Cryptanalysis of Round-Reduced LED

Ivica Nikolić(✉), Lei Wang, and Shuang Wu

Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore, Singapore
{inikolic,wang.lei,wushuang}@ntu.edu.sg

Abstract. In this paper we present known-plaintext single-key and chosen-key attacks on round-reduced LED-64 and LED-128. We show that with an application of the recently proposed *slidex* attacks [5], one immediately improves the complexity of the previous single-key 4-step attack on LED-128. Further, we explore the possibility of multicollisions and show single-key attacks on 6 steps of LED-128. A generalization of our multicollision attack leads to the statement that no 6-round cipher with two subkeys that alternate, or 2-round cipher with linearly dependent subkeys, is secure in the single-key model. Next, we exploit the possibility of finding pairs of inputs that follow a certain differential rather than a differential characteristic, and obtain chosen-key differential distinguishers for 5-step LED-64, as well as 8-step and 9-step LED-128. We provide examples of inputs that follow the 8-step differential, i.e. we are able to practically confirm our results on 2/3 of the steps of LED-128. We introduce a new type of chosen-key differential distinguisher, called *random-difference* distinguisher, and successfully penetrate 10 of the total 12 steps of LED-128. We show that this type of attack is generic in the chosen-key model, and can be applied to any 10-round cipher with two alternating subkeys.

Keywords: LED · Lightweight · Multicollision · Single-key attack · Chosen-key attack

1 Introduction

The lightweight block cipher LED was proposed by Guo *et al.* at CHES 2011 [10]. It is a hardware optimized 64-bit cipher, with two main instances LED-64 for 64-bit key support, and LED-128 for 128-bit keys. Based on the AES design, LED uses modified, hardware-friendly operations and a trivial key schedule. As the authors targeted compact design, but as well secure even against related-key attacks, the number of rounds of LED is relatively large, i.e. LED-64 uses 32 rounds grouped in 8 steps of 4 rounds, while LED-128 has 48 rounds, or equivalently 12 steps. A round of LED is similar to a round of AES, with one exception: the addition of the round keys in AES is replaced with an addition of constants in LED. The subkeys are added only after every fourth round, thus

S. Moriai (Ed.): FSE 2013, LNCS 8424, pp. 112–129, 2014.
DOI: 10.1007/978-3-662-43933-3_7,

one step of LED (which consists of 4 rounds), behaves as 4 rounds of single-key AES – a construction with well analyzed differential and linear properties.

In the submission paper, the designers provide analysis of LED against various attacks – we mention the attacks in the chosen-key model: 15 rounds for LED-64 and 27 rounds for LED-128. Isobe and Shibutani [11] show single-key attacks on LED-64 reduced to 8 rounds, and LED-128 reduced to 16 rounds. Mendel *et al.* [14] give a supplementary cryptanalysis in different single and related-key models for both versions of the cipher. They are able to penetrate 16 rounds in the related-key model for LED-64, and 24 rounds for LED-128, with an additional single-key attack on 16 rounds of LED-128. An independent work proposed by Bodganov *et al.* in [2] also introduced similar related-key attacks on the generic structure of two-round SEM [5] with three identical keys.

We start our analysis with a brief overview of the previous results on the scheme used in LED as well as of the techniques applied in the attacks on LED (Sect. 2). The overview would help us to clearly describe our attacks in the single-key model (Sect. 3), and in the chosen-key model (Sect. 4). Our first result is an improvement of the single-key attack on 16-round LED-128 presented in [14]. We show that instead of using Daemen's attack [4] as a preliminary step, one can use the recently proposed slidex attack [5], and end up with an immediate twofold gain in terms of the data requirements: the attack from a chosen plaintext as in [14] becomes a known plaintext, while the data complexity from the whole codebook drops to 2^d, where d can be any value chosen by the attacker. Next, by exploiting the idea of multicollisions, we show a *single-key* attack on 24 rounds of LED-128. We eliminate one of the subkeys by guessing, and then we are able to attack the remaining construction by creating a set of multicollisions which allows to find the second subkey. It is important to note that our technique is applicable to LED for any step function, that is the number of rounds we can attack depends strictly on the number of used subkeys. Moreover, using the same approach one can mount attacks on any two-round construction with three equal (or linearly dependent) subkeys, e.g. SEM [5] with an additional round. The idea of using differentials instead of differential characteristic is examined in our chosen-key attacks on 20-round LED-64, and 32-,36-round LED-128. We show that two consecutive active steps in a differential path, can be threated as a differential. This leads to a significant reduction of the complexity for finding a pair that follows the path. We are able with a complexity of around 2^{32} encryptions to construct a pair that follows our defined path, and give an example of such pair found on a computer for 32 rounds of LED-128, i.e. we can show a practical chosen-key distinguisher for 2/3 of the cipher rounds. We propose a new type of chosen-key distinguishers, called *random-difference distinguishers*, where the attacker is supposed to find a pair of inputs that follow a certain differential, for any input difference. We show that LED-128 is vulnerable to this type of distinguishers for 40 rounds out of the total 48 rounds, i.e. 5/6 of the rounds of LED-128 can be distinguished in the chosen-key model. Furthermore, we show that this distinguisher is generic to all 10-round/step ciphers with two subkeys that alternate. An overview of the results on LED is given in Table 1.

Table 1. Attacks on LED

Cipher	Framework	Type	Steps	Time	Data	Memory	Ideal	Source
LED-64	single-key	Key recovery	2	2^{56}	2^{8}	2^{11}	2^{64}	[11]
(8 steps)	chosen-key	Distinguiher	3.75	2^{16}	–	2^{16}	2^{32}	[10]
	related-key‡	Key recovery	4	$2^{62.7}$	$2^{62.7}$	$2^{62.7}$	2^{64}	[14]
	chosen-key	Distinguisher	4	$2^{33.5}$	–	2^{32}	$2^{41.4}$	4.1
	chosen-key	Distinguisher	5	$2^{60.2}$	–	$2^{61.5}$	$2^{66.1}$	4.1
LED-128	single-key	Key recovery	4	2^{112}	2^{16}	2^{19}	2^{128}	[11]
(12 steps)	single-key	Key recovery	4	2^{96}	2^{64}	2^{32}	2^{128}	[14]
	single-key	Key recovery	4	2^{96}	2^{32}	2^{32}	2^{128}	3.1
	related-key	Key recovery	6	2^{96}	2^{64}	2^{32}	2^{128}	[14]
	single-key	Key recovery	6	$2^{124.4}$	2^{59}	2^{59}	2^{128}	3.2
	chosen-key	Distinguisher	6.75	2^{16}	–	2^{16}	2^{32}	[10]
	chosen-key	Distinguisher	8	$2^{33.5}$	–	2^{32}	$2^{41.4}$	4.2
	chosen-key	Distinguisher	9	$2^{60.8}$	–	2^{62}	$2^{66.1}$	4.2
	chosen-key	Distinguisher	10	$2^{60.3}$	–	2^{60}	2^{64}	4.3

‡: Complexity is based on the 6 found pairs that follow the iterative characteristic.

2 Specification and Related Works

In this section we give a brief description of LED and present related analysis relevant for understanding our attacks.

2.1 The Block Cipher LED [10]

LED uses a block size of 64 bits and a key size ranging from 64 bits to 128 bits. The two primary instances, LED-64 and LED-128, use a 64-bit key and an 128-bit key, respectively.

The key schedule is trivial and very efficient: LED-64 uses the 64-bit secret key in each step as a subkey, while LED-128 divides the 128-bit secret key K into halves $K_0||K_1$ and uses K_0 and K_1 alternatively as the subkeys, i.e. K_0 is used in the even steps, while K_1 is used in the odd steps. LED follows the standard iterative cipher structure and produces a ciphertext C from the plaintext P in t iterations of a so-called step function F_i (see Fig. 1):

$$\begin{aligned} S_0 &\longleftarrow P \\ S_{i+1} &\longleftarrow F_i(S_i \oplus K_i), 0 \leq i \leq t-1 \\ C &\longleftarrow S_t \oplus K_t \end{aligned}$$

In LED-64 the number of steps t is 8, while in the other instances including LED-128, t is defined as 12. The step function F_i is a 4-round AES-like permutation where the addition of the subkeys is replaced with an addition of constants. Thus, all the step functions F_i can be seen as public permutations and differ only in the round constants they use. Since most of our attacks can be mounted independently of the specification of the step functions, we omit their description and refer the interested reader to [9,10] for a full specification.

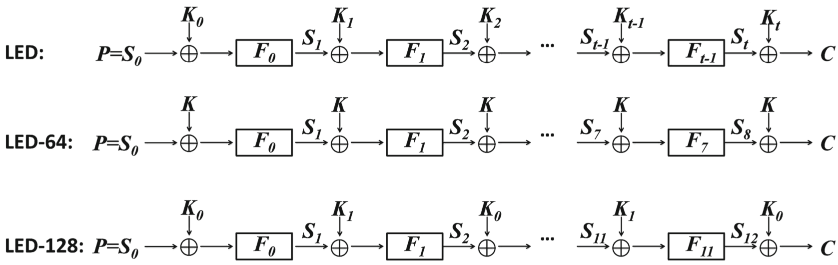

Fig. 1. LED and its two primary instances LED-64 and LED-128

2.2 Related Attacks on the Even-Mansour Scheme

The Even-Mansour scheme [6] uses two secret keys (K_0, K_1) and a public permutation F to construct a cipher $\mathrm{EM}_{K_0,K_1}(P) = F(P \oplus K_0) \oplus K_1$ (see Fig. 2). This scheme is very attractive due to its extremely simple design with a provable security margin. Several papers on cryptanalysis of Even-Mansour have been published. This section briefly describes the attacks relevant to our paper.

Daemen's Attack [4]**.** The chosen-plaintext attack of Daemen can be sketched as:

1. Choose a non-zero difference Δ.
2. Choose 2^d different random values as plaintexts P, query P and $P \oplus \Delta$ to the Even-Mansour scheme to receive the corresponding ciphertexts C and C' respectively, and compute and store $\Delta C = C \oplus C'$.
3. Choose a random value X, compute $\Delta F(X) = F(X) \oplus F(X \oplus \Delta)$, and check if $\Delta F(X)$ is among the stored ΔC computed at step 2. If a match is found, then compute $K_0 = P \oplus X$ and $K_1 = F(X) \oplus C$ and confirm on another pair of plaintext-ciphertext that the values are correct.

After repeating the step 3 around 2^{n-d} times, where n is the block size, the secret keys are expected to be recovered. Thus the overall complexity is 2^d chosen plaintexts and 2^{n-d} encryptions.

Slidex Attack [5]**.** Dunkelman *et al.* were able to match the complexity of Daemen's attack with only known-plaintexts, using a so-called *slidex attack*.

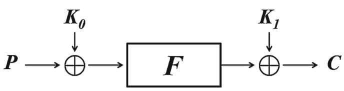

Fig. 2. Even-Mansour scheme

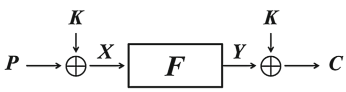

Fig. 3. Single-key Even-Mansour scheme (SEM)

Let us assume the attacker obtains 2^d known plaintext-ciphertext pairs (P_i, C_i). Then the slidex attack can be described as:

1. Choose a random non-zero difference Δ.
2. For all (P_i, C_i) compute a set of $F(P_i \oplus \Delta) \oplus C_i$ and look for a collision in the set.
3. If a collision is found, e.g. $F(P \oplus \Delta) \oplus C = F(P' \oplus \Delta) \oplus C'$, then $K_0 = P \oplus P' \oplus \Delta$.
4. Otherwise, go to step 1.

After repeating the steps $1 - 4$ around 2^{n-2d} times, the correct value of K_0 is expected to be recovered. With the knowledge of K_0, the value of K_1 can be trivially recovered using a single known pair (P, C). Thus the overall complexity is 2^d known plaintexts and 2^{n-d} encryptions.

An Attack on SEM [5]**.** Dunkelman *et al.* proposed a single-key variant of the Even-Mansour scheme depicted in Fig. 3, which uses the same secret key as both the pre- and the post-whitening keys, i.e. $F(P \oplus K) \oplus K$. Following the notation from [5], we refer to this single-key variant as SEM. Dunkelman *et al.* provided once more a known-plaintext attack on SEM based on the observation that $P \oplus C = X \oplus Y$. Again, we assume the attacker obtains 2^d known plaintext-cihertext pairs (P_i, C_i). The steps of the attack are as follows:

1. Compute a set of $P_i \oplus C_i$ for all 2^d (P_i, C_i).
2. Choose a random value of X, compute $Y = F(X)$ and match $X \oplus Y$ to the values of $P \oplus C$ from the set computed at step 1.
3. If a match is found, $K = P \oplus X$.
4. Otherwise, go to to step 2.

After repeating the steps $2 - 4$ around 2^{n-d} times, the correct value of K is expected to be recovered. Thus the complexity is 2^d known plaintexts and $2^d + 2^{n-d}$ computations.

2.3 Key-Recovery Attacks on LED

Several chosen-plaintext key-recovery attacks on LED have been published. This section briefly describes the attacks related to this paper.

Three-Subset Meet-in-the-Middle Attacks on LED [11]**.** Isobe and Shibutani applied the attack framework formalized by Bogdanov and Rechberger [3] to LED in a very original and non-trivial manner [11] and presented chosen-plaintext attacks on 2-step LED-64 and 4-step LED-128. Their complexity on 4-step LED-128 is 2^{16} chosen plaintexts and 2^{112} encryptions. We stress that the time complexity of their attacks cannot be reduced when more data is available.

Guess-and-Recover Attacks on LED-128 **[14].** Mendel *et al.* published key-recovery attacks on 4-step and 6-step LED-128 in the single-key and the related-key settings, respectively. The main strategy of their attacks is first to guess the value of K_0 in order to peel off the first and the last step functions, and then to efficiently recover the value of K_1 by attacking the shortened cipher. In this paper we call such attack strategy *guess-and-recover*. The attack on 4-step LED-128 (depicted in Fig. 4) starts by guessing the key K_0, thus the 4-step LED-128 is shortened to a cipher E, and moreover G (in Fig. 4) becomes now a public permutation. As E follows the Even-Mansour scheme, Mendel *et al.* adopted Daemen's attack [4] sketched in Sect. 2.2 to recover the key K_1. In particular, for an input S_1 to the cipher E, in order to get the value of $E(S_1)$, the attacker computes $P = F_0^{-1}(S_1) \oplus K_0$, then queries P to LED-128 to receive the corresponding ciphertext C, and finally computes $F_3^{-1}(C \oplus K_0)$ as $E(S_1)$. Note, Mendel *et al.*'s attack is a known/chosen-plaintext attack and since the Daemen's attack procedure is executed for each guess of K_0 (thus repeated 2^{64} times), the data complexity of the attack equals the entire codebook while the time complexity is 2^{96} encryptions. The authors point out that the attacker is able to reduce the data complexity below the entire codebook, however then he has to sacrifice the time complexity, i.e. the time will increase proportionally. We stress that the attack becomes a chosen-plaintext attack if the data complexity is less than the entire codebook, otherwise it can be considered known-plaintext attack (it requires the whole codebook, hence there is no difference between chosen and known plaintext).

Mendel *et al.* were able to extended the above attack on 4 steps to 6 steps of LED-128 in the related key settings. A pictorial view of the guess-and-recover strategy on 6-step LED-128 is given in Fig. 5. The attack uses a related key $K' = K_0||K_1'$, where K_1' is $K_1 \oplus \Delta$. Let E' be the shortened cipher under the related key K'. For a random value S_1, inside the computations of $E(S_1)$ and $E'(S_1 \oplus \Delta)$, the difference $\Delta G_1(S_1) = G_1(S_1 \oplus K_1) \oplus G_1(S_1 \oplus \Delta \oplus K_1')$ is always 0. Hence the input difference of G_2 is always Δ. Thus Daemen's attack can be applied to recover the value of K_1 in a straightforward way with the same data and time complexity.

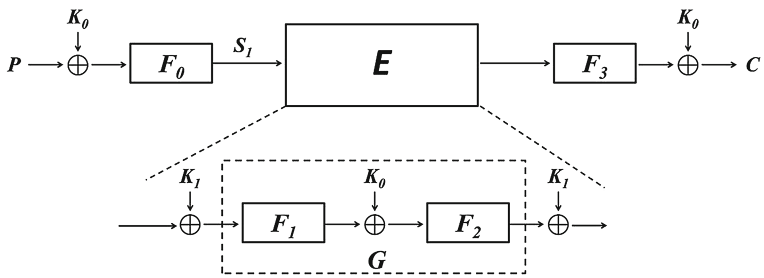

Fig. 4. Guess-and-recover strategy on 4-step LED-128

Attacks on LED-64 Exploiting Differential Characteristics for the Step Functions [14]. Mendel *et al.* proposed as well attacks on 3-step and 4-step LED-64 in the related-key setting, by investigating the differential properties of the step functions of LED, in particular differential characteristics with high height as well as iterative differential characteristics. For the public permutations used in the step function, the authors found differential characteristics with a probability of around 2^{-54}, while theoretically it may go up to 2^{-50} (25 active Sboxes and each with 2^{-2}). In one part of our analysis, we use the results of [14], and in order to provide conservative results, we assume the optimal differential characteristic for the step functions to hold with probability 2^{-54}. However, as pointed out by Mendel *et al.*, differential characteristics with a better probability may exist and if such characteristic is found, our attack complexity will be immediately improved.

2.4 Differential Multicollisions for Block Ciphers [1]

This concept was introduced by Biryukov *et al.* [1]. It can be defined as follows:

Definition 1. *A differential q-multicollision for the block cipher $E_K(\cdot)$ is defined as a set of two differences ΔP and ΔK and q key-plaintext pairs (K_1, P_1), (K_2, P_2), ..., (K_q, P_q) that satisfy the relation:*

$$\begin{aligned}&E_{K_1}(P_1) \oplus E_{K_1 \oplus \Delta K}(P_1 \oplus \Delta P) = \\ &E_{K_1}(P_2) \oplus E_{K_2 \oplus \Delta K}(P_2 \oplus \Delta P) = \\ &\cdots = \\ &E_{K_q}(P_q) \oplus E_{K_q \oplus \Delta K}(P_q \oplus \Delta P),\end{aligned}$$

Biryukov *et al.* have proven that it takes at least $q \cdot 2^{\frac{q-2}{q+2}n}$ queries to produce a differential q-multicollision for an *ideal* n-bit block cipher. Thus if an attacker can find a differential q-multicollision on a dedicated block cipher with a complexity less than the lower bound $q \cdot 2^{\frac{q-2}{q+2}n}$, he can distinguish the cipher from ideal in the chosen-key model.

3 Key-Recovery Attacks on LED-128 in the Single-Key Setting

In this section we present key recovery attacks on 4 steps and 6 steps of LED-128 in the single-key framework. The attacks are independent of the definition of the step function, and the data is always known-plaintext.

3.1 Attack on 4 Steps

We can improve the previous key-recovery attacks on 4-step LED-128 in a relatively straightforward way. Our attack follows the guess-and-recover strategy, which is depicted in Fig. 4. First, note that the shortened cipher E is the SEM scheme. Thus after guessing the value K_0, to recover K_1 instead of adopting Daemen's approach [4] as in the previous attack [14], we apply Dunkelman *et al.*'s slidex attack or their attack approach on SEM [5] sketched in Sect. 2.2. This immediately gives us the first advantage: our attack is a known-plaintext attack. Moreover, based on the complexity evaluation given below, our approach has a second advantage: the complexity also gets improved. Since we will extend the below approach to attack 6-step `LED-128` in Sect. 3.2, here we give a detailed description of the complete attack approach. The notations below follow the one from Fig. 4.

Attack Procedure. Suppose the attacker obtains 2^d known plaintext-ciphertext pairs (P, C).

1. Guess the value of K_0.
2. For all 2^d pairs (P, C), compute $S_1 = F_0(K_0 \oplus P)$ and $E(S_1) = F_3^{-1}(K_0 \oplus C)$, then compute $S_1 \oplus E(S_1)$, and store the pairs $(S_1, S_1 \oplus E(S_1))$.
3. Choose 2^{64-d} different random values denoted as X. For each X:
   (a) Compute $G(X) \oplus X$ and match it to $S_1 \oplus E(S_1)$ stored at step 2.
   (b) If a match is found, compute the value $S_1 \oplus X$ as a candidate of K_1. Otherwise, go to step 3(a) with the next value of X.
   (c) Verify the correctness of the candidate for K_1 by using another $(S_1', E(S_1'))$, where S_1' is not equal to S_1. In particular, compute the value for $E(S_1')$ using the current guessed K_0 and the candidate K_1, and check whether it is equal to the value for $E(S_1')$ computed at Step 2. If they are equal, output the currently guessed K_0 and the candidate K_1 as the real key, and terminate the procedure. Otherwise, go to step 3(a) with the next value of X.
4. Change the value of K_0, and repeat steps $1-3$ until all possible values of K_0 are tested.

Complexity. The unit is one computation of the whole 4-step `LED-128` consisting of four step functions. The steps $1-3$ are repeated 2^{64} times. One execution of step 2 requires $2^d \times \frac{2}{4} = 2^{d-1}$ computations. In one execution of step 3, step 3(a) is repeated 2^{64-d} times, and therefore the total complexity is $2^{64-d} \times \frac{2}{4} = 2^{63-d}$. At step 3(b), on average there is one match among all the 2^{64-d} repetitions. Hence the complexity of steps 3(b) and 3(c) is 1. Thus the overall time complexity is $2^{64} \cdot (2^{d-1} + 2^{63-d} + 1) \approx 2^{63+d} + 2^{127-d}$, while the data complexity is 2^d known plaintext-ciphertext pairs, and 2^d memory required in step 2.

Success Probability. When the guessed value of K_0 is correct, if one random X at step 3 collides with $S_1 \oplus K_1$ for some S_1 computed at step 2, the value of K_1 will be correctly recovered. The probability of a such collision is $1 - \frac{1}{e} \approx 0.63$.

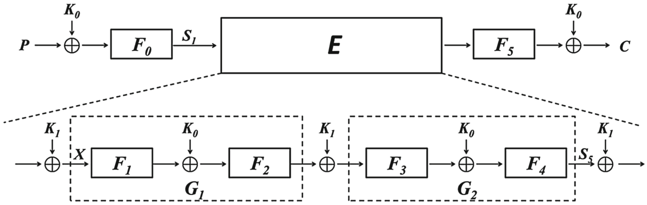

Fig. 5. Guess-and-recover attack on 6-step LED-128

Comparison to Previous Attacks. The optimal time complexity of our attack is 2^{96} by setting d to 32, while the data complexity is 2^{32} known plaintexts. Previous attacks either cannot reach such low time complexity (e.g. [11]) or with a much higher data complexity, i.e. the entire codebook, for the same time complexity (e.g. [14]).

3.2 Attack on 6 Steps

We can extend the above attack to 6-step LED-128 by using multicollisions. As depicted in Fig. 5, the shortened cipher E after guessing K_0 can be regarded as a two-step SEM. The relation $S_1 \oplus E(S_1) = X \oplus S_5$ holds. Suppose we have a q-multicollision on $E(S_1) \oplus S_1$. Namely, we find q values $S_1^{(1)}$, $S_1^{(2)}$, ..., $S_1^{(q)}$ such that $E(S_1^{(1)}) \oplus S_1^{(1)} = E(S_1^{(2)}) \oplus S_1^{(2)} = \cdots = E(S_1^{(q)}) \oplus S_1^{(q)}$ holds. Denote the value of $E(S_1^{(i)}) \oplus S_1^{(i)}$, $1 \leq i \leq q$, by Y. Let us select a random value as X, then set S_5 as $X \oplus Y$, and compute the value $G_1(X) \oplus G_2^{-1}(S_5)$ as a candidate value of K_1, which can be verified trivially. Note that if X is equal to any of $S_1^{(i)} \oplus K_1$, $1 \leq i \leq q$, the computed candidate is the correct value of K_1. Thus after testing $2^{64}/q$ random values as X, the real value of K_1 is expected to be recovered. Recall that such attack procedure needs to be repeated for each guess of K_0, i.e. in total 2^{64} times. Hence the overall complexity is $2^{128}/q$. The details of the attack procedure are given below - for $q = 8$ the attack has the lowest complexity.

Attack Procedure. The attacker obtains 2^{59} known plaintext-ciphertext pairs (P, C).

1. Guess the value of K_0.
2. For all 2^{59} (P, C), compute $S_1 = F_0(P \oplus K_0)$ and $E(S_1) = F_5^{-1}(C \oplus K_0)$. Then compute $S_1 \oplus E(S_1)$ and store $(P, S_1, S_1 \oplus E(S_1))$.
3. Find an 8-multicollision on $S_1 \oplus E(S_1)$, namely a set of $(P^{(1)}, S_1^{(1)}, S_1^{(1)} \oplus E(S_1^{(1)}))$, ..., $(P^{(8)}, S_1^{(8)}, S_1^{(8)} \oplus E(S_1^{(8)}))$ such that $S_1^{(1)} \oplus E(S_1^{(1)}) = S_1^{(2)} \oplus E(S_1^{(2)}) = \cdots = S_1^{(8)} \oplus E(S_1^{(8)})$. Denote the value of $S_1^{(i)} \oplus E(S_1^{(i)})$, $1 \leq i \leq 8$, as Y. If no such 8-multicollision exists, go to step 1 with another guess value as K_0.

4. Choose 2^{61} random values as X. For each value of X:
   (a) Compute $X \oplus Y$ as S_5.
   (b) Compute $G_1(X) \oplus G_2^{-1}(S_5)$ denoted as Z.
   (c) Compute $X \oplus Z$, and match it to $\{S_1^{(1)}, \ldots, S_1^{(8)}\}$. If it coincides with some $S_1^{(i)}$, then Z is regarded as a candidate value of K_1. Otherwise, go to step 4(a) with the next value of X.
   (d) Verify the correctness of Z as K_1 by using another relation $(S_1, E(S_1))$ with $S_1 \neq S_1^{(i)}$. If it is correct, set $K_1 = Z$, then output the current guessed value of K_0 and K_1 as the real key, and terminate the attack procedure. Otherwise, go to step 4(a) with the next value of X.
5. Change the value of K_0, and repeat steps $1-4$ until all possible values of K_0 are tested.

Complexity. The unit is one computation of the whole 6-step LED-128. The steps $1-4$ are repeated 2^{64} times. One execution of step 2 has the complexity of $2^{59} \times \frac{2}{6} \approx 2^{57.4}$. In one execution of step 4, steps 4(a), 4(b) and 4(c) are repeated 2^{61} times, and the total complexity is $2^{61} \times \frac{4}{6} \approx 2^{60.4}$. On average, there is only one match at step 4(d) among 2^{62} random values. Thus the complexity of step 4(e) is 1. Therefore the overall time complexity is $2^{64} \cdot (2^{59.4} + 2^{60.4} + 1) \approx 2^{124.4}$, while data complexity is 2^{59} known plaintexts. The memory requirement is 2^{59} for step 2.

Success Probability. We focus on the success probability of recovering K_1, when the guessed value of K_0 is correct. First we evaluate the probability of 8-multicollisions at step 2. It has been proven that a q-multicollision among $\sqrt[q]{q!} \times 2^{\frac{q-1}{q}n}$ n-bit random values exists with a probability 0.5 [7,16]. By setting $q = 8$ and $n = 64$, $\sqrt[q]{q!} \times 2^{\frac{q-1}{q}n}$ is smaller than 2^{58}. Since we have in total 2^{59} values, the probability of an 8-multicollision is almost 1. Then we evaluate the probability of a collision between a random value X and a $S_1^{(i)} \oplus K_1$. The probability of such a collision is $1 - \frac{1}{e} \approx 0.63$. Thus the overall success probability is 0.63.

Remark. We emphasize that our attack is not related to the specification of step functions, and thus applicable to any 6-step Even-Mansour scheme with the key schedule of alternating two keys. The advantage of our attack is related to the block size n. A shown above for the case $n = 64$, q is chosen as 8, and the complexity is $2^{3.6}$ times faster than the brute-force attack. In particular, for the common block size $n = 128$, q can be 16, and our attack becomes $2^{4.6}$ times faster than the brute-force attack.

As we can see from the above analysis, the 6-step attack is actually based on a 2-step multicollision-type attack (the permutations G_1, G_2 with subkey additions), that is applicable to any permutations G_1, G_2. Thus we can derive the following interesting fact:

Observation 1. *For any two-round n-bit cipher* $E_K(P) = G_2(G_1(P \oplus K) \oplus L_1(K)) \oplus L_2(K)$*, where* G_1, G_2 *are arbitrary permutations, and* L_1, L_2 *are linear bijective functions, exists a known-plaintext attack a with time complexity of less than* 2^n *encryption queries.*

It is interesting to note that Observation 1 actually answers affirmatively the open problem proposed in [2] if there exist a single-key attack on two-round SEM structure with three identical keys and computational complexity below 2^n.

4 Chosen Key Differential Distinguishers for LED-64 and LED-128

The designers of LED pointed out in the specification document [10], that in order to gain confidence in the cipher, one should study the security of the cipher in the framework where the attacker knows or controls the key. Using the rebound [13] and Super-Sbox [8,12] techniques, they were able to penetrate 15 rounds (3.75 steps) of LED-64, and 27 rounds (6.75 steps) of LED-128. The design strategy underlying LED, in particular the trivial key schedule and fact that the best probability of a differential characteristic in an active step of LED cannot be higher than 2^{-50}, seem to confirm the findings of the designers. As LED-64 has 128-bit input (64-bit key and 64-bit state), it leads that a differential characteristic cannot have more than 2 active steps, otherwise the probability (for 3 steps) would be at most 2^{-150}, and the freedom of the 128-bit input is insufficient to satisfy the characteristic. Similarly, for LED-128, the best characteristic cannot have more than 3 active steps, as the probability of a 4-step characteristic would be at most 2^{-200}, hence the 192-bit input (128-bit key and 64-bit plaintext) is insufficient for this characteristic.

The above reasoning however, applies to the case of *differential characteristics*. Further we show that the situation changes when one investigates the effects of *differentials*. To clarify our reasoning, let us examine the case of a 2-step differential where both steps are active and assume the input and the output difference take some predefined values. The probability of a single differential characteristic that composes the differential is at most 2^{-100}. However, the probability of the differential is much higher, i.e. 2^{-64} for *any* input-output differences. Hence if we can efficiently find a pair of inputs that follow this differential, then we would spend only 64-bits of freedom, instead of 100 bits as in the case of characteristics.

The results presented in this section give solutions for finding such pairs, and use the additional freedom to penetrate more steps of LED.

4.1 Differential Multicollision on 5-Step LED-64

Our distinguisher is based on the differential path given in Fig. 6. The path is built by fixing an optimal differential characteristic in the last step function

F_4: $\Delta \rightarrow \Delta *$, which determines the value of Δ and $\Delta *$, and then the following values are set as well: $\Delta P = \Delta$, $\Delta K = \Delta$ and $\Delta C = \Delta \oplus \Delta *$. Note, the differential characteristic $\Delta \rightarrow \Delta *$ holds with a probability of at least 2^{-54}, following Mendel *et al.*'s investigation [14] described in Sect. 2.3. After the path is determined, we search for pairs (P, K) satisfying LED-64 $_K(P) \oplus$ LED-64 $_{K \oplus \Delta}(P \oplus \Delta) = \Delta \oplus \Delta *$. The search procedure starts with launching a meet-in-the-middle attack between step functions F_1 and F_2. Note that both the input difference of F_1 and the output difference of F_2 are fixed as Δ. We select random values X and Y, and independently compute $\Delta F_1(X) = F_1(X) \oplus F_1(X \oplus \Delta)$ and $\Delta F_2^{-1}(Y) = F_2^{-1}(Y) \oplus F_2^{-1}(Y \oplus \Delta)$. Then we match between $\Delta F_1(X)$ and $\Delta F_2^{-1}(Y) \oplus \Delta$. For a match, by adaptively selecting two values $F_1(X) \oplus F_2^{-1}(Y)$ and $F_1(X) \oplus F_2^{-1}(Y \oplus \Delta)$ as the key K and computing the corresponding values of P from (K, X), we obtain two pairs (K, P) which can satisfy the path on the first four step functions in Fig. 6. Finally, the differential characteristic on the last step function F_4 is satisfied probabilistically.

Attack Procedure

1. Select 2^s random values X, compute $\Delta F_1(X) = F_1(X) \oplus F_1(X \oplus \Delta)$, and store $(X, \Delta F_1(X))$. The value of s will be determined in the complexity evaluation below.
2. Select 2^s random values Y, compute $\Delta F_2^{-1}(Y) = F_2^{-1}(Y) \oplus F_2^{-1}(Y \oplus \Delta)$ and match $\Delta F_2^{-1}(Y) \oplus \Delta$ to stored ΔF_1 at step 1. On average, there are 2^{2s-64} matches.
3. For each matched pair X and Y,
   (a) Compute two values as K: $K = F_1(X) \oplus F_2^{-1}(Y)$ and $K = F_1(X \oplus \Delta) \oplus F_2^{-1}(Y \oplus \Delta)$.
   (b) Compute C and C' for each pair (K, Y) and $(K \oplus \Delta, Y \oplus \Delta)$ respectively.
   (c) If ΔC is equal to $\Delta \oplus \Delta *$, compute the corresponding value of P, and store the values of (P, K). On average, there are 2^{2s-117} values of (P, K) stored.

Complexity of Finding Differential q-Multicollision. The unit is one computation of the whole 5-step LED-64. The dominant complexity comes from steps 1 and 2, each of them requires $2^s \times \frac{1}{5}$ units, hence the total complexity is approximately $2^{s-1.3}$. To produce a differential q-multicollision, set $2^{2s-117} = q$, which implies $s = 58.5 + \log_2 \sqrt{q}$, and thus the complexity is $\sqrt{q} \cdot 2^{57.2}$. For $q = 2^6$, the overall complexity of our attack is $2^{60.2}$, while the generic attack requires at least $2^{66.1} > 2^{64}$ encryptions.

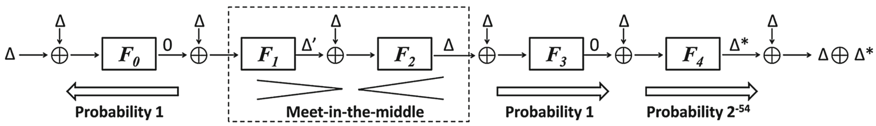

Fig. 6. Distinguisher on 5-step LED-64

Table 2. An example of pair of inputs following the 8-STEP (32 rounds) differential for LED-128. The two rows of each step denote the input and output values/differences of the steps.

	Input 1	Input 2	XOR difference
K_0	63686a8c6ed193f6	63686a8c6ed193f7	0000000000000001
K_1	0000000000000000	0000000000000000	0000000000000000
plaintext	33960e4a40a0f740	33960e4a40a0f740	0000000000000001
step 0	50fe64c62e7164b6	50fe64c62e7164b6	0000000000000000
	e82c1e07da3b4304	e82c1e07da3b4304	0000000000000000
step 1	e82c1e07da3b4304	e82c1e07da3b4304	0000000000000000
	3bb5fd710efb3bba	3bb5fd710efb3bba	0000000000000000
step 2	58dd97fd602aa84c	58dd97fd602aa84d	0000000000000001
	50fdeb1af852210e	56c051f2c88d007a	063dbae830df2174
step 3	50fdeb1af852210e	56c051f2c88d007a	063dbae830df2174
	eb82dccf19e68610	fe5507900afd76ad	15d7db5f131bf0bd
step 4	88eab643773715e6	9d3d6d1c642ce55a	15d7db5f131bf0bc
	c6dbdb083c8dfccb	b688dc44effea528	7053074cd37359e3
step 5	c6dbdb083c8dfccb	b688dc44effea528	7053074cd37359e3
	ef7e6ce5ebb78007	ef7e6ce5ebb78006	0000000000000001
step 6	8c160669856613f1	8c160669856613f1	0000000000000000
	5f2a1e2a6f01e9eb	5f2a1e2a6f01e9eb	0000000000000000
step 7	5f2a1e2a6f01e9eb	5f2a1e2a6f01e9eb	0000000000000000
	337e6d7828ea8fec	337e6d7828ea8fec	0000000000000000
ciphertext	501607f4463b1c1a	501607f4463b1c1b	0000000000000001

4.2 Differential Multicollision for 8-Step and 9-Step LED-128

Our distinguisher on 8-step LED-128 is based on a differential path given in Fig. 7, where Δ can be any non-zero value. We set $\Delta P = \Delta$, $\Delta K = (\Delta K_0 = \Delta, \Delta K_1 = 0)$ and $\Delta C = \Delta$. First we select a random value as K_1, which makes G_1 and G_2 to become two public permutations. Then we carry out a meet-in-the-middle attack between G_1 and G_2. Note both the input differences of G_1 and the output differences of G_2 are fixed as Δ. We adopt the same meet-in-the-middle procedure as the one presented in Sect. 4.1, and adaptively choose the value of K_0. As the rest of the differential path holds with probability 1, the chosen K_0 with previously fixed K_1 and P, which can be computed trivially from X, is the expected solution, namely it satisfy the whole differential path. Following the complexity evaluation as in Sect. 4.1, our attack needs $q \cdot 2^{30.5}$ computations to produce a differential q-multicollision, hence for $q = 8$, the overall complexity is $2^{33.5}$, while the generic attack needs at least $2^{41.4}$.

We would like to emphasize two aspects (freedoms) of our attack on 8 steps of LED-128: first, the difference in K_0 can be any, and second, the value of K_1 can be arbitrary as well. Even with such relaxed requirements, we are still able to find a pair that follows the differential path with a complexity of around $2^{30.5}$

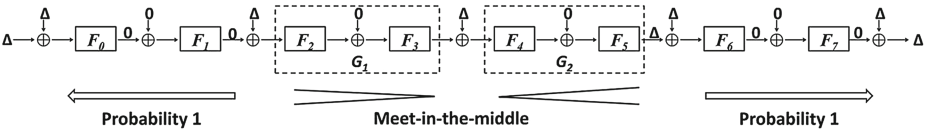

Fig. 7. Distinguisher on 8-step LED-128

8-step encryptions. An example of such pair, found on a computer, is given in Table 2. Note, in the example the difference in K_0 is 1 and the value of K_1 is 0.

Extension to 9 Steps. The above path can be extended with an additional step at the end, thus leading to a 9-step path. First, we find an optimal differential characteristic for the last step function F_9: $\Delta \rightarrow \Delta*$, i.e. we use again the same characteristic that holds with 2^{-54}. Then the differential is defined as $\Delta P = \Delta$, $\Delta K = (\Delta K_0 = \Delta, \Delta K_1 = 0)$, and $\Delta C = \Delta*$. The distinguisher uses a differential path, which is a concatenation of the path on the first 8 step functions from Fig. 7 and the characteristic $\Delta \rightarrow \Delta*$ for the last step function F_9. After selecting a random value as K_1, we apply exactly the same search procedure as in Sect. 4.1. However, this time instead of producing q pairs that follow the 8-step differential, we produce $q2^{54}$ such pairs. Obviously, after the last step, there would be around q pairs that satisfy the whole 9-step differential.

The complexity is dominated by the meet-in-the-middle attack and the generation of $q2^{54}$ pairs for the 8-step differential. To optimize the complexity, we should create $\sqrt{q}2^{59}$ differences for each G_1 and G_2, hence there would be $q2^{118}$ pairs in the middle and $q2^{118-64} = q2^{54}$ that follow the 8-step differential or q pairs for the whole 9-step differential. Thus taking into account that the G_1, G_2 take $\frac{2}{9}$ of the total number of rounds, the overall complexity for $q = 2^6$ is $2 \cdot 2^3 2^{59} \frac{2}{9} = 2^{60.8}$ encryptions of 9-step LED-128. The generic case again requires $2^{66.1}$ encryptions.

4.3 A Differential Distinguisher on 10-Step LED-128

In this section we introduce the concept of chosen-key *random-difference* distinguisher and present such distinguisher for 10 steps of LED-128.

In differential multicollisions, the attacker finds a set of two differences for the key and the plaintext, such that all the differences in the ciphertext of q pairs of keys/plaintexts, are the same. Thus the freedom is three differences: in the key, in the plaintext, and in the ciphertext, and therefore, to prove the distinguisher is not trivial, the attacker has to find many pairs of keys/plaintexts that follow the same differential. Now assume, the freedom is only in one of the input differences, and the other two depend on (or are equal to) this single difference, i.e. the attacker wants to find a key/plaintext (K, P) such that for some given difference Δ, $E_{K \oplus \Delta}(P \oplus \Delta) \oplus E_K(P) = \Delta$ holds. Obviously, if the difference Δ is random, he cannot find the input pair with a complexity lower than 2^n (see below), where n is the block size. However, one might reasonably argue, that if the attacker has to provide a single pair of key/plaintext, then he can use the additional freedom of the difference and come up with his own Δ in time complexity lower than 2^n, and thus achieve such distinguisher. Our distinguisher below thwarts such approach, since it requires the attacker to be able to build the input pair for *any random* difference Δ. This type of problem already has been analyzed in the work of Patarin [15] – he has shown that the xor of two random permutations cannot be distinguished from a pseudo-random function with less than 2^n queries. In our case, the permutations are defined as $P_1(X) = P_1(K, P, \Delta) = E_{K \oplus \Delta}(P \oplus \Delta)$ and $P_2(X) = P_2(K, P, \Delta) = E_K(P) \oplus \Delta$, i.e. they are keyed with both K and Δ, and for fixed values of these two parameters they are two distinct permutations (as long as $\Delta \neq 0$). In the chosen-key scenario discussed below, although the key can be chosen, the difference Δ is still arbitrary and unknown, hence Patarin's proof again applies to the pseudo-random function (PRF) $P_1(X) \oplus P_2(X)$, which can be translated into finding a preimage of 0 for the PRF, as from $E_{K \oplus \Delta}(P \oplus \Delta) \oplus E_K(P) = \Delta$ is follows we are looking at the condition $P_1(X) \oplus P_2(X) = 0$. The complexity of finding such preimage for an n-bit PRF is 2^n queries, and thus encryptions/decryptions. Now we are ready to give a formal definition of this non-trivial distinguisher:

Definition 2. *A random-difference distinguisher exists for the cipher $E_K(P)$, if for any randomly chosen Δ, the attacker with a complexity less than 2^n encryptions/decryptions can find a plaintext P and a key K, such that $E_K(P) \oplus E_{K \oplus \Delta}(P \oplus \Delta) = \Delta$.*

Further, we show that this type of distinguisher can be found for 10-step LED-128, i.e. we show that for a randomly chosen Δ, with less than 2^{64} queries/encryptions we can find the input P, K_0, K_1 such that $E_{K_0 \oplus \Delta || K_1}(P \oplus \Delta) \oplus E_{K_0 || K_1}(P) = \Delta$. Our analysis is based on a differential path given in Fig. 8, where the step functions denoted in a black color are active, while the white steps are non-active. In Fig. 8 we also sketch the attack procedure. We start with a meet-in-the-middle (MITM) attack between F_2 and F_3. Note that both the input difference of F_2 and the output difference of F_3 are fixed as Δ. We carry out the same MITM procedure as the one in Sect. 4.1, and find pairs (K_1, X), where X is the output value of F_3, satisfying the differential path on the first four step functions. Similarly, we perform MITM on the other side, between F_6 and F_7, and find pairs (K_1, Y) where Y is the input value of F_6, satisfying the differential path on the last four step functions. Next, we match (K_1, X) and (K_1, Y) on the value of K_1, and store (K_1, X, Y) if the value of

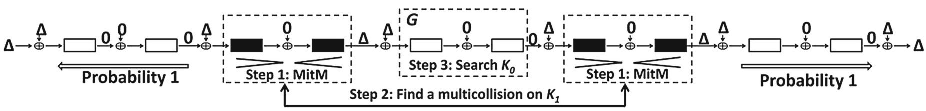

Fig. 8. Distinguisher on 10-step LED-128

K_1 is matched. Then we search for a q-multicollision among (K_1, X, Y) on the value of K_1. Namely we find a set of $(K_1^{(1)}, X^{(1)}, Y^{(1)})$, $(K_1^{(2)}, X^{(2)}, Y^{(2)})$, ..., $(K_1^{(q)}, X^{(q)}, Y^{(q)})$ with $K_1 = K_1^{(1)} = K_2^{(2)} = \cdots = K_1^{(q)}$. For this fixed K_1, G becomes a public permutation. The last step is to find a value of K_0, which links $X^{(i)}$ to $Y^{(i)}$ for some $1 \leq i \leq q$, i.e. $G(X^{(i)} \oplus K_0) \oplus K_0 = Y^{(i)}$. The search procedure is similar to the attack on SEM [5], i.e. if we have q possible values for $(X^{(i)}, Y^{(i)})$, we need only $2^n/q$ values for the inputs/outputs of G in order to find one match. A single match suggests immediately the value of K_0, hence we have fixed as well the second key K_0, and thus finding the input plaintext is trivial.

Attack Procedure. Let Δ be any non-zero value.

1. Choose 2^{60} different random values A. Compute and store $\Delta F_2(A) = F_2(A) \oplus F_2(A \oplus \Delta)$. Then choose 2^{60} different random values X, compute $\Delta F_3^{-1}(X) = F_3^{-1}(X) \oplus F_3^{-1}(X \oplus \Delta)$, and match it to the stored $\Delta F_2(A)$. For each matched $(\Delta F_2(A),\ \Delta F_3^{-1}(X))$, compute $F_2(A) \oplus F_3^{-1}(X)$ and $F_2(A \oplus \Delta) \oplus F_3^{-1}(X)$ as K_1, and store (K_1, X). On average, there are 2^{57} stored (K_1, X).
2. Launch the same procedure between F_6 and F_7 as in step 1, and store 2^{57} (K_1, Y), where Y is the input value of F_6.
3. Match (K_1, X) and (K_1, Y) on the value of K_1, and store (K_1, X, Y) if (K_1, X) and (K_1, Y) are matched. On average there are 2^{50} (K_1, X, Y).
4. Find a 4-multicollision among (K_1, X, Y) on the value of K_1. Namely, find $(K_1^{(1)}, X_1^{(1)}, Y_1^{(1)})$, $(K_1^{(2)}, X_1^{(2)}, Y_1^{(2)})$, $(K_1^{(3)}, X_1^{(3)}, Y_1^{(3)})$ and $(K_1^{(4)}, X_1^{(4)}, Y_1^{(4)})$ with $K_1^{(1)} = K_1^{(2)} = K_1^{(3)} = K_1^{(4)}$. Compute $X_1^{(1)} \oplus Y_1^{(1)}$, $X_1^{(2)} \oplus Y_1^{(2)}$, $X_1^{(3)} \oplus Y_1^{(3)}$ and $X_1^{(4)} \oplus Y_1^{(4)}$.
5. Choose 2^{62} random value Z, and compute $G(Z) \oplus Z$, where G uses $K_1^{(i)}$, $1 \leq i \leq 4$ as K_1. Match the value of $G(Z) \oplus Z$ to $X_1^{(1)} \oplus Y_1^{(1)}$, $X_1^{(2)} \oplus Y_1^{(2)}$, $X_1^{(3)} \oplus Y_1^{(3)}$ and $X_1^{(4)} \oplus Y_1^{(4)}$. If a match to $(X^{(i)}, Y^{(i)})$ for some $1 \leq i \leq 4$ is found, compute $X^{(i)} \oplus Z$ as K_0, and output it with $K_1^{(i)}$ as K_1 and P, which can be trivially computed from $X_1^{(i)}$.

Complexity. The unit is one computation of the whole 10-step LED-128. Steps 1 and 2 are both with a complexity $2^{60} \times \frac{2}{10} \approx 2^{57.7}$ encryptions. Step 5 requires $2^{62} \times \frac{2}{10} \approx 2^{59.7}$ encryptions. Thus the overall complexity is $2^{57.7} + 2^{57.7} + 2^{59.7} \approx 2^{60.3}$, hence lower than 2^{64}.

Remark. As shown from the analysis above, again our attack is not related to the specification of the step functions, and can be applied to any 10-round construction with subkeys that come one after another, in a form of a chosen-key random-difference distinguisher. Thus we can conclude that:

Observation 2. *For any ten-round n-bit cipher with arbitrary round functions and alternating subkeys, exists a chosen-key distinguisher with time complexity less than 2^n queries.*

5 Conclusion

In this paper, we have presented various attacks on LED in the single-key and chosen-key models. We have improved the data complexity of the single-key attack on 16 rounds of LED-128 in terms of lower and known-plaintext data. We have also shown the first single-key attack on 24 rounds of LED-128. In the chosen-key model, we have given practical results on 32 rounds, and have reached as far as 40 rounds, using a novel chosen-key distinguisher.

The main contribution of this work is actually the idea of multicollisions and their applications. The vast majority of our results/attacks, in particular the attacks that penetrate through the largest number of rounds, are based on creating multicollisions for some intermediate states inside the cipher, thus obtaining a small set of independent values that are used further in meet-in-the-middle attacks. As we have seen from our analysis, the primary advantage of multicollisions is that they can be applied regardless of the specification of the internal rounds/steps. Both Observations 1 and 2 are surprising to a large extend as they state that the round transformation plays no role in the security against 2-round single-key and 10-round chosen-key attacks. This result is indeed due to the multicollisions and their property given above. Another condition for applying the observations is simplicity of the key schedule. Although it seems very compelling to use a trivial key schedule, especially in lightweight primitives, its application leads to a huge reduction of the security margin at least in the chosen-key model.

The two primary instances of LED apply 8, and 12 steps, respectively. However, when K_1 in LED-128 is fixed, then this cipher has only 6 steps, i.e. 2 steps less than LED-64. Although the steps now contain 8 rounds, the security margin of the cipher against attacks (such as most of our attacks presented here) independent of the step function, is less than the one of LED-64. Hence, it seems that an attack on 6-step LED-64, that does not use the structural properties of the step functions, might result in an attack on full-round LED-128. We were not able to trivially extend our 5-round chosen-key attack on LED-64, to 10-step chosen-key attack on LED-128, only because it uses a differential characteristic in the last step. We leave as an open research topic the problem of finding a 6-step attack on LED-64, independent of the step function.

Acknowledgement. The authors would like to thank the FSE 2013 reviewers and the LED team members Jian Guo and Thomas Peyrin for their valuable comments. Ivica Nikolić is supported by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03. Lei Wang and Shuang Wu are supported by the Singapore National Research Foundation Fellowship 2012 NRF-NRFF2012-06.

References

1. 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)

2. Bogdanov, A., Knudsen, L.R., Leander, G., Standaert, F.-X., Steinberger, J., Tischhauser, E.: Key-alternating ciphers in a provable setting: encryption using a small number of public permutations (extended abstract). In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 45–62. Springer, Heidelberg (2012)
3. Bogdanov, A., Rechberger, C.: A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 229–240. Springer, Heidelberg (2011)
4. Daemen, J.: Limitations of the Even-Mansour construction. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 495–498. Springer, Heidelberg (1993)
5. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in cryptography: the Even-Mansour scheme revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer, Heidelberg (2012)
6. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. J. Cryptology **10**(3), 151–162 (1997)
7. Feller, W.: An Introduction to Probability Theory and Its Applications, vol. 1, 3rd edn. Wiley, New York (1968)
8. Gilbert, H., Peyrin, T.: Super-Sbox cryptanalysis: improved attacks for AES-like permutations. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 365–383. Springer, Heidelberg (2010)
9. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. Cryptology ePrint Archive, Report 2012/600 (2012). http://eprint.iacr.org/2012/600
10. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
11. Isobe, T., Shibutani, K.: Security analysis of the lightweight block ciphers XTEA, LED and piccolo. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 71–86. Springer, Heidelberg (2012)
12. Lamberger, M., Mendel, F., Rechberger, C., Rijmen, V., Schläffer, M.: Rebound distinguishers: results on the full whirlpool compression function. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 126–143. Springer, Heidelberg (2009)
13. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The rebound attack: cryptanalysis of reduced whirlpool and Grøstl. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 260–276. Springer, Heidelberg (2009)
14. Mendel, F., Rijmen, V., Toz, D., Varici, K.: Differential analysis of the LED block cipher. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 190–207. Springer, Heidelberg (2012)
15. Patarin, J.: A proof of security in $O(2^n)$ for the Xor of two random permutations. In: Safavi-Naini, R. (ed.) ICITS 2008. LNCS, vol. 5155, pp. 232–248. Springer, Heidelberg (2008)
16. Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K.: Birthday paradox for multi-collisions. In: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 29–40. Springer, Heidelberg (2006)