \mathcal{F}(P_j)$ is 2^{-n}, and thus, only a few collisions are expected in the table. These collisions include the collision induced by the slid pair, which suggests the correct values of K_1 and K_2. The data complexity of the attack is $D = 2^{(n+1)/2}$ known plaintexts, and the number of queries to $\mathcal{F}$ it requires is $T = 2^{(n+1)/2}$. Thus, $DT = 2^{n+1}$, which matches the lower bound up to a constant factor of 2.

3.2. *The New Slidex Attack*

The *slidex* attack is an enhancement of the slide with a twist technique, which makes it possible to use a smaller number of known plaintexts (i.e., queries to the E-oracle), in exchange for a higher number of queries to the $\mathcal{F}$-oracle. The basic idea of the attack is as follows: Assume that a pair of plaintexts P, P^* satisfies

$$P \oplus P^* = K_1 \oplus \Delta,$$

for some $\Delta \in \{0, 1\}^n$. In such a case,

$$E(P) = \mathcal{F}(P \oplus K_1) \oplus K_2 = \mathcal{F}\big(P^* \oplus \Delta\big) \oplus K_2,$$

and similarly,

$$E\big(P^*\big) = \mathcal{F}\big(P^* \oplus K_1\big) \oplus K_2 = \mathcal{F}(P \oplus \Delta) \oplus K_2$$

(see Fig. 1(b)). Hence,

$$E(P) \oplus E\big(P^*\big) = \mathcal{F}\big(P^* \oplus \Delta\big) \oplus \mathcal{F}(P \oplus \Delta),$$

or equivalently,

$$E(P) \oplus \mathcal{F}(P \oplus \Delta) = E\big(P^*\big) \oplus \mathcal{F}\big(P^* \oplus \Delta\big).$$

This allows mounting the following attack, for any $d \leq n$:

1. Query the E-oracle at $2^{(d+1)/2}$ arbitrary values (i.e., known plaintexts) $P_1, P_2, \ldots$.
2. Choose 2^{n-d} arbitrary values $\Delta_1, \Delta_2, \ldots$ of Δ. For each Δ_ℓ, query the $\mathcal{F}$-oracle at the values $\{P_i \oplus \Delta_\ell\}_{i=1,2,\ldots,2^{(d+1)/2}}$, store in a hash table the pairs $(E(P_i) \oplus \mathcal{F}(P_i \oplus \Delta_\ell), i)$, sorted by the first coordinate, and search for a collision.
3. For each collision in any of the hash tables, i.e., when P_i, P_j for which $E(P_i) \oplus \mathcal{F}(P_i \oplus \Delta_\ell) = E(P_j) \oplus \mathcal{F}(P_j \oplus \Delta_\ell)$ are detected, check the guess $K_1 = P_i \oplus P_j \oplus \Delta_\ell$ and $K_2 = E(P_i) \oplus \mathcal{F}(P_j \oplus \Delta_\ell)$.

[5] Formally, the adversary obtains known plaintext/ciphertext pairs $(P_i, E(P_i))$ and queries the $\mathcal{F}$-oracle at the value P_i.

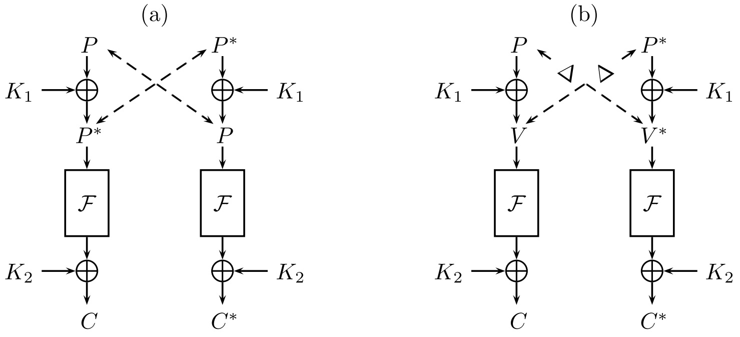

Fig. 1. (**a**) A twisted-slid pair; (**b**) A slidex pair.

Table 1. Comparison of results on the Even–Mansour scheme.

Known Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Guess and determine [13]	2	2^n	2	–
Slide with a twist [6]	$2^{n/2}$	$2^{n/2}$	$2^{n/2}$	–
Slidex (Sect. 3.2)	D	T	D	$DT = 2^n$
Chosen Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Differential [9]	D	T	D	$DT = 2^n$
Adaptive Chosen Plaintext Attacks				
Attack	Data	Time	Memory	Tradeoff
Slide with a twist (Sect. 6)	D	T	1	$DT = 2^n$, $D \geq 2^{n/2}$

For each triplet (P_i, P_j, Δ_ℓ), the probability that $P_i \oplus P_j \oplus \Delta_\ell = K_1$ is 2^{-n}. Since the data contains $2^d \cdot 2^{n-d} = 2^n$ such triplets, it is expected that with a non-negligible constant probability the data contains at least one *slidex triplet* (i.e., a triplet for which $P_i \oplus P_j \oplus \Delta_\ell = K_1$). On the other hand, since the probability of a collision in each hash table is 2^{d-n} and there are 2^{n-d} tables, it is expected that only a few collisions occur, and one of them suggests the correct key guess.

The number of queries to the E-oracle in the attack is $D = 2^{(d+1)/2}$, and the number of queries to the $\mathcal{F}$-oracle is $T = 2^{n-(d-1)/2}$. Thus, $DT = 2^{n+1}$, which matches the lower bound of [13] up to a constant factor of 2.

A summary of the complexities of all the old and new attacks on the Even–Mansour scheme appears in Table 1.

4. The Single-Key Even–Mansour Scheme

In this section, we analyze the single-key variant of the Even–Mansour scheme (abbreviated in the sequel as "SEM"), which has the same level of security while using only n

secret key bits (compared to $2n$ bits in EM). First, we define the scheme and show that the security proof of [13] can be adapted to yield a similar lower bound on its security. Then, we present a simple attack on the new scheme which matches the lower bound, thus proving its optimality.

We note that variants of SEM were considered (in different contexts) in several previous papers, but without proving the equivalence of SEM to the original two-key Even–Mansour scheme: In [18], Kilian and Rogaway studied a variant of SEM in which the internal permutation is keyed, and obtained a lower bound on its security in the indistinguishability (rather than key recovery) model. In [19], Kurosawa studied a variant of SEM in which the pre-/post-whitening keys are changed in each block. Finally, constructions similar to SEM were used in the design of several cryptographic primitives, including the stream cipher Salsa20 [2].

4.1. *Definition of the Scheme and Sketch of Its Security Proof*

Given a publicly known permutation $\mathcal{F}$ over n-bit strings and an n-bit secret key K, the Single-Key Even–Mansour (SEM) scheme is defined as follows:

$$SEM_K^{\mathcal{F}}(P) = \mathcal{F}(P \oplus K) \oplus K.$$

The attack model is the same as in the EM scheme. That is, the adversary can query an encryption/decryption E-oracle and an $\mathcal{F}$-oracle, and the complexity of an attack is determined by the number D of queries to the E-oracle and their type (known/chosen, etc.), and the number T of queries to the $\mathcal{F}$-oracle.

Surprisingly, the security proof of the EM scheme [13] holds almost without a change when we apply it to the single-key SEM variant. The only modification we have to make is to define a key K as *bad* with respect to sets of oracle queries $\mathcal{D}$ and $\mathcal{T}$ if there exist i, j such that either $P_i \oplus K = X_j$ or $C_i \oplus K = Y_j$, and K as good otherwise. It is easy to see that if $|\mathcal{D}| = s$ and $|\mathcal{T}| = t$, then at least $2^n - 2st$ keys are still "good" keys. Exactly the same proof as for EM shows that all the good keys are equally likely to be the right key, and the bounds on the success probability of an attack apply without change for SEM. Therefore, for any successful attack on SEM, we must have $DT = \Omega(2^n)$, which means that SEM provides the same security as EM, using only half as many key bits.

4.2. *A Simple Optimal Attack on SEM*

The slidex attack presented in Sect. 3 applies also to SEM, and is optimal since it uses only known plaintexts and matches everywhere the tradeoff curve of the security proof.

However, in the case of SEM, there is an even simpler attack (though, with the same complexity). Consider an encryption of a plaintext P through SEM, and denote the intermediate values in the encryption process by:

$$x = P, \qquad y = P \oplus K, \qquad z = \mathcal{F}(P \oplus K), \qquad w = E(P) = \mathcal{F}(P \oplus K) \oplus K.$$

Note that $x \oplus w = y \oplus z$. This allows mounting the following simple attack, applicable for any $D \leq 2^n$:

1. Query the $\mathcal{F}$-oracle at $2^n/D$ arbitrary values $X_1, X_2, \ldots, X_{2^n/D}$, and store in a hash table the values $(X_j \oplus \mathcal{F}(X_j), j)$, sorted by the first coordinate.

2. Query the E-oracle at D arbitrary values $P_1, P_2, \ldots, P_D$ insert the values $P_i \oplus E(P_i)$ to the hash table and search for a match.
3. If a match is found, i.e., $P_i \oplus E(P_i) = X_j \oplus \mathcal{F}(X_j)$, check the guess $K = P_i \oplus X_j$.

The analysis of the attack is exactly the same as that of the slide with a twist attack (see Sect. 3.1).

The security model of EM defined in [13] does not distinguish between precomputation and online computations, and thus, both EM and SEM enjoy the same level of security. However, we note that this attack has an advantage over the slidex attack, since its first step can be performed as a precomputation, thus reducing the time complexity of the on-line phase of the attack.

5. The Security of Other Variants of the Even–Mansour Scheme

In this section, we consider several natural variants of the Even–Mansour scheme, and analyze their security.

The first variant replaces the XOR operations with modular additions, which are not involutions and are thus immune to standard slide-type attacks (as noted in [6]). However, we show that a new *addition slidex* attack can break it with the same complexity as that of the slidex attack on the original EM scheme.

The second variant considers the case in which the mapping $\mathcal{F}$ is chosen as an involution. This is motivated by the fact that in many "real-life" implementations of the EM scheme we would like to instantiate $\mathcal{F}$ by a keyless variant of a block cipher. Since in Feistel structures and many other schemes (e.g., KHAZAD, Anubis, Noekeon) the only difference between the encryption and decryption processes is the key schedule, such schemes become involutions when we make them keyless. In this section, we show that this seemingly mild weakness of $\mathcal{F}$ can be used to mount a devastating attack on the EM scheme. In particular, we show that even when $\mathcal{F}$ is chosen uniformly at random among the set of all the possible involutions on n-bit strings, the adversary can recover the value $K_1 \oplus K_2$ with $O(2^{n/2})$ queries to the E-oracle and no queries at all (!) to the $\mathcal{F}$-oracle. This clearly violates the lower bound proof that no significant information about the key can be obtained unless $DT = \Omega(2^n)$ (which was proven for random permutations but seems intuitively to be equally applicable to random involutions), and is achieved by a new variant of the slide attack, which we call the *mirror slidex* attack.

After considering these two basic variants of EM, we consider combinations of them, such as "Addition Even–Mansour with a random involution as the permutation", and compare them with their single-key analogues. Our results are summarized in Table 2 which contains the security bounds and the matching attacks for 12 variants of the Even–Mansour construction.

5.1. *Even–Mansour with Addition*

Consider the following scheme:

$$AEM^{\mathcal{F}}_{K_1,K_2}(P) = \mathcal{F}(P + K_1) + K_2,$$

where $\mathcal{F}$ is a publicly known permutation over n-bit strings, and '+' denotes modular addition in the additive group Z_{2^n}. In the sequel, we call it "Addition Even–Mansour" (AEM).

It is clear that the lower bound security proof of EM holds without any change for AEM. Similarly, it is easy to see that Daemen's differential attack on EM [9] can be easily adapted to AEM, by replacing XOR differences with modular differences.

It may seem that the new variant has better security with respect to slide-type attacks. As noted in [6], ordinary slide attacks (and even the slide-with-a-twist attack) can be applied only for ciphers in which the secret key is inserted through a *symmetric* operation such as XOR, and not through modular addition. In the specific case of EM, the slide with a twist attack relies on the observation that if for two plaintexts P, P^*, we have $P^* = P \oplus K_1$, then surely, $P = P^* \oplus K_1$ as well. This observation fails for AEM: If $P^* = P + K_1$, then $P^* + K_1 = P + 2K_1 \neq P$ (unless $K_1 = 0$ or $K = 2^{n-1}$). The slidex attack presented in Sect. 3.2 fails against AEM for the same reason. Hence, it seems that none of the previously known attacks can break AEM in the *known plaintext* model.

We present an extension of the slidex attack, which we call *addition slidex*, which can break AEM with data complexity of D known plaintexts and time complexity of T $\mathcal{F}$-oracle queries, for any D, T such that $DT = 2^n$, hence showing that the security of AEM is identical to that of EM.

The basic idea of the attack is as follows: Assume that a pair of plaintexts P, P^* satisfies $P + P^* = -K_1 + \Delta$. (Note that somewhat counter intuitive, we consider the modular sum of the plaintexts rather than their modular difference!) In such a case,

$$E(P) = \mathcal{F}(P + K_1) + K_2 = \mathcal{F}\big(-P^* + \Delta\big) + K_2,$$

and similarly,

$$E\big(P^*\big) = \mathcal{F}\big(P^* + K_1\big) + K_2 = \mathcal{F}(-P + \Delta) + K_2.$$

Hence,

$$E(P) - E\big(P^*\big) = \mathcal{F}\big(-P^* + \Delta\big) - \mathcal{F}(-P + \Delta),$$

or equivalently,

$$E(P) + \mathcal{F}(-P + \Delta) = E\big(P^*\big) + \mathcal{F}\big(-P^* + \Delta\big). \tag{2}$$

Equation (2) allows us to mount an attack similar to the slidex attack, with the only change that instead of the values $(E(P_i) \oplus \mathcal{F}(P_i \oplus \Delta), i)$, the adversary stores in the hash table the values $(E(P_i) + \mathcal{F}(-P_i + \Delta), i)$.

We note that the addition slidex attacks applies not only to addition but to any *group operation*. In particular, its application to the XOR operation, which is the group operation in the additive group $(Z_2)^n$, yields the slidex attack presented in Sect. 3.2. Moreover, the attack can be extended to the case where two different group operations are used in the pre- and the post-whitening. For example, if XOR is used in the pre-whitening and modular addition is used in the post-whitening, the attack requires storing in the hash table the values $(E(P_i) + \mathcal{F}(P_i \oplus \Delta), i)$ and proceeds like the slidex attack.

5.2. *Even–Mansour with a Random Involution as the Permutation*

Let Involutional Even–Mansour (IEM) be the following scheme:

$$IEM^{\mathcal{I}}_{K_1,K_2}(P) = \mathcal{I}(P \oplus K_1) \oplus K_2,$$

where $\mathcal{I}$ is chosen uniformly at random amongst the set of involutions on n-bit strings. We present a new technique, which we call *mirror slidex*, that allows recovering the value $K_1 \oplus K_2$ using $2^{n/2}$ queries to the E-oracle, and with no queries to the $\mathcal{I}$-oracle.

The idea of the technique is as follows. Consider two input/output pairs (P, C), (P^*, C^*) for IEM. Assume that we have

$$P \oplus C^* = K_1 \oplus K_2. \tag{3}$$

In such a case,

$$P \oplus K_1 = C^* \oplus K_2,$$

and hence, since $\mathcal{I}$ is an involution,

$$\mathcal{I}(P \oplus K_1) = \mathcal{I}^{-1}\bigl(C^* \oplus K_2\bigr).$$

However, by the construction, we have

$$C = \mathcal{I}(P \oplus K_1) \oplus K_2 \quad \text{and} \quad P^* = \mathcal{I}^{-1}\bigl(C^* \oplus K_2\bigr) \oplus K_1,$$

and thus,

$$C \oplus K_2 = P^* \oplus K_1,$$

or equivalently,

$$P^* \oplus C = K_1 \oplus K_2 = P \oplus C^*,$$

where the last equality follows from Eq. (3). Therefore, assuming that $P \oplus C^* = K_1 \oplus K_2$, we must have

$$P \oplus C = P^* \oplus C^*.$$

This allows mounting a simple attack, similar to the slide with a twist attack. In the attack, the adversary queries the E-oracle at $2^{(n+1)/2}$ arbitrary values $P_1, P_2, \ldots,$ and stores in a hash table the pairs $(E(P_i) \oplus P_i, i)$, sorted by the first coordinate. It is expected that only a few collisions exist, and that with a non-negligible probability, one of them results from a pair (P_i, P_j), for which $P_i \oplus E(P_j) = K_1 \oplus K_2$.

Therefore, the attack supplies the adversary with only a few possible values of $K_1 \oplus K_2$, after performing $2^{(n+1)/2}$ queries to the E-oracle and no queries at all to the $\mathcal{I}$-oracle. As we show later, the adversary cannot obtain K_1 or K_2 themselves (without additional effort or data), but at the same time, the adversary does learn a nontrivial information about the key, which contradicts the security proof of the original EM scheme.

Where the Security Proof Fails One may wonder, which part of the formal security proof fails when $\mathcal{F}$ is an involution. It turns out that the only part that fails is the argument in the first step of the proof showing that all good keys are equally likely to be the right key. Recall that in order to show this, one has to show that the probability

$$p = \Pr_{K,\mathcal{F}}\big[(K,\mathcal{F}) \text{ is consistent with } \mathcal{D}, \mathcal{T} | K = k\big]$$

is the same for all good keys. In the case of EM, p is shown to be the probability of consistence of a random permutation $\mathcal{F}$ with $s+t$ given distinct input/output pairs, which indeed does not depend on k (since such pairs are independent). In the case of IEM, the input/output pairs may be dependent, since it may occur that an encryption query to the E-oracle results in querying $\mathcal{I}$ at some value x, while a decryption query to the E-oracle results in querying $\mathcal{I}^{-1}$ at the same value x. Since $\mathcal{I}$ is an involution, these queries are not independent and thus, the probability p depends on whether such dependency has occurred, and this event does depend on k. An examination of the mirror slidex attack shows that this property is exactly the one exploited by the attack.

It is interesting to note that in the single-key case (i.e., for SEM where $\mathcal{F}$ is an involution, which we denote by SIEM), such an event cannot occur, as in order to query $\mathcal{I}$ and $\mathcal{I}^{-1}$ at the same value, one must query E and E^{-1} at the same value. Since in the single-key case, the entire construction is an involution, such two queries result in the same answer for any value of the secret key, and hence, do not create dependency on the key. It can be shown, indeed, that the security proof does hold for SIEM and yields the same security bound, thus showing that in the case of involutions, the single-key variant is clearly more efficient than the original two-key variant! Moreover, it can be noticed that in the case of EM, after the adversary recovers the value $K_1 \oplus K_2$, the encryption scheme becomes equivalent to a single-key Even–Mansour scheme with the key K_1, i.e., $E'(P) = \mathcal{I}(P \oplus K_1) \oplus K_1$. Thus, using two different keys in this case is totally obsolete, and also creates a security flaw which can be deployed by an adversary if the keys K_1 and K_2 are used also in other systems.

We note that SIEM provides an example of the gap between the indistinguishability security notion and the cost of finding a key. Obviously, one can easily distinguish SIEM from a random permutation using two adaptive queries with an extremely high probability (as SIEM is an involution). At the same time, the lower bounds of the Even–Mansour security proof assure us that it is impossible to decrypt a ciphertext C encrypted by SIEM or to produce a new (P, C) pair for SIEM without first obtaining $DT = \Omega(2^n)$ queries.

5.3. *Addition Even–Mansour with an Involution as the Permutation*

In this subsection, we consider a combination of the two variants discussed in the previous subsections, i.e., AEM where $\mathcal{F}$ is a random involution. We abbreviate this variant as AIEM.

It can be easily shown that the mirror slidex attack can be adapted to the case of AIEM, by modifying the assumption to $C^* - P = K_1 + K_2$, and the conclusion to $P + C = P^* + C^*$. The attack allows recovering the value $K_1 + K_2$, and then the scheme becomes equivalent to a *conjugation* EM scheme with a single key:

Table 2. Summary of the security of the 12 Even–Mansour variants.

	$\mathcal{F}$ is a Random Permutation		$\mathcal{F}$ is a Random Involution	
	Single Key	Two Keys	Single Key	Two Keys
Pre/Post-Whitening XOR	SEM	EM	SIEM	IEM
Provable Security Bound	$DT \geq 2^n$	$DT \geq 2^n$	$DT \geq 2^n$	$DT \geq 2^n$
Best Attack	Slidex (or Sect. 4.2)	Slidex	Slidex	Mirror Slidex
	(matches bound)	(matches bound)	(matches bound)	Retrieves $K_1 \oplus K_2$
				with $D = 2^{n/2}$
Pre/Post-Whitening Addition	ASEM	AEM	ASIEM	AIEM
Provable Security Bound	$DT \geq 2^n$	$DT \geq 2^n$	N/A	$DT \geq 2^n$
Best Attack	Addition Slidex	Addition Slidex	Addition Slidex	Addition Slidex
	(matches bound)	(matches bound)	Complete break	Retrieves $K_1 + K_2$
			$D = 2^{n/2}$	with $D = 2^{n/2}$
Conjugation	CSEM	CEM	CSIEM	CIEM
Pre/Post-Whitening				
Provable Security Bound	$DT \geq 2^n$	$DT \geq 2^n$	N/A	$DT \geq 2^n$
Best Attack	Addition Slidex	Addition Slidex	Addition Slidex	Addition Slidex
	(matches bound)	(matches bound)	(matches bound)	Retrieves $K_1 + K_2$
				with $D = 2^{n/2}$

$CSIEM(P) = \mathcal{I}(P + K_1) - K_1$, and it can be shown that the security proof of EM applies also to CSIEM. Thus, the security of AEM under the assumption that $\mathcal{F}$ is an involution is identical to that of the original EM.

An interesting phenomenon is that in the involution case, the security of single-key AEM (which we denote by ASIEM) is much worse than that of AIEM. Indeed, the mirror slidex attack allows recovering $K_1 + K_1 = 2K_1$, and hence finding K_1 (up to the value of the MSB) which breaks the scheme completely. This suggests that in the case of addition, the "natural" variant of single-key AEM is the conjugation variant, i.e., $CSEM(P) = \mathcal{F}(P + K_1) - K_1$, for which the security proof of EM indeed applies even if $\mathcal{F}$ is an involution, as mentioned above.

In Table 2, we list 12 variants of the Even–Mansour construction (single key/two keys, random permutation/random involution, and whether the keys are XORed, added, or conjugated). For each variant we list the security bound (if possible), and the attack that matches the bound.

6. Memoryless and Ciphertext-Only Attacks on the Even–Mansour Scheme

In this section, we consider two attack scenarios in which the adversary is severely restricted—memoryless attacks in which the adversary can use only a few cells of memory, and ciphertext-only attacks in which the adversary is given only a partial information about the plaintexts (e.g., only knows that the plaintext consists of words in English). We show that in both scenarios, we can obtain the tradeoff curve $DT = \Omega(2^n)$, but only for part of the possible values of D.

6.1. *Memoryless Attacks on the Even–Mansour Scheme*

All previous papers on the Even–Mansour scheme, including the lower bounds proved by the designers [13], Daemen's attack [9], and Biryukov–Wagner's slide attack [6], considered only the data and time complexities of attacks, but not the memory complexity. Analysis of the previously proposed attacks shows that in all of them, the memory complexity is at least $\min\{D, T\}$, where D is the data complexity (i.e., the number of E-queries) and T is the time complexity (i.e., the number of $\mathcal{F}$-queries). Thus, it is natural to ask whether the memory complexity can also be inserted into the lower bound security proofs, e.g., in the form $M \geq \min(D, T)$.

In this section, we show that such a general lower bound cannot exist, by constructing an attack with data and time complexities of $O(2^{n/2})$, and with only a constant memory complexity. The attack is a memoryless variant of the slide with a twist attack described in Sect. 3.1. Recall that the main step of the slide with a twist attack is to find collisions of the form $E(P) \oplus \mathcal{F}(P) = E(P^*) \oplus \mathcal{F}(P^*)$.

We observe that such collisions can be found in a memoryless manner. We treat the function

$$\mathcal{G} : P \to E(P) \oplus \mathcal{F}(P)$$

as a random function, and apply Floyd's cycle finding algorithm [15] (or any of its variants, such as Nivasch's algorithm [25]) to find a collision in $\mathcal{G}$. The attack algorithm is as follows:

1. Query the E-oracle at a sequence of $O(2^{n/2})$ adaptively chosen values $P_1, P_2, \ldots$ such that P_1 is arbitrary and for $k > 1$, $P_i = E(P_{i-1}) \oplus \mathcal{F}(P_{i-1})$. (Here, after each query to the E-oracle, the adversary queries the $\mathcal{F}$-oracle at the same value and uses its answer in choosing the next query to the E-oracle.)
2. Use Floyd's cycle finding algorithm to find P_i, P_j such that $E(P_i) \oplus \mathcal{F}(P_i) = E(P_j) \oplus \mathcal{F}(P_j)$.
3. For each colliding pair, check the guess $K_1 = P_i \oplus P_j$ and $K_2 = E(P_i) \oplus \mathcal{F}(P_j)$.

The analysis of the attack is identical to the analysis of the slide with a twist attack. The memory complexity is negligible, and the data and time complexities remain $O(2^{n/2})$. The only downside of this algorithm is the fact that the queries to the E-oracle are chosen adaptively, whereas in the slide with a twist attack we could choose arbitrary queries to the E-oracle.

6.2. *Ciphertext-Only Attacks on the Even–Mansour Scheme*

In ciphertext-only attacks, the assumption is that the adversary is not given any plaintext/ciphertext pairs, but only knows the ciphertexts and some partial information on the plaintexts, e.g., that the plaintexts are English words encoded by ASCII characters. Such a situation is very realistic in passive eavesdropping attacks.

We show that if the partial information on each plaintext contains k linear equations on its bits, then a variant of the slidex attack can break the scheme with time complexity T and data complexity D such that $DT = O(2^n)$, as long as $D \leq 2^k$. In particular, in the case of English words encoded by ASCII characters, it is known that the most significant bit of each byte equals zero in the ASCII encoding, which yields $n/8$ linear equations in the bits of each n-bit plaintext. Hence, the tradeoff curve $DT = O(2^n)$ can be obtained for all $D \leq 2^{n/8}$.

Recall that in the slidex attack on EM, the adversary looks for collisions in the function $P_i \mapsto E(P_i) \oplus \mathcal{F}(P_i \oplus \Delta_\ell)$ (for a fixed Δ_ℓ) and uses them to find pairs of plaintexts (P_i, P_j) such that $P_i \oplus P_j \oplus \Delta_\ell = K_1$. In the ciphertext-only attack, the adversary cannot check this function, as she does not know the value $P_i \oplus \Delta_\ell$. However, we observe that the same attack procedure can be performed in the inverse direction, i.e., looking at collisions in the function $C_i \mapsto E^{-1}(C_i) \oplus \mathcal{F}^{-1}(C_i \oplus \Delta_\ell)$ in order to find pairs of ciphertexts (C_i, C_j) such that $C_i \oplus C_j \oplus \Delta_\ell = K_2$.

As the ciphertexts are fully known to the adversary, she can compute the value $\mathcal{F}^{-1}(C_i \oplus \Delta_\ell)$. The value $E^{-1}(C_i)$ is not known, but, by the assumption, the adversary knows k linear equations in its bits (e.g., knows that the most significant bit of each byte equals zero in the English ASCII case). Hence, she can search for a collision between the values $E^{-1}(C_i) \oplus \mathcal{F}^{-1}(C_i \oplus \Delta_\ell)$ in the k known linear combinations of bits. Each such partial collision (C_i, C_j) suggests a value $K_2 = C_i \oplus C_j \oplus \Delta_\ell$ for the key K_2, and these suggestions can be checked easily.[6]

As the total number of triplets (C_i, C_j, Δ_ℓ) examined in the attack is 2^n (see the analysis of the slidex attack in Sect. 3.2), the expected number of partial collisions is $2^n \cdot 2^{-k} = 2^{n-k}$. Thus, the phase of examining key suggestions arising from the partial collisions, which is the only additional phase compared to the slidex attack, requires time complexity of 2^{n-k} and no additional data complexity. Therefore, for $D \leq 2^k$ (for which the time complexity of the basic slidex attack is $T \geq 2^{n-k}$), the overall data and time complexities of the attack satisfy the tradeoff curve $DT = O(2^n)$.

An interesting question for future research is whether partial information on the plaintexts which cannot be represented in the form of linear equations (e.g., quadratic equations in the plaintext bits) can also be used by some variant of the slidex attack.

7. Further Applications of the Mirror Slide Attack

In this section, we present the general framework of the *mirror slidex* attack that was presented in Sect. 5.2 in the special case of the Even–Mansour scheme. We show that the mirror slidex attack generalizes the *slide with a twist* attack [6] and can be combined with the *complementation slide* attack [6]. We apply the new technique to a 20-round variant of the block cipher GOST [28], and to variants of the DESX cryptosystem [27] in which the subkeys of the internal DES cipher are replaced by a 2-round or a 4-round self-similar sequence.

7.1. *The General Framework*

The mirror slidex attack applies to block ciphers that can be decomposed as a cascade of three sub-ciphers: $E = E_2 \circ E_1 \circ E_0$, where the middle layer E_1 is an involution, i.e., $E_1 = (E_1)^{-1}$.[7]

[6] Note that the attack allows recovering only the key K_2, and k bits of partial information on the key K_1 which correspond to the partial information on the plaintexts. The rest of the key can be found using statistical information on the plaintexts, as the scheme is reduced to a Vigenére cipher.

[7] We note that the attack can be applied also if E_1 has some other symmetry properties, as shown in Sect. 7.3 below.

Let E be such a cipher, and assume that for two plaintext/ciphertext pairs (P, C), (P^*, C^*), we have

$$E_0(P) = E_2^{-1}\bigl(C^*\bigr). \tag{4}$$

In such case, since E_1 is an involution,

$$E_1\bigl(E_0(P)\bigr) = E_1^{-1}\bigl(E_2^{-1}\bigl(C^*\bigr)\bigr).$$

By the construction, this implies that

$$E_2^{-1}(C) = E_1\bigl(E_0(P)\bigr) = E_1^{-1}\bigl(E_2^{-1}\bigl(C^*\bigr)\bigr) = E_0\bigl(P^*\bigr). \tag{5}$$

If Eq. (4) holds (and thus, Eq. (5) also holds), the pair (P, P^*) is called a *mirror slid pair*.

The way to exploit mirror slid pairs in a cryptanalytic attack is similar to standard slide-type attacks [5,6]: The adversary asks for the encryption of $2^{(n+1)/2}$ known plaintexts $P_1, P_2, \ldots$ (where n is the block size of E) and denotes the corresponding ciphertexts by $C_1, C_2, \ldots$. For each pair (P_i, P_j), the adversary assumes that it is a mirror slid pair and tries to solve the system of equations:

$$\begin{cases} C_j = E_2(E_0(P_i)), \\ C_i = E_2(E_0(P_j)) \end{cases} \tag{6}$$

(which is equivalent to Eqs. (4) and (5)). If E_0 and E_2 are "simple enough", the adversary can solve the system efficiently and recover the key material used in E_0 and E_2.

If the amount of subkey material used in E_0 and E_2 is at most n bits (in total), it is expected that at most a few of the systems of equations generated by the 2^n plaintext pairs are consistent (since the equation system is a $2n$-bit condition). One of them is the system generated by the mirror slid pair, which is expected to exist in the data with a constant probability since the probability of a random pair to be a mirror slid pair is 2^{-n}. Hence, the adversary obtains only a few suggestions for the key, which contain the right key with a constant probability. If the amount of key material used in E_0 and E_2 is bigger than n bits, the adversary can still find the right key, by enlarging the data set by a small factor and using key ranking techniques (exploiting the fact that the right key is suggested by all mirror slid pairs, while the other pairs suggest "random" keys).

The data complexity of the attack is $O(2^{n/2})$ known plaintexts, and its time complexity is $O(2^n) \cdot t$, where t is the time required for solving the system (6).

We note that the attack can be applied even when E_0 and E_2 are not "simple" ciphers using a meet-in-the-middle attack. If both E_0 and E_2 use $\kappa \le n$ key bits at most, one can try and find the solutions to the above set of equations in time $\min\{O(2^{n+\kappa}), O(2^{n/2+2\kappa})\}$.[8]

[8] One can either take all plaintext/ciphertext pairs and partially encrypt the plaintext under all 2^κ keys for E_0 and partially decrypt the ciphertext under all 2^κ keys for E_2 to find the mirror pairs. Another option is to

7.2. *The Slide with a Twist Attack and an Application to 20-Round GOST*

The first special case of the mirror slidex framework we consider is where in the subdivision of E, we have $E_2 = \mathit{Identity}$. In such a case, the system of equations presented above is simplified to

$$\begin{cases} C_j = E_0(P_i), \\ C_i = E_0(P_j). \end{cases} \tag{7}$$

It turns out that in this case, the attack is reduced exactly to the slide with a twist attack presented in [6]! (Though, in [6] the attack is described in a different way.)

A concrete example of this case is a reduced-round variant of the block cipher GOST [28] that consists of the last 20 of its 32 rounds. It is well-known that the last 16 rounds of GOST compose an involution, and hence, this variant can be represented as $E = E_1 \circ E_0$, where E_0 is 4-round GOST, and E_1 (which is the last 16 rounds of GOST) is an involution.[9] As shown in [10], a 4-round variant of GOST can be broken with two plaintext/ciphertext pairs and time complexity of 2^{12} encryptions. Therefore, the mirror slidex attack can break this 20-round variant of GOST with data complexity of 2^{33} known plaintexts (since the block size of GOST is 64 bits), and time complexity of $2^{65} \cdot 2^{12} = 2^{77}$ encryptions.[10]

We note that a similar attack was described in [6] using the slide with a twist technique, but only on a 20-round version of a modified variant of GOST called GOST⊕ in which the key addition is replaced by XOR.

7.3. *Combination with the Complementation Slide Attack and Application to 2K-DESX*

In this subsection, we consider the case where E_1 is not an involution, but rather a Feistel cipher with a 2-round self-similarity property (see Fig. 2). Such a cipher (but without the key whitening) was considered in [6], and it was shown that it can be broken with complexity of $O(2^{n/2})$, using a technique called *complementation slide*.[11] We show that

try for each pair of plaintexts (P_i, P_j) to solve the system

$$\begin{cases} E_2^{-1}(C_j) = E_0(P_i), \\ E_2^{-1}(C_i) = E_0(P_j) \end{cases}$$

which can be easily done in a meet-in-the-middle approach in time 2^{κ} for each (P_i, P_j).

[9] We note that due to the Feistel structure of GOST, we do not have $E_1 \circ E_1 = Id$, but rather $E_1 \circ \mathit{swap} \circ E_1 = Id$. This can be handled easily by inserting swap to the left-hand side of Eq. (7). The same correction can be performed in the other Feistel constructions discussed in the sequel.

[10] We note that the mirror slide attack allows recovering only the subkeys K_5, K_6, K_7, K_8. However, the remaining key bits can be recovered easily by an auxiliary technique using the fact that the knowledge of (K_5, K_6, K_7, K_8) allows reducing the cipher to the 16 last rounds of GOST which compose an involution. For example, the adversary can look for one of the 2^{32} fixed points of the reduced cipher, use the fact that for most of these fixed points, the intermediate state after 8 rounds is of the form (x, x) for some 32-bit value x, guess the value of x and recover the keys K_1, K_2, K_3, K_4 by a 4-round attack. The data complexity of this procedure is 2^{32} known plaintexts which can be obtained from the plaintexts used in the mirror slide attack, and the time complexity is $2^{32} \cdot 2^{12} = 2^{44}$ encryptions.

[11] We note that in [6] a Feistel cipher with a 2-round self-similarity property is also attacked using the slide with a twist technique (with even better results). In the attack, the cipher is represented as $E = E_1 \circ E_0$, where E_0 is a single round and E_1 is a $(2m - 1)$-round Feistel structure with 2-round self-similarity, which can be

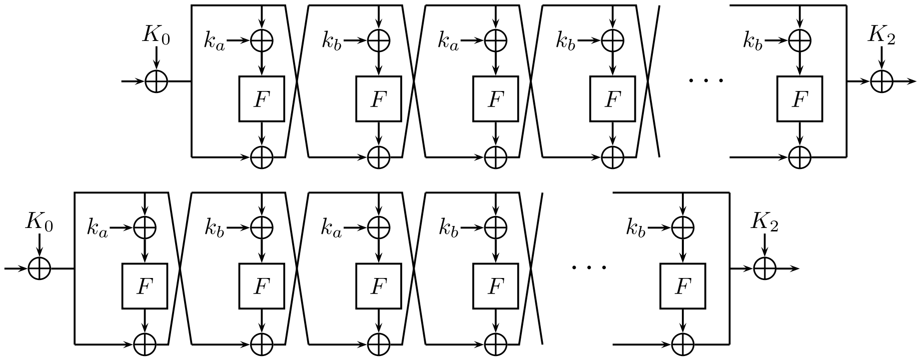

Fig. 2. Pre-/post-whitened cipher with 2-round self similarity.

the complementation slide technique can be combined with the mirror slidex technique to yield an attack on the scheme including pre- and post- key whitening, with the same complexity.

A concrete example of such construction one may consider is a variant of DESX [27] in which the subkeys generated by the DES key schedule are replaced by the periodic sequence $(k_a, k_b, k_a, k_b, \ldots)$. Using the terminology of [5,6], this variant can be called 2K-DESX. For the sake of simplicity, we demonstrate the attack on the example of 2K-DESX.

Consider two plaintext/ciphertext pairs (P, C), (P^*, C^*) of 2K-DESX, and assume that

$$P \oplus C^* = K_0 \oplus K_2 \oplus (k_a \oplus k_b || k_a \oplus k_b),$$

where $||$ denotes concatenation of bit strings. In such a case,

$$P \oplus K_0 = \big(C^* \oplus K_2\big) \oplus (k_a \oplus k_b || k_a \oplus k_b). \tag{8}$$

We would like to apply E_1 to the left-hand side and E_1^{-1} to the right-hand side, like in the standard mirror slidex attack. In our case, E_1 is not an involution. However, this is compensated by the term $(k_a \oplus k_b || k_a \oplus k_b)$ in the right-hand side of the equation. Indeed, in the first round of E_1, the subkey is k_a, and thus, the input to the F-function is $P_R \oplus K_{0R} \oplus k_a$ (where X_R denotes the right half of X). On the other side, the subkey in the first round of E_1^{-1}, which is the subkey in the last round of E_1, is k_b, and hence, the input to the F-function is $C^*_R \oplus K_{2R} \oplus k_b$. Therefore, by Eq. (8), the two inputs are equal. A similar analysis shows that equality holds for the inputs of the F-functions in all rounds, and thus,

$$E_1(P \oplus K_0) = E_1^{-1}\big(C^* \oplus K_2\big) \oplus (k_a \oplus k_b || k_a \oplus k_b),$$

easily seen to be an involution. As described in Sect. 7.2, such an attack can be viewed as a special case of the mirror slidex attack. We do not consider it in this subsection since the existence of pre- and post-whitening raises its time complexity to $\Theta(2^n)$, while the complexity of our attack on this cipher is $O(2^{n/2})$.

or equivalently,

$$C \oplus K_2 = P^* \oplus K_0 \oplus (k_a \oplus k_b || k_a \oplus k_b). \tag{9}$$

XORing Eqs. (8) and (9), we get

$$C \oplus C^* = P \oplus P^*.$$

This allows applying an attack similar to the attack on IEM and recovering the value $K_0 \oplus K_2 \oplus (k_a \oplus k_b || k_a \oplus k_b)$ with data and time complexities of $O(2^{n/2})$.

7.4. *Application to a Variant of 4K-DESX*

The last case we consider is a variant of DESX in which the number of rounds in DES is changed to $4m + 1$, and the subkeys are replaced by the sequence $(k_a, k_b, k_c, k_d)^m, k_a$. We show that another combination of the complementation slide technique with the mirror slidex technique allows breaking this variant with data and time complexity of $O(2^{n/2})$.

Consider two plaintext/ciphertext pairs (P, C), (P^*, C^*), and assume that

$$P \oplus C^* = K_0 \oplus K_2 \oplus (k_b \oplus k_d || 0),$$

where $||$ denotes concatenation of bit strings. In such a case,

$$P \oplus K_0 = \bigl(C^* \oplus K_2\bigr) \oplus (k_b \oplus k_d || 0). \tag{10}$$

We apply E_1 to the left-hand side of the equation, and E_1^{-1} to the right-hand side of the equation. In the first round of E_1, the subkey is k_a, and thus, the input to the F-function is $P_R \oplus K_{0R} \oplus k_a$. The subkey in the first round of E_1^{-1} is also k_a, and hence, the input to the F-function in that round is $C^*_R \oplus K_{2R} \oplus k_a$. Therefore, by Eq. (10), the two inputs are equal. In the second round of E_1 and E_1^{-1}, the subkey in E_1 is k_b, while the subkey in E_1^{-1} is k_d. However, this difference is canceled with the term $k_b \oplus k_d$ in Eq. (10). A similar analysis shows that equality holds for the inputs of the F-functions in all rounds, and thus,

$$E_1(P \oplus K_0) = E_1^{-1}\bigl(C^* \oplus K_2\bigr) \oplus (k_b \oplus k_d || 0),$$

or equivalently,

$$C \oplus K_2 = P^* \oplus K_0 \oplus (k_b \oplus k_d || 0). \tag{11}$$

XORing Eqs. (10) and (11), we get

$$C \oplus C^* = P \oplus P^*,$$

and the attack can be concluded as in the previous case and retrieve the value $K_0 \oplus K_2 \oplus (k_b \oplus k_d || 0)$.

We note that this technique does not apply to the standard variant of 4K-DESX, in which the subkeys are $(k_a, k_b, k_c, k_d)^m$ (without an additional subkey k_a at the end). The reason is that the rate of symmetry between E_1 and E_1^{-1} is insufficient. While the asymmetry in the first two rounds can be compensated by adding the term $(k_a \oplus k_d||k_b \oplus k_c)$ to the equation, the inputs to the F-function in the third round will not be equal anymore.

8. Further Applications of the Addition Slidex Attack

In Sect. 5, we presented two new slide-type attacks that are applicable to ciphers in which the subkeys are inserted through modular addition (rather than XOR). The first was a variant of the *slidex* attack that was used in Sect. 5.1 to attack AEM, i.e., an Even–Mansour scheme in which the key XOR is replaced by modular addition. The second was a variant of the *mirror slidex* attack that was used in Sect. 5.3 to attack AIEM, i.e., AEM in which the internal permutation is an involution. These two attacks can be considered as special cases of a more general technique which we call the *addition slidex* attack. The main feature of the technique (that appears in both special cases) is that the relation between the elements of a slid pair concerns their modular sum, rather than their difference (as one may expect in light of the standard slide-type attacks).

In this section, we present another application of the *addition slidex* technique. The attack targets Addition DESX, i.e., a variant of DESX [27] in which the whitening keys are inserted using modular addition (instead of XOR). We show that while this variant seems to be as secure as DESX, it can be broken using only two related keys and practical complexity of either 2^{34} in the chosen plaintext model, or 2^{43} in the known plaintext model. The attack exploits the well-known *complementation property* of DES, namely, that for any P, K,

$$DES_K(P) = \overline{DES_{\bar{K}}(\bar{P})},$$

where $\bar{X}$ denotes the bitwise complement of X (i.e., $\bar{X} = X \oplus FF\ldots FF_x = 2^{64} - 1 - X$). It is interesting to note that while in the cases of DES and DESX, this property can be used only either for a distinguishing attack or for speeding up exhaustive key search by a factor of 2, in our case it can be deployed to mount a key recovery attack.

After presenting the attack on Addition DESX, we show that a slightly modified variant of the attack applies (with the same complexities) to another variant of DESX in which the key pre/post whitenings are replaced by key-dependent decorrelation modules [31].

8.1. *Attack on Addition DESX*

The addition DESX block cipher is defined as

$$E_{K_0,K_1,K_2}(P) = K_2 + DES_{K_1}(P + K_0),$$

where '+' denotes addition modulo 2^{64}. The basic idea of the attack is as follows. Let (P, C), (P^*, C^*) be two plaintext/ciphertext pairs, such that P is encrypted under (K_0, K_1, K_2) and P^* is encrypted under $(K_0, \overline{K_1}, K_2)$. Assume that the pair (P, P^*) satisfies

$$P + P^* + 2K_0 \equiv 2^{64} - 1 \pmod{2^{64}}. \qquad (12)$$

In such a case, we have

$$P + K_0 = \overline{P^* + K_0}.$$

By the complementation property, this implies

$$DES_{K_1}(P + K_0) = \overline{DES_{\overline{K_1}}\big(P^* + K_0\big)},$$

or equivalently,

$$DES_{K_1}(P + K_0) + DES_{\overline{K_1}}\big(P^* + K_0\big) \equiv 2^{64} - 1 \pmod{2^{64}}.$$

This, in turn, implies

$$C + C^* = E_{K_0,K_1,K_2}(P) + E_{K_0,\overline{K_1},K_2}\big(P^*\big) \equiv 2^{64} - 1 + 2K_2 \pmod{2^{64}}. \qquad (13)$$

Equation (13) cannot be exploited directly (like in all previous attacks) since the value of K_2 is not known to the adversary. However, we observe that since the right hand side of Eq. (13) does not depend on P and P^*, it can be canceled using another pair of plaintexts.

Let (P, C), (P^*, C^*) be plaintext/ciphertext pairs such that the pair (P, P^*) satisfies Eq. (12), and let $a \in Z_{2^{64}}$ be arbitrary. Consider the encryptions of $P + a$ and $P^* - a$ under the keys (K_0, K_1, K_2) and $(K_0, \overline{K_1}, K_2)$, respectively, and denote the corresponding ciphertexts by C' and C'^*. It is clear that the pair $(P + a, P^* - a)$ also satisfies Eq. (12). Hence, we have

$$C' + C'^* \equiv 2^{64} - 1 + 2K_2 \pmod{2^{64}}. \qquad (14)$$

Combining Eqs. (13) and (14), we get

$$C + C^* = C' + C'^*,$$

or equivalently,

$$C - C' = C'^* - C^*.$$

This allows mounting the following attack:

1. Choose some arbitrary $a \in Z_{2^{64}}$.[12]
2. Ask for the encryption of 2^{32} arbitrary plaintexts $P_1, P_2, \ldots$ under the key (K_0, K_1, K_2), and denote the corresponding ciphertexts by $(C_1, C_2, \ldots)$. Ask for the encryption of the 2^{32} plaintexts $P_1 + a, P_2 + a, \ldots$ under the same key, and denote the corresponding ciphertexts by $(C_1', C_2', \ldots)$. Store in a hash table the pairs $((C_i - C_i'), i)$, sorted by the first coordinate.
3. Ask for the encryption of 2^{32} arbitrary plaintexts $P_1^*, P_2^*, \ldots$ under the key $(K_0, \overline{K_1}, K_2)$, and denote the corresponding ciphertexts by $(C_1^*, C_2^*, \ldots)$. Ask for the encryption of the 2^{32} plaintexts $P_1^* - a, P_2^* - a, \ldots$ under the same key, and denote the corresponding ciphertexts by $(C_1'^*, C_2'^*, \ldots)$. Insert the values $C_j'^* - C_j^*$ into the hash table and search for collisions.

[12] For example, if the encryption is performed in counter mode, it may be desirable to choose $a = 1$.

4. For each collision in the table, i.e., $C_i - C'_i = C'^*_j - C^*_j$, check the guess $2K_0 = 2^{64} - 1 - P_i - P^*_j \pmod{2^{64}}$ and $2K_2 = C_i + C^*_j - (2^{64} - 1) \pmod{2^{64}}$.

As in the previous attacks, it is expected that only a few collisions occur, and that with a constant probability, one of them suggests the right key (K_0, K_2). A key guess suggested by the pair (P_i, P^*_j) can be checked by choosing another $a' \in Z_{2^{64}}$, asking for the encryption of $P_i + a'$ and $P^*_j - a'$ under the keys (K_0, K_1, K_2) and $(K_0, \overline{K_1}, K_2)$, respectively, and checking whether the corresponding ciphertexts (denoted by C''_i and C''^*_j) satisfy

$$C_i - C''_i = C''^*_j - C^*_j.$$

If the equation is satisfied, then the pair (P_i, P^*_j) satisfies Eq. (12) with overwhelming probability, and thus, the suggestion for (K_0, K_2) is correct (with the same probability). The value of K_1 can be found using auxiliary techniques (e.g., a differential or a linear attack on DES). The data complexity of the attack is 2^{34} chosen plaintexts encrypted under two keys, and its memory and time complexities are about 2^{34} (except for the part of recovering K_1). As in the previous cases, the attack can be transformed into a memoryless attack with the same time complexity, where the data complexity is 2^{34} adaptively chosen plaintexts.

A Known-Plaintext Variant of the Attack A variant of the attack can be performed in the known plaintext model without enlarging the number of examined plaintexts, at the expense of enlarging the time complexity. The attack uses the fact that the procedure described above succeeds for any value of a, and thus, the adversary can exploit many values of a simultaneously. The algorithm of the known plaintext attack is as follows:

1. Ask for the encryption of two pools of 2^{32} arbitrary plaintexts each under the key (K_0, K_1, K_2), and denote the plaintext/ciphertext pairs in the pools by $(P_1, C_1), (P_2, C_2), \ldots$, and $(P'_1, C'_1), (P'_2, C'_2), \ldots$, respectively.
2. Ask for the encryption of two pools of 2^{32} arbitrary plaintexts each under the key $(K_0, \overline{K_1}, K_2)$, and denote the plaintext/ciphertext pairs in the pools by $(P^*_1, C^*_1), (P^*_2, C^*_2), \ldots$, and $(P'^*_1, C'^*_1), (P'^*_2, C'^*_2), \ldots$, respectively.
3. Search for a four-collision of 128-bit values, of the form

$$\left(P_i - P'_j + P^*_k - P'^*_\ell, C_i - C'_j + C^*_k - C'^*_\ell\right) = 0. \tag{15}$$

4. For each such collision, check the guess $2K_0 = 2^{64} - 1 - P_i - P^*_j \pmod{2^{64}}$ and $2K_2 = C_i + C^*_j - (2^{64} - 1) \pmod{2^{64}}$.

It is expected that among the 2^{128} examined plaintext quartets, about 2^{64} quartets satisfy the equation $P_i - P'_j + P^*_k - P'^*_\ell = 0$, and thus can be represented as $(P_i, P_i + a, P^*_k, P^*_k - a)$, for $a = P'_j - P_i$. Thus, with a constant probability, in at least one of these quartets, P_i and P^*_k satisfy Eq. (12). For such a quartet, we must have $C_i - C'_j + C^*_k - C'^*_\ell = 0$, and thus, it generates a collision of the form needed for the attack. On the other hand, the probability that Eq. (15) is satisfied for a random quartet is 2^{-128}, and hence, it is expected that only a few collisions exist, and at least one of them suggests the right key.

The data complexity of the attack is 2^{34} known plaintexts encrypted under two keys, and the memory and time complexities are about 2^{64}.

As the collision search performed in the attack is a solution of a standard *generalized birthday* problem, one can obtain a time/memory/data tradeoff using the improved algorithms for the generalized birthday problem presented by Wagner [32]. For example, if the data complexity is increased to $2^{42.6}$ known plaintexts, then the memory and time complexities can be reduced to $2^{42.6}$. As the key K_1 can be found with about 2^{43} known plaintexts using linear cryptanalysis [22], this allows recovering the full key (K_0, K_1, K_2) of Addition DESX with data complexity of about 2^{43} known plaintexts and time and memory complexities of 2^{43} in total.

8.2. *Attack on DES Surrounded by Decorrelation Modules*

Decorrelation modules, introduced by Vaudenay [31] in 1997, are tools to ensure security against statistical attacks such as differential and linear cryptanalysis. One of the basic decorrelation modules (used in COCONUT98 [30]) is the *NUT-II* decorrelation module defined as $DM_{K_1,K_2}(X) = (X \oplus K_1) \cdot K_2$, where the multiplication is done over the field $GF(2^n)$, and $K_2 \neq 0$.

One property of this decorrelation module is that once the key is set, the decorrelation module is linear, but when the key is random, the probability of any non-trivial differential going through the module equals $1/(2^n - 1)$ on average. A similar condition can be proved with respect to linear cryptanalysis as well. Thus, inserting decorrelation modules as an element in a block cipher is suggested in order to make it secure against differential and linear cryptanalysis.

It seems that surrounding a block cipher with key-dependent decorrelation modules is a stronger measure than adding pre/post key whitening.[13] However, it turns out that in the case of DES, due to the complementation property, this leads to related-key attacks which are significantly stronger than the best known attacks on DESX in the related-key model.

Consider the block cipher *Decorrelation-DES*, defined as

$$E_{(K_0,K_1),K_2,(K_3,K_4)}(P) = M_1\bigl(DES_{K_2}\bigl(M_0(P)\bigr)\bigr),$$

where $M_0(X) = (X \oplus K_0) \cdot K_1$, $M_1(X) = (X \oplus K_3) \cdot K_4$, and $K_1 \neq 0$, $K_4 \neq 0$.

Consider two plaintext/ciphertext pairs (P, C) and (P^*, C^*), encrypted under the keys $(K_0, K_1, K_2, K_3, K_4)$ and $(K_0, K_1, \overline{K_2}, K_3, K_4)$, respectively. Assume that the plaintext pair (P, P^*) satisfies

$$M_0(P) \oplus M_0\bigl(P^*\bigr) = \bigl(P \oplus P^*\bigr) \cdot K_1 = FF \dots FF_x.$$

[13] It should be emphasized that while surrounding a cipher with decorrelation modules seems a reasonable way to strengthen the cipher with respect to differential and linear cryptanalysis, this specific way was not suggested by Vaudenay in [30]. Our attack does not apply when the module is inserted *in the middle* of the cipher, as proposed in [30].

Then, by the complementation property of DES, we have

$$DES_{K_2}\big(M_0(P)\big) \oplus DES_{\overline{K_2}}\big(M_0\big(P^*\big)\big) = FF \ldots FF_x.$$

Since for a fixed key, the decorrelation module M_1 is linear, this implies

$$C \oplus C^* = M_1\big(DES_{K_2}\big(M_0(P)\big)\big) \oplus M_1\big(DES_{\overline{K_2}}\big(M_0\big(P^*\big)\big)\big) = FF \ldots FF_x \cdot K_4. \quad (16)$$

As the right-hand side of Eq. (16) does not depend on the plaintexts, one can mount an attack similar to the attack on Addition DESX presented in Sect. 8.1, with the pair $(P \oplus a, P^* \oplus a)$ considered instead of the pair $(P + a, P^* - a)$. The data and time complexities of the attack are exactly the same as the complexities of the attack on Addition DESX (including its known plaintext variant), and the attack allows recovering the subkeys K_1 and K_4.

Note that after recovering these subkeys, the cipher is equivalent (up to pre/post multiplication by known constants) to

$$E_{K_0', K_2, K_3'}(P) = DES_{K_2}\big(P \oplus K_0'\big) \oplus K_3',$$

that is, to DESX![14] Hence, our attack shows that *with respect to the related-key model*, surrounding DES by decorrelation modules may be weaker than adding pre/post key whitening, since it does not increase the security and on the other hand, it allows the adversary to retrieve part of the secret key efficiently.[15]

9. Open Problems

We conclude this paper with a few open problems and topics for further research that arise naturally from our results.

Memoryless Attacks on EM with $D \ll 2^{n/2}$ In Sect. 6, we showed that a lower bound on the memory complexity of attacks on EM cannot be obtained in general, by presenting a memoryless attack with $D = T = 2^{n/2}$. However, our attack is based on the slide-with-a-twist technique, which is applicable only for $D \geq 2^{n/2}$. What can be said about the case $D \ll 2^{n/2}$?

At first glance, it seems that we can obtain a memoryless attack by adapting the slidex attack described in Sect. 3.2, in the same way like the slide-with-a-twist attack is adapted to the memoryless scenario in Sect. 6. However, it appears that such an adaptation does not work. The main obstacle is that the adversary has to reuse the data many times in order to construct the hash tables for different values of Δ, and this can be done only if the data is stored somewhere rather than used in an on-line manner which discards it after computing the next plaintext. This leads to the following open problem:

[14] Note that, actually, DESX is a special case of Decorrelation-DES, in which $K_1 = K_4 = 1$. Our attack is not effective against DESX since it allows only recovering the subkeys K_1 and K_4 which are known in the case of DESX to be equal to 1.

[15] We stress that our conclusion applies only to the related-key model, and not to a general comparison between the security of Decorrelation-DES and of DESX.

Problem 1. Does there exist a memoryless attack on the Even–Mansour scheme with D E-oracle queries and $2^n/D$ $\mathcal{F}$-oracle queries, where $D \ll 2^{n/2}$?

A similar question can be asked with respect to the Single-Key Even–Mansour scheme, where in addition to the slidex attack, the simple attack presented in Sect. 4.2 can also break the scheme when $D \ll 2^{n/2}$. The attack of Sect. 4.2 can also be transformed to a memoryless attack, by defining a random function

$$\mathcal{H}(X) = \begin{cases} X \oplus E(X), & LSB(X) = 1, \\ X \oplus \mathcal{F}(X), & LSB(X) = 0, \end{cases}$$

and using Floyd's cycle finding algorithm to find a collision of $\mathcal{H}$. In the case when D and T are both close to $2^{n/2}$, with a constant probability such collision yields a pair (X_1, X_2) such that $X_1 \oplus E(X_1) = X_2 \oplus \mathcal{F}(X_2)$, concluding the attack. The problem is that if $D \ll 2^{n/2}$, then with overwhelming probability, a collision in $\mathcal{H}$ is of the form $X_1 \oplus \mathcal{F}(X_1) = X_2 \oplus \mathcal{F}(X_2)$, which is not useful to the adversary. Therefore, we state an additional open problem:

Problem 2. Does there exist a memoryless attack on the Single-Key Even–Mansour scheme with D E-oracle queries and $2^n/D$ $\mathcal{F}$-oracle queries, where $D \ll 2^{n/2}$?

If such a memoryless attack can be found only for Single-Key EM and not for the ordinary EM, this will show that at least in some respect, the use of an additional key in EM does make the scheme stronger.

Multi-Round Even–Mansour Construction The standard security notion for block ciphers (that was used, e.g., in the AES competition) states that a block cipher provides n-bit security if any attack that can break it in the single-user setting requires at least 2^n data, time, or memory. An n-bit key block cipher is considered secure if it provides n-bit security.

According to this notion, the Even–Mansour construction is not secure, as it has a $2n$-bit key and provides only $n/2$-bit security (due to the attacks with data and time complexity of $2^{n/2}$). Single-key EM is better in this sense, as it provides $n/2$-bit security while using an n-bit key, but is still far from security level equal to the key length.

A natural way to increase the security of EM while preserving its general structure is to use several consecutive rounds of EM, that is,

$$EM_r(X) = K \oplus P_r\big(K \oplus P_{r-1}\big(K \oplus \big(\cdots\big(K \oplus P_1(K \oplus X)\big)\big)\big)\big),$$

where K is the secret key, and $P_1, \ldots, P_r$ are publicly known permutations.

This extension was studied in several recent papers (e.g., [7,20,29]) with respect to its security in the indistinguishability model, and served as the basis to the design of several block ciphers, such as LED and Zorro.

In [24], Nicolic et al. presented an attack on EM_2 (i.e., two-round EM) with data, memory, and time complexities of roughly $2^{n-\log n}$. Recently, Dinur et al. [11] enhanced the attack to target EM_3 with similar data, memory, and time complexities.

Based on these works, we state an additional open problem:

Problem 3. What is the minimal number of rounds r such that any attack on r-round EM requires at least 2^n data, time or memory?

Other Applications of the Slidex Technique In this paper, we presented three new slide-type attacks: the slidex attack, the mirror slidex attack, and the addition slidex attack. We applied them to the Even–Mansour construction and to variants of several block ciphers, such as GOST and DESX. We believe that the generic form of the techniques can make them applicable to other block ciphers as well. Hence, we conclude the paper with a quest:

Problem 4. Find other applications of the new slide-type techniques proposed in this paper.

References

[1] E. Andreeva, A. Bogdanov, Y. Dodis, B. Mennink, J.P. Steinberger, On the indifferentiability of key-alternating ciphers. IACR Cryptology ePrint Archive **61** (2013). Accepted to CRYPTO 2013. doi:10.1007/978-3-642-40041-4-29

[2] D.J. Bernstein, The Salsa20 family of stream ciphers, in *The eSTREAM Finalists*, ed. by M.J.B. Robshaw, O. Billet. Lecture Notes in Computer Science, vol. 4986 (Springer, Berlin, 2008), pp. 84–97

[3] E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard* (Springer, Berlin, 1993)

[4] E. Biham, O. Dunkelman, N. Keller, Improved slide attacks, in *FSE*, ed. by A. Biryukov. Lecture Notes in Computer Science, vol. 4593 (Springer, Berlin, 2007), pp. 153–166

[5] A. Biryukov, D. Wagner, Slide attacks, in *FSE*, ed. by L.R. Knudsen. Lecture Notes in Computer Science, vol. 1636 (Springer, Berlin, 1999), pp. 245–259

[6] A. Biryukov, D. Wagner, Advanced slide attacks, in *EUROCRYPT*, ed. by B. Preneel. Lecture Notes in Computer Science, vol. 1807 (Springer, Berlin, 2000), pp. 589–606

[7] A. Bogdanov, L.R. Knudsen, G. Leander, F.X. Standaert, J.P. Steinberger, E. Tischhauser, Key-alternating ciphers in a provable setting: encryption using a small number of public permutations (extended abstract), in Pointcheval and Johansson, [26], pp. 45–62

[8] S. Chen, J.P. Steinberger, Tight security bounds for key-alternating ciphers. IACR Cryptology ePrint Archive **222** (2013)

[9] J. Daemen, Limitations of the Even–Mansour construction, in Imai et al. [17], pp. 495–498

[10] I. Dinur, O. Dunkelman, A. Shamir, Improved attacks on full GOST, in *FSE*, ed. by A. Canteaut. Lecture Notes in Computer Science, vol. 7549 (Springer, Berlin, 2012), pp. 9–28

[11] I. Dinur, O. Dunkelman, N. Keller, A. Shamir, Key recovery attacks on 3-round Even–Mansour, 8-step LED-128, and full AES^2. IACR Cryptology ePrint Archive 391 (2013). Accepted to ASIACRYPT 2013

[12] O. Dunkelman, N. Keller, A. Shamir, Minimalism in cryptography: the Even–Mansour scheme revisited, in Pointcheval and Johansson [26], pp. 336–354

[13] S. Even, Y. Mansour, A construction of a cipher from a single pseudorandom permutation. *J. Cryptol.* **10**(3), 151–162 (1997)

[14] S. Even, Y. Mansour, A construction of a cipher from a single pseudorandom permutation, in Imai et al. [17], pp. 210–224

[15] R.W. Floyd, Nondeterministic algorithms. *J. ACM* **14**(4), 636–644 (1967)

[16] J. Guo, T. Peyrin, A. Poschmann, M.J.B. Robshaw, The LED block cipher, in *CHES*, ed. by B. Preneel, T. Takagi. Lecture Notes in Computer Science, vol. 6917 (Springer, Berlin, 2011), pp. 326–341

[17] H. Imai, R.L. Rivest, T. Matsumoto (eds.), *Advances in Cryptology—ASIACRYPT '91, Proceedings of International Conference on the Theory and Applications of Cryptology*, Fujiyoshida, Japan, November 11–14, 1991. Lecture Notes in Computer Science, vol. 739 (Springer, Berlin, 1993)

[18] J. Kilian, P. Rogaway, How to protect DES against exhaustive key search (an analysis of DESX). *J. Cryptol.* **14**(1), 17–35 (2001)

[19] K. Kurosawa, Power of a public random permutation and its application to authenticated encryption. *IEEE Trans. Inf. Theory* **56**(10), 5366–5374 (2010)

[20] R. Lampe, Y. Seurin, How to construct an ideal cipher from a small set of public permutations. Cryptology ePrint Archive, Report 2013/255. http://eprint.iacr.org/ (2013)

[21] R. Lampe, J. Patarin, Y. Seurin, An asymptotically tight security analysis of the iterated Even–Mansour cipher, in Wang and Sako [33], pp. 278–295

[22] M. Matsui, The first experimental cryptanalysis of the data encryption standard, in *CRYPTO*, ed. by Y. Desmedt. Lecture Notes in Computer Science, vol. 839 (Springer, Berlin, 1994), pp. 1–11

[23] F. Mendel, V. Rijmen, D. Toz, K. Varici, Differential analysis of the LED block cipher, in Wang and Sako [33], pp. 190–207

[24] I. Nikolić, L. Wang, S. Wu, Cryptanalysis of round-reduced LED, in *FSE*. Lecture Notes in Computer Science (2013, to appear)

[25] G. Nivasch, Cycle detection using a stack. *Inf. Process. Lett.* **90**(3), 135–140 (2004)

[26] D. Pointcheval, T. Johansson (eds.), *Advances in Cryptology—EUROCRYPT 2012—Proceedings 31st of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Cambridge, UK, April 15–19, 2012. Lecture Notes in Computer Science, vol. 7237 (Springer, Berlin, 2012)

[27] R.L. Rivest, DESX. Never published (1984)

[28] Russian National Bureau of Standards, Federal information processing standard-cryptographic protection—cryptographic algorithm. GOST 28147-89, 1989

[29] J.P. Steinberger, Improved security bounds for key-alternating ciphers via Hellinger distance. IACR Cryptology ePrint Archive 481 (2012)

[30] S. Vaudenay, Provable security for block ciphers by decorrelation, in *STACS*, ed. by M. Morvan, C. Meinel, D. Krob. Lecture Notes in Computer Science, vol. 1373 (Springer, Berlin, 1998), pp. 249–275

[31] S. Vaudenay, Decorrelation: a theory for block cipher security. *J. Cryptol.* **16**(4), 249–286 (2003)

[32] D. Wagner, A generalized birthday problem, in *CRYPTO*, ed. by M. Yung. Lecture Notes in Computer Science, vol. 2442 (Springer, Berlin, 2002), pp. 288–303

[33] X. Wang, K. Sako (eds.), *Advances in Cryptology—ASIACRYPT 2012—Proceedings of 18th International Conference on the Theory and Application of Cryptology and Information Security*, Beijing, China, December 2–6, 2012. Lecture Notes in Computer Science, vol. 7658 (Springer, Berlin, 2012)