Probabilistic Slide Cryptanalysis and Its Applications to LED-64 and Zorro

Hadi Soleimany

Aalto University School of Science,
Department of Information and Computer Science
hadi.soleimany@aalto.fi

Abstract. This paper aims to enhance the application of slide attack which is one of the most well-known cryptanalysis methods using self-similarity of a block cipher. The typical countermeasure against slide cryptanalysis is to use round-dependent constants. We present a new probabilistic technique and show how to overcome round-dependent constants in a slide attack against a block cipher based on the general Even-Mansour scheme with a single key. Our technique can potentially break more rounds than any previously known cryptanalysis for a specific class of block ciphers. We show employing round constants is not always sufficient to provide security against slide variant cryptanalysis, but also the relation between the round constants should be taken into account. To demonstrate the impact of our model we provide analysis of two round-reduced block ciphers LED-64 and Zorro, presented in CHES 2011 and CHES 2013, respectively. As a first application we recover the key for 16 rounds of Zorro. This result improves the best cryptanalysis presented by the designers which could be applied upto 12 rounds of its 24 rounds. In the case of LED-64 the cryptanalysis leads to the best results on 2-step reduced LED-64 in the known-plaintext model.

Key words: block cipher, slide attack, Zorro, LED-64, Even-Mansour

1 Introduction

Block ciphers can be attacked using a large variety of attacks employing different properties of the ciphers. Statistical cryptanalysis like differential [4] and linear [25] attacks make use of non-randomness characteristics of the cipher and the complexity of the attack is increased by adding more rounds to the cipher. In contrast, self-similarity cryptanalysis techniques are applicable on a small class of block ciphers and the complexity of the attacks is usually independent of the number of rounds. Self-similarity attacks exploit the weakness of the key schedule rather than non-random statistical properties of the cipher. *Slide cryptanalysis* is a well-known example of such techniques and it utilizes the symmetry properties of the cipher [7]. If an iterative block cipher with identical round functions has a periodic key schedule, it can be presented as a cascade of repeated copies of a single function F_k with an identical key k, where F_k consists of one or more

rounds of the cipher. Slide attacks are based on the observation that if two plaintexts P and P' satisfy the relation $P' = F_k(P)$ then $C' = F_k(C)$ holds for free. Such a pair $((P, C), (P', C'))$ is called the *slid pair*. Given a slid pair the security of the cipher is reduced to finding the key for the function F_k in the known plaintext model. In general, $2^{n/2}$ known plaintexts are required to find at least one slid pair for an n-bit block cipher. The total time complexity of the cryptanalysis consists of the complexities of three steps: preparing the required data, detecting a slid pair, and finally obtaining the key from the slide pair.

Slide attacks and other self-similarity cryptanalysis can be prevented by a careful design of a key schedule. But a strong key schedule cannot be achieved for free. It has impact on latency, power consumption and size of the implementation. Therefore the designers of lightweight ciphers such as PRINTcipher [23], LED [20], PRINCE [11] and Zorro [17], adopted another direction to establish security against self-similarity cryptanalysis. They introduced round-dependent constants added to the data input at each round to make the rounds different. Then a single master key is simply added after every fixed number of rounds called as *step*. Even if in such a construction each step has the same structure and the key used at every step is the same, the computation at each step is varied by the round constants. In this manner, the slide cryptanalysis can be prevented.

Such a cipher construction can be seen as an instance of the generalized Even-Mansour scheme [15] with a single key. It is defined as $C = E_K(P) = F_s(\cdots F_2(F_1(P \oplus K) \oplus K) \oplus K \cdots \oplus K)$ where F_i are constructed as cascades of the same fixed permutations but with different round constants. In this work, we investigate the security of this cipher structure and develop a new statistical variant of slide cryptanalysis. The idea is to slide one instance of encryption against another instance of encryption by one step and investigate, if for a pair $((P, C), (P', C'))$ it would be possible to predict, with significant probability, the difference $C \oplus F_s(C' \oplus K)$ given the difference $P' \oplus F_1(P \oplus K)$. By taking a probabilistic approach we can circumvent the devastating effect of different round constants in the deterministic slide cryptanalysis.

Potentially, the described attack has two main advantages compared to the classical differential cryptanalysis. The first one is that the attacker has more freedom to control the active S-boxes. If the difference in the round constants and data are identical, they cancel each other, which leads to a smaller number of active S-boxes in the characteristic. Example of such a situation is given in Section 4.2 for LED-64. While it is proved that the normal differential characteristic over four rounds has at least 25 S-boxes, we can find differential characteristics of LED-64 over four iterative rounds which in our slide setting have just 13 active S-boxes. We also note that even if we exploit a kind of related-key differential characteristics, our attack is in the single-key model, since we exploit the relation between the round constants instead of keys to control the differential pattern. The second merit is the existence of an efficient key-recovery method for our model. As each step of our target ciphers consists of several rounds, it makes it hard to convert a distinguisher to the key-recovery attack over more steps by

guessing just a part of the key. But we present an efficient method to obtain the key of s steps of the cipher based on the differential property in slide mode for $s-1$ steps.

To demonstrate the effect of our statistical slide framework, we apply key-recovery attack on block ciphers LED-64 and Zorro. We present a key-recovery cryptanalysis on a reduced 16-round version, while the best previous key-recovery cryptanalysis is on 12 rounds. Also we present a novel cryptanalysis on 2-reduced steps of LED-64 which leads to the best attack on this version of the cipher in the known-plaintext model. Our results and the previous results are summarized in Table 1.

Table 1. Summary of single-key attacks on round-reduced LED-64 and Zorro

Cipher	Attack Type	Steps	Data	Time	Memory	Source
Zorro	Impossible differential	2.5	2^{115}CP	2^{115}	2^{115}	[17]
	Meet-in-the-middle	3	2^{2}KP	2^{104}	-	[17]
	Probabilistic slide	4	$2^{123.62}$KP	$2^{123.8}$	$2^{123.62}$	Section 4.1
	Probabilistic slide	4	$2^{121.59}$KP	$2^{124.23}$	$2^{121.59}$	Section 4.1
	Internal differential†	6	$2^{54.25}$CP	$2^{54.25}$	$2^{54.25}$	[19]
LED-64	Meet-in-the-middle	2	2^{8}CP	2^{56}	2^{11}	[21]
	Meet-in-the-middle	2	2^{16}CP	2^{48}	2^{17}	[13]
	Meet-in-the-middle	2	2^{48}KP	2^{48}	2^{48}	[13]
	Generic	2	2^{45}KP	$2^{60.1}$	2^{60}	[14]
	Probabilistic slide	2	$2^{45.5}$KP	$2^{46.5}$	$2^{46.5}$	Section 4.2
	Probabilistic slide	2	$2^{41.5}$KP	$2^{51.5}$	$2^{42.5}$	Section 4.2
	Generic	3	2^{49}KP	$2^{60.2}$	2^{60}	[14]

† – this attack is applicable just on 2^{64} keys (out of 2^{128}), CP – Chosen Plaintexts, KP – Known Plaintext.

Some previous cryptanalysis of LED-64 do not exploit any specific property of the cipher. In [27] it is shown that the behavior of the function $x \oplus F(x)$, for a random permutation F, is not ideally random and they exploit this fact in a generic attack on the EM-construction with two alternative keys. The result is improved in [14] via a generic attack on a 3-step EM-construction with a single key. This attack is independent of the permutations but has high complexity. An accelerated exhaustive search for 2-step reduced version of LED-64 is presented in [21] in the chosen-plaintext model. Our attack requires only known plaintext and has much lower time complexity. Recently, parallel to our work, an advanced meet-in-the-middle cryptanalysis is presented for a 2-step version of LED-64 in IACR Eprint Archive [13], but is slightly slower than our cryptanalysis in known-plaintext model.

Zorro was presented recently at CHES 2013. It is a tweaked version of the standard block cipher AES and targets on efficient masking to establish side-channel security with better performance. The best key-recovery cryptanalysis

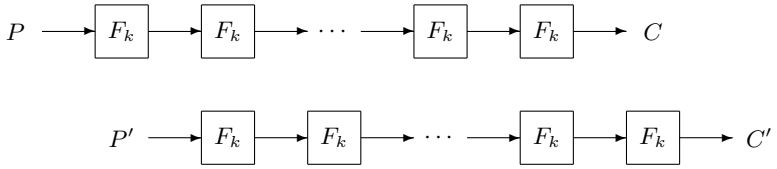

Fig. 1. Slide cryptanalysis

so far was presented by the designers in the classical single-key model. It is a meet-in-the-middle attack on 12 rounds. We carry out the first third-party cryptanalysis of Zorro and mount a key-recovery cryptanalysis on 16 rounds of Zorro. Let us also mention that simultaneously a cryptanalysis of the full Zorro is presented in [19]. It works for a fraction 2^{64} out of 2^{128} keys in the weak-key model and requires chosen plaintexts, whereas our result work for all keys in known-plaintext model.

This paper is organized as follows. In Section 2, we start by recalling the typical slide cryptanalysis and previous works, and then continue to introduce the basic concepts of our method. In Section 3, we describe the target ciphers Zorro and LED briefly. Section 4 gives an overview of our strategy to construct a distinguisher followed by applications of the probabilistic slide cryptanalysis on step-reduced Zorro and LED-64. We conclude in Section 5.

2 Slide Cryptanalysis

2.1 Basic Idea

A typical block cipher can be described as a cipher with iteration of a key-dependent function called as round. Each round is a keyed permutation $R_{k_i}(X)$, where k_i is the ith round key derived from the master key using a key schedule. More precisely, the encryption procedure of the n-bit plaintext P via the iterated application of r rounds is described as $X_i = R_{k_i}(X_{i-1})$, for $i = 1, \cdots, r$ where X_0 and X_r represents plaintext P and corresponding ciphertext C respectively. Let us assume that the cipher can be represented as an iteration of a single permutation F_k. Depending on the key-schedule this permutation may consist of one or more rounds of the cipher. If plaintexts P and P' satisfy the relation $P' = F_k(P)$, then $C' = F_k(C)$ holds for free independently of the number of rounds as illustrated in Figure 1. A pair $((P, C), (P', C'))$ with this property is called a *slid pair*. For an n-bit block cipher $P' = F(P)$ occurs with probability 2^{-n}. Given $2^{n/2}$ plaintext-ciphertexts (P, C) there exists 2^n pairs which emphasize to expect one slide pair. To convert the distinguisher to a key-recovery cryptanalysis the only requirement for F_k is to be weak enough against known-plaintext cryptanalysis.

2.2 Previous Works

Since the basic slide cryptanalysis is a too restrictive approach, several developments have been proposed to enhance the basic idea. Two advanced techniques termed as *sliding with a twist* and *complementation slide* are presented by Biryukov and Wagner [8]. If two keys are used alternatively in a Feistel cipher, then we can slide two instances of encryption against each other by one round to cancel the difference between the keys by the complementation slide property. In slide-with-a-twist cryptanalysis an instance of decryption is slid against an instance of encryption. This technique is applicable on another class of ciphers.

This idea is extended to introduce a weak key class of involution block ciphers in [5]. In [28] Feistel ciphers with independent pre- and post-whitening keys are studied and shown that there is a cryptanalysis with time and data complexity $n2^{n/2+1}$. Furuya uses the observation presented in [8] that if (P, P') is a slid pair, then $(F_K(P), F_K(P'))$ is also a slid pair. This technique provides more known plaintexts to mount efficient slide cryptanalysis on more complicated functions F_K [16]. Biham et al. pursue another direction which allows to find a slid pair much faster with the cost of almost the whole codebook [3]. As another direction slide cryptanalysis can be leveraged to a distinguisher on hash functions as it is done for the inner component of SHA-1 [30]. It does not seem useful for collision or (second) preimage cryptanalysis but works for distinguishing and also key recovery in MAC mode [18].

To the best of our knowledge, this paper is the first application of probabilistic slide cryptanalysis in the single-key model. In [8] it is suggested to use a differential property of the identical function to find the key from a slid pair which is a different approach and have not been applied in practice. Also in [29] the method of *realigning slide cryptanalysis* is presented to pass the middle round in a nondeterministic way in related-key scenario.

2.3 Probabilistic Slide Cryptanalysis

In this section we present our new technique which combines a usual slide attack with differential type characteristics. We focus our attention to an n-bit block cipher with general Even-Mansour construction that consists of s different permutations and one key. Analogically to the basic slide distinguisher we consider an encryption instance and slide it against another instance of the same encryption by one step. Due to the differences between round functions the basic slide cryptanalysis is not applicable. Assume there exists a sequence of differences $\mathcal{D} = \{\Delta_r : 0 \leq r \leq s-1\}$ such that $\Pr[F_r(x) \oplus F_{r-1}(x \oplus \Delta_{r-2}) = \Delta_{r-1}] = 2^{-p_{r-1}}$ where $0 \leq p_r$ and $2 \leq r \leq s$. Thanks to the equality of keys we obtain a differential type characteristic by concatenating the differences in $\mathcal{D}$. This characteristic has probability $2^{-p} = \prod_{r=1}^{s-1} 2^{-p_r}$. So $F_{s-1} \circ \cdots \circ F_1(x) \oplus F_s \circ \cdots \circ F_2(x \oplus \Delta_{in}) = \Delta_{out}$ holds with probability 2^{-p} where $\Delta_{in} = \Delta_0$ and $\Delta_{out} = \Delta_{s-1}$ as illustrated in Figure 2. In other words, if a pair (P, P') satisfies $F_1(P \oplus K) \oplus P' = \Delta_{in}$ then $C \oplus F_s^{-1}(C' \oplus K) = \Delta_{out}$ occurs with the probability 2^{-p}.

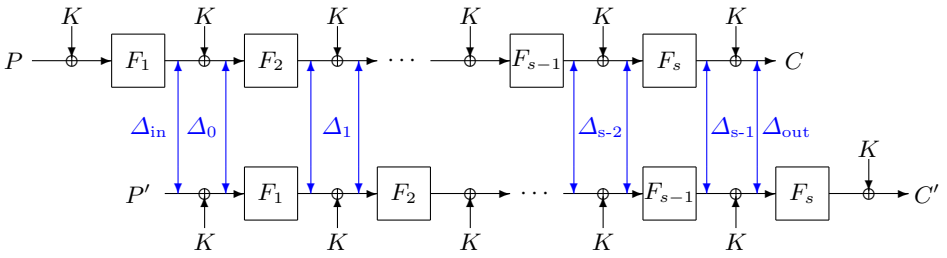

Fig. 2. Slide cryptanalysis on general Even-Mansour scheme with one key

Analogously to the usual slide attack, such a pair for which the characteristic holds is called the *right slid pair*. Given $2^m = 2^{n/2+p/2}$ known plaintext there exist 2^{n+p} pairs of which 2^p are expected to satisfy the relation $F_1(P \oplus K) \oplus P' = \Delta_{in}$ for the unknown key. The right slid pairs are among them. Since the input difference Δ_{in} yields the output Δ_{out} with probability 2^{-p}, we expect to get one right slid pair for the characteristic.

Key-recovery Next we describe the key-recovery algorithm. For a correct slid pair $((P, C), (P', C'))$ we have

$$C' \oplus F_s(C \oplus \Delta_{out}) = K = P \oplus F_1^{-1}(\Delta_{in} \oplus P',)$$

where K is the correct key. So the correct slid pair satisfies the relation

$$C' \oplus F_1^{-1}(\Delta_{in} \oplus P') = P \oplus F_s(C \oplus \Delta_{out}).$$

We utilize this property to find a slid pair efficiently by storing these values in hash tables for all plaintext-ciphertext pairs. The attack procedure is as follows:

1. Ask for the encryption of $2^m = 2^{n/2+p/2}$ arbitrary plaintexts.
2. For all plaintext-ciphertext pairs (P, C) compute the value of $C \oplus F_1^{-1}(P \oplus \Delta_{in})$ and store the computed value with the corresponding C in the hash table T_1. Sort them according to the value $C \oplus F_1^{-1}(P \oplus \Delta_{in})$.
3. For all plaintext-ciphertext pairs (P, C) compute the value of $P \oplus F_s(\Delta_{out} \oplus C)$ and store the computed value with C in the hash table T_2. Sort them according to the value $P \oplus F_s(\Delta_{out} \oplus C)$. Keep some (P, C) pairs to test the key candidates.
4. For each collision in the hash tables T_1 and T_2 find corresponding ciphertexts C and C' then compute a key candidate $K = C' \oplus F_s(C \oplus \Delta_{out})$. Use a (P, C) to test the key.

Step 1 requires 2^m full encryptions. To prepare each hash tables we compute one step of the cipher for all known plaintexts. So Step 2 and Step 3 requires totally $2 \cdot 2^m/r$ full encryptions. We expect to have $2^{2m-n} = 2^p$ key candidates to try in Step 4 which requires 2^p encryptions. The total time complexity is $2^m + 2^{m+1}/r + 2^p$. To perform the attack, one needs two hash tables T_1 and T_2 to store 2^m ordered pairs of n-bit values in both T_1 and T_2.

More Output Differences In this part we study the improvements of the described distinguisher. A natural improvement is to consider a differential instead of a single differential characteristic, and try to improve the estimate of the differential probability.

Another approach is to consider L different output differences Δ_{out}^i, $i \in \{1, \cdots, L\}$, to decrease the data requirement by increasing the total probability. This comes with the cost of repeating the attack algorithm L times and a small increase in the memory requirement. Without loss of generality we can assume that each output difference Δ_{out}^i occurs with equal probability 2^{-p} for a fixed input difference Δ_{in}, that is, if $F_1(P \oplus K) \oplus P' = \Delta_{in}$, then $C \oplus F_r^{-1}(C' \oplus K) = \Delta_{out}^i$ holds with probability 2^{-p}, for all $i \in \{1, \cdots, L\}$. If the probabilities are not equal, order the output differences in the decreasing order according to their probabilities.

The attack procedure is described as follows:

1. Get encryptions C of 2^m arbitrary plaintexts P.
2. For all pairs (P, C) compute the value of $C \oplus F_1^{-1}(P \oplus \Delta_{in})$ and store the computed value with P and C in a hash table T_1. Sort them according to the value $C \oplus F_1^{-1}(P \oplus \Delta_{in})$.
3. for $i \in \{0, \cdots, 2^\ell\}$
   - 3.1 Allocate $2 \cdot 2^m$ memory for the hash table T_2.
   - 3.2 For all plaintext-ciphertext pairs (P, C) compute the value of $P \oplus F_s(\Delta_{out}^i \oplus C)$ and store the computed value and the corresponding C in the hash table T_2. Sort them based on the value $P \oplus F_s(\Delta_{out}^i \oplus C)$.
   - 3.3 For each collision in the hash tables T_1 and T_2 find corresponding ciphertexts C and C' then compute the key candidate as $K = C' \oplus F_s(C \oplus \Delta_{out}^i)$. Given some P and C, test the key.
   - 3.4 If all key candidates are wrong, free the allocated memory of T_2.

We denote by ℓ the logarithm of L. Since $\Pr[C \oplus F_r^{-1}(C' \oplus K) = \Delta_{out}^i, \text{ for some } i \in \{1, \cdots, 2^\ell\} | F_1(P \oplus K) \oplus P' = \Delta_{in}] = 2^{\ell - p}$ the attack requires $2^m = 2^{n/2 + (p-\ell)/2}$ known plaintexts. Time complexity is $2^m + 2^m/r + 2^\ell(2^m/r + 2^{2m-n})$ encryptions which is dominated by $2^\ell(2^m/r + 2^{2m-n})$. Given L output differences, any number of them can be used in the attack allowing trade-off between data and time complexity. The memory requirement is the same for all $L > 1$. To perform the attack, one needs two hash tables T_1 and T_2 where T_1 is used to store 2^m triplets of n-bit values and T_2 to store 2^m ordered pairs n-bit values.

3 Target Ciphers

In this section we present a brief description of the block ciphers to be analyzed. We describe first the block cipher Zorro and continue with the description of the block cipher LED, and finally introduce the notation to be used in this paper.

3.1 Description of Zorro

Zorro is a 128-bit block cipher and supports 128-bit key. The state can be illustrated as a 4×4 matrix where each cell represents a byte. Zorro is a generalized Even-Mansour cipher consisting 6 steps. There exist no key schedule and after each step the same key is xored to the state. Each step is composed of 4 rounds. One round consists of four transformations. One is the adding of the round constant and the other three are borrowed from the AES and applied in the following order.

1. `SubCells`: A byte-wise transformation that applies an 8-bit S-box to each byte of the first row.
2. `AddConstants`: XOR operation between the first row of the state and the current round constant. Let r be the number of the current round represented as a byte, for $1 \leq r \leq 24$. Then the round constant is defined as $r \parallel r \parallel r \parallel r \ll 3$.
3. `ShiftRows`: A linear transformation that cyclically shifts the i'th row i bytes to the left.
4. `MixColumns`: A linear transformation represented by a 4×4 matrix over $GF(2^8)$.

The last two operations are exactly like in the AES. For the definition of S-box and more details we refer to [17].

3.2 Description of LED

LED is a 64-bit block cipher. Two main variants of the cipher are LED-64 and LED-128, which support the key sizes 64 and 128, respectively. The 64-bit state is represented by a 4×4 matrix, where each cell represents a nibble in $GF(2^4)$. The construction of LED-64 is a generalized Even-Mansour with one key and 8 steps. Each step includes four rounds. Each round consists of four transformations of which three are inspired by the AES.

1. `AddConstants` adds a round-dependent constant to the state. To construct the round constants one proceeds as follows. A string of six bits $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$ is initialized to zero. Then at each round, the bits are shifted to the left by one position, and the new value of rc_0 is computed as $rc_5 \oplus rc_4 \oplus 1$. Let us denote by $(ks_7, ks_6, ...ks_0)$ the bits of the byte that represents the key size. Then the corresponding round constant is defined as follows.

$ks_7 \|\| ks_6 \|\| ks_5 \|\| ks_4$	$0 \|\| rc_5 \|\| rc_4 \|\| rc_3$	0	0
$ks_7 \|\| ks_6 \|\| ks_5 \|\| ks_4 \oplus 1$	$0 \|\| rc_2 \|\| rc_1 \|\| rc_0$	0	0
$ks_3 \|\| ks_2 \|\| ks_1 \oplus 1 \|\| ks_0$	$0 \|\| rc_5 \|\| rc_4 \|\| rc_3$	0	0
$ks_3 \|\| ks_2 \|\| ks_1 \oplus 1 \|\| ks_0 \oplus 1$	$0 \|\| rc_2 \|\| rc_1 \|\| rc_0$	0	0

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S-box(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 2. S-box of LED

2. `SubCells` applies the same 4-bit to 4-bit S-box given in Table 2 in parallel on each of the 16 nibbles of the state.
3. `ShiftRows` cyclically rotates the i'th row by i nibble(s) to the left.
4. `MixColumns` multiplies each column by an MDS matrix $M = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix}$, over the field $GF(2^4)$ under the polynomial $x^4 + x + 1$.

3.3 Notations

Thanks to the similarity of Zorro and LED we can use almost the same notations to describe the states or operations for both ciphers. We represent the state as a 4×4 matrix where each cell is a byte or a nibble in Zorro and LED, respectively. We denote by X_r^I the input of the r'th round while X_r^S ,X_r^R, X_r^M and X_r^A denote the intermediate states after the application of SubCells, ShiftRows, MixColumns and AddConstants operations in the r'th round, respectively. We also denote the cell in the ith row and jth column of the state X by $X(i + 4j)$, where $0 \leq i, j \leq 4$. This notation is illustrated in Figure 3.

$X(0)$	$X(1)$	$X(2)$	$X(3)$
$X(4)$	$X(5)$	$X(6)$	$X(7)$
$X(8)$	$X(9)$	$X(10)$	$X(11)$
$X(12)$	$X(13)$	$X(14)$	$X(15)$

Fig. 3. State representation of Zorro and LED

Each step of Zorro and LED has four rounds. In our cryptanalysis, we slide two instances of encryption against each other by one step and compare their states. Let us denote by RC_r the round constant in r'th round and by DRC_r the difference between round constants in the rounds r and $r + 4$. Then $DRC_r = RC_r \oplus RC_{r+4}$. Let us note that in Zorro this difference has only four non-zero bytes $DRC_r(0, 1, 2, 3)$ on the first row. Similarly, the round constant difference DRC_r of LED can have non-zero nibbles only on the second column $DRC_r(1, 5, 9, 13)$. Throughout the paper SC, SR, MC and AC stands for SubCells, ShiftRows, MixColumns and AddConstants operations.

4 Applications of the Probabilistic Slide Cryptanalysis

Our aim is to find a differential type characteristic with high probability between two slid instances of the cipher as depicted in Figure 2. Finding differential characteristics have the highest probability among all possible choices is a challenging task as a general problem since even an automatic search for the whole space is not feasible. There are some techniques to make this simpler and speed up the search effectively. Matsui presents an algorithm to find the best differential characteristic and linear approximation in [26]. The algorithm uses a branch-and-bound method recursively to find the r-round characteristic of DES with highest probability based on the best $r-1$-round characteristics. The algorithm is not feasible for all ciphers but the main principles of the algorithm have been adopted widely in several works (for example look at [1, 6, 9, 10]). The probability of differential characteristic is proportional to the number of involved active S-boxes. One direction in the word-oriented block ciphers and hash functions is to find a general pattern of active and inactive differences holds in the cipher properties such that the number of active S-boxes is as small as possible. Next the differential characteristic with specific differential values for the existence pattern can be found by guess-and-determine methods. In this section we explore this approach for two block ciphers Zorro and LED-64 to construct a distinguisher described in Section 2.3 for each cipher separately and proceed by applying key-recovery cryptanalysis. We use the notations introduced in Section 3.3 to denote the various intermediate states X of the encryption process of P to C. We use similar notation for the pair (P', C') but now with X'.

4.1 Slide Cryptanalysis on Zorro

We start by an observation about the degrees of freedom of constructing a 2-round differential characteristic of Zorro block cipher. Let us consider a non-zero difference $X'^I_r(i) \oplus X^I_{r+4}(i)$ where $0 \leq i \leq 3$. After the S-box operation it may be transfered to a difference such that at the end of the round $X'^M_r(i) \oplus X^M_{r+4}(i) = 0$ which indicates that we can bypass the next round in the same byte position for free. So potentially for a given differential characteristic on $r-1$ rounds we can extend it over two more rounds with the cost of at most four active S-boxes. We use the following procedure to specify difference $X'^S_r(i) \oplus X^S_{r+4}(i)$ such that $X'^M_r(i)$ and $X^M_{r+4}(i)$ have identical values. Since `SC` and `AC` operations do not change the differences in the last three rows, the differences in the bytes $X'^R_r(i+4, i+8, i+12) \oplus X^R_{r+4}(i+4, i+8, i+12)$ are determined. By assuming $X'^M_r(i) \oplus X^M_{r+4}(i) = 0$, four bytes of the input and output of the Mixcolumn matrix are known and we are able to obtain the difference value $X'^R_r(i) \oplus X^R_{r+4}(i)$. After that $X'^S_r(i) \oplus X^S_{r+4}$ can be determined by xoring $X'^R_r(i) \oplus X^R_{r+4}(i)$ with $DRC_r(i)$. This procedure is illustrated in Figure 4, where we denote the bytes with known differences by '$*$' and aim to find unknown differences denoted by '?'.

Since about one half of all S-box differentials exist, to construct a 2-round differential characteristic as described, a fraction of 2^{-4} choices can match the

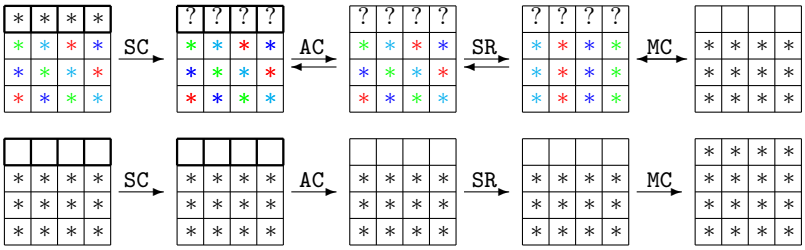

Fig. 4. Two rounds differential characteristic pattern with four active S-boxes

conditions of four S-boxes. We can start with an initial state such that the bytes in the first row have no difference. Since there exist 2^{96} different states we expect 2^{96-2r} states to satisfy the pattern of r-rounds. This shows that even for the full round cipher there exist numerous candidates. A naive question arises how one can exploit these degrees of freedom to create a differential characteristic which has still less active S-boxes. Our choice is to select the difference value $\Delta_{in} = X_1'^I \oplus X_5^I$ such that the first three rounds have no active S-boxes. The first row after `MC` has no difference with probability around $2^{-8\times 4} = 2^{-32}$. So if we start with a state by no difference in the first row a fraction $2^{-32\times 2}$ out of 2^{96} states can bypass two more rounds with probability one. While trying all 2^{96} states is not feasible as is described in [17] we can find these 3-round characteristics efficiently by utilizing an alternative method. In the remainder of this part we take a detailed look at this technique summarized in Algorithm 1.

We aim to find a 3-round characteristic that holds with probability one. If we guess the difference of bytes $X_2'^I(i+4, i+8) \oplus X_6^I(i+4, i+8)$ located in the i'th column where $0 \leq i \leq 3$, the difference of the third active byte $X_2'^I(i+12) \oplus X_6^I(i+12)$ in the same column can be obtained considering the Mixcolumn matrices of the first round. Hence we can find two first columns of $X_2'^I \oplus X_6^I$ by guessing only four bytes $X_2'^I(4,5,8,9) \oplus X_6^I(4,5,8,9)$. Consequently $X_2'^R(7,10,11,14) \oplus X_6^R(7,10,11,14)$ would be known after `SR`. By assumption $X_2'^M(2,3) \oplus X_6^M(2,3) = 0$ and using MixColumn matrices of the second round, difference $X_2'^R(7,14) \oplus X_6^R(7,14)$ can be found. We save the values $X_2^I(4,8,9,13) \oplus X_6'^I(4,8,9,13)$ in the row indexed by $X_2'^I(5,7,12,14) \oplus X_6^I(5,7,12,14)$ in the hash table T_1. Similarly in a separate computation by guessing four bytes $X_2'^I(6,7,10,11) \oplus X_6^I(6,7,10,11)$ the differences of four more bytes $X_2'^I(5,12,14,15) \oplus X_6^I(5,12,14,15)$ are obtained. We save the values $X_2'^I(6,10,11,15) \oplus X_6^I(6,10,11,15)$ in the row indexed by $X_2'^I(5,7,12,14) \oplus X_2^I(5,7,12,14)$ in the hash table T_2. Matching two hash tables leads to finding all 3-round differential characteristics which have probability one.

Key Recovery For all 2^{32} states obtained from Algorithm 1 we extend the characteristic by two rounds iteratively. The best differential characteristic we found for 12 rounds has probability $\Pr[X_{13}'^I \oplus X_{17}^I = \Delta_{out} | X_1'^I \oplus X_5^I = \Delta_{in}] =$

Algorithm 1 Finding 3-round characteristics of Zorro with probability one

for all $X_2'^I(4,8) \oplus X_6^I(4,8)$ **do**
  Find $X_2'^I(12) \oplus X_6^I(12)$ using the MixColumn matrix of the first round.
  for all $X_2'^I(5,9) \oplus X_6^I(5,9)$ **do**
    Find $X_2'^I(13) \oplus X_6^I(13)$ using the MixColumn matrix of the first round.
    Find $X_2'^I(7,14) \oplus X_6^I(7,14)$ using the MixColumn matrix of the second round.
    Save the value $X_2'^I(4,8,9,13) \oplus X_6^I(4,8,9,13)$ in the row is indexed by $X_2'^I(5,7,12,14) \oplus X_6^I(5,7,12,14)$ in T_1.
  end for
end for
for all $X_2'^I(6,10) \oplus X_6^I(6,10)$ **do**
  Find $X_2'^I(14) \oplus X_6^I(14)$using the MixColumn matrix of the first round.
  for all $X_2'^I(7,11) \oplus X_6^I(7,11)$ **do**
    Find $X_2'^I(15) \oplus X_6^I(15)$using the MixColumn matrix of the first round.
    Find $X_2'^I(5,12) \oplus X_6^I(5,12)$using the MixColumn matrix of the second round.
    Save the value $X_2'^I(6,10,11,15) \oplus X_6^I(6,10,11,15)$ in the row is indexed by $X_2'^I(5,7,12,14) \oplus X_6^I(5,7,12,14)$ in T_2.
  end for
end for

$2^{-119.24}$ where the values Δ_{in} and Δ_{out} with the details of characteristic are given in Appendix A.2. It leads to the the key-recovery cryptanalysis described in Section 2.3 on 16-reduced round of Zorro. The attack requires $2^{64+59.62} = 2^{123.62}$ known plaintexts and the time complexity is $2^{123.62} + 2^{124.62}/4 + 2^{119.24} \simeq 2^{123.8}$ encryptions. To reduce the data complexity we can allow degrees of freedom for an active S-box in the last round of the characteristic. There exist 25 different $\alpha \in GF(2^8)$ such that $\Pr[S(x) \oplus S(x \oplus \texttt{0x76}) = \alpha] = \texttt{2}^{-\texttt{6}}$. Let us consider the same characteristic in Appendix A.2 while $X_{16}^S(1) \oplus X_{12}'^S(1) = \alpha$. The probability for such a characteristic is $2^{-113.82} \cdot (25 \cdot 2^{-6}) = 2^{-115.17}$ which indicates that the data complexity decreases to $2^{64+57.59} = 2^{121.59}$ with the cost of increasing the time complexity to $25 \cdot (2^{121.59}/4 + 2^{115.17}) \simeq 2^{124.23}$ encryptions.

4.2 Slide Cryptanalysis of LED-64

Let us recall that the difference between RC_r and RC_{r+4} of LED-64 has nonzero value only in the second column. Then we start by looking at the state after `AC` in an arbitrary round r to investigate different scenarios. Since we are interested in characteristic with less active S-boxes, one may think the best case happens when all active nibbles in $X_r'^I \oplus X_{r+4}^I$ get canceled by the difference DRC_r to bypass `SC` with probability one. We note it can activate four nibbles $X_{r+1}'^A(1,5,9,13) \oplus X_{r+5}^A(1,5,9,13)$. Next each active nibble propagates to a different column after `SR` and makes all nibbles active at $X_{r+1}'^M \oplus X_{r+5}^M$, which is against our goal. Another choice is to have just one active nibble $X_r'^A(1) \oplus X_{r+4}^A(1)$ such that after `MC` it transfers to four nibbles which cancel the three out of four nibbles differences injected by round constants in the next round. It can be considered as an iterative 1-round characteristic. After testing this approach we found that

due to the differences between round constants we cannot utilize this iteratively for the round constants chosen by the designers, because it works just for one round. So we chose a moderate strategy to find the best pattern. First we try to hesitate activate the nibbles from the first, third and forth columns of $X'^I_r \oplus X^I_{r+4}$ since we cannot cancel them by the differences of round constants. Secondly we aim to cancel as many nibbles in the second column as possible. To find the best pattern we start from the middle rounds and try to extend it in both backward and forward directions. It is proved in [20] that any differential characteristic over four consecutive rounds of the cipher has at least 25 active S-boxes and probability at most 2^{-50}. Thanks to the differences in the round-constants, the best characteristic we found for four rounds has probability 2^{-27} and just 13 active S-boxes. This demonstrates the superiority of our model in comparison with differential cryptanalysis of LED-64.

Key Recovery The 4-round differential characteristic illustrated in Appendix A.1 has probability 2^{-27}. This property enables us to retrieve the key of 8-round reduced of LED-64 with the same technique described in Section 2.3. The cryptanalysis requires $2^{32+13.5} = 2^{45.5}$ known plaintexts and the time complexity is roughly $2^{45.5} + 2^{46.5}/2 + 2^{27} \simeq 2^{46.5}$ encryptions. To have less data complexity we can consider a truncated type differential for the last round. The differences 2, c, d and 6 can be transfered through the S-box to the set of differences $\mathcal{A}_1 = \{3, 5, 6, a, c, d, e\}$, $\mathcal{A}_2 = \{2, 5, 7, 8, 9, a, e\}$, $\mathcal{A}_3 = \{1, 2, 3, 4, 7, a, b\}$ and $\mathcal{A}_4 = \{2, 6, 8, b, c, f\}$ respectively. If $X^I_5 \oplus P' = \Delta_{in}$ holds for the given Δ_{in} in Appendix A.1 then the truncated difference $X^S_8 \oplus X'^S_4 \in \{0a_1000a_2000a_3000a_400 | a_i \in \mathcal{A}_i, 1 \leq i \leq 4\}$ with its corresponding truncated difference $\Delta_{out} = X^M_8 \oplus X'^M_4$ holds with probability 2^{-19}. Using this characteristic the data complexity decreases to $2^{32+9.5} = 2^{41.5}$ known plaintexts while the time complexity increases to $6 \cdot 7^3(2^{41.5}/2 + 2^{19}) \simeq 2^{51.5}$ encryptions.

5 Conclusion

In this paper we provide a new insight into slide cryptanalysis which is illustrated by cryptanalysis of step-reduced block ciphers Zorro and LED-64. We describe a new framework to enhance slide cryptanalysis against general Even-Mansour scheme with one key in a probabilistic setting. Our method exploits some features from related-key differential cryptanalysis to build a kind of differential characteristic that is applicable in the single key model. In the related-key cryptanalysis model [2, 22] one can consider the encryption under unknown secret keys but with a determined difference, which allows attacker to control the data difference by differences injected by the key difference. The probabilistic slide cryptanalysis presented in this paper is inspired by the same idea but instead of using two different keys it slides a copy of encryption to take advantage of the round constant differences in a single key model. Since known statistical cryptanalysis is not affected by the values of round constants, choosing their values usually has not been taken into account by the designers of block ciphers

(for example look at [24, 31]). In this work we shed more light on how round constants can potentially weaken the security of the cipher. One possible direction of future research is to inquire the application of probabilistic slide cryptanalysis against other block ciphers based on the general Even-Mansour scheme with a single key like PRINCE, PRINTcipher and 3-WAY [12].

Acknowledgments. The author wishes to thank Kaisa Nyberg for the helpful discussions, detailed comments and insightful suggestions that significantly improved the quality of this paper. The program for finding the characteristic used for cryptanalysis of Zorro was performed on the Triton computing cluster provided by the Aalto University Science-IT programme. The work of Hadi Soleimany is supported by Helsinki Doctoral Program in Computer Science - Advanced Computing and Intelligent Systems (HECSE). He was partially supported by the Nokia Foundation which is also gratefully acknowledged.

References

1. Kazumaro Aoki, Kunio Kobayashi, and Shiho Moriai. Best Differential Characteristic Search of FEAL. In Eli Biham, editor, *FSE 1997*, volume 1267 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 1997.
2. Eli Biham. New Types of Cryptanalytic Attacks Using Related Keys. *J. Cryptology*, 7(4):229–246, 1994.
3. Eli Biham, Orr Dunkelman, and Nathan Keller. Improved Slide Attacks. In Alex Biryukov, editor, *FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 153–166. Springer, 2007.
4. Eli Biham and Adi Shamir. differential Cryptanalysis of DES-like Cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *CRYPTO 1990*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.
5. Alex Biryukov. Analysis of Involutional Ciphers: Khazad and Anubis. In Thomas Johansson, editor, *FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 45–53. Springer, 2003.
6. Alex Biryukov and Ivica Nikolic. Automatic Search for Related-Key Differential Characteristics in Byte-Oriented Block Ciphers: Application to AES, Camellia, Khazad and Others. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2010.
7. Alex Biryukov and David Wagner. Slide Attacks. In Lars R. Knudsen, editor, *FSE 1999*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999.
8. Alex Biryukov and David Wagner. Advanced Slide Attacks. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 589–606. Springer, 2000.
9. Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. spongent: A Lightweight Hash Function. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011.
10. Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. Spongent: The design space of lightweight cryptographic hashing. *IEEE Trans. Computers*, 62(10):2041–2053, 2013.

11. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.
12. Joan Daemen, René Govaerts, and Joos Vandewalle. A New Approach to Block Cipher Design. In Ross J. Anderson, editor, *FSE*, volume 809 of *Lecture Notes in Computer Science*, pages 18–32. Springer, 1993.
13. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved Linear Sieving Techniques withApplications to Step-Reduced LED-64. Cryptology ePrint Archive, Report 2013/634, 2013. `http://eprint.iacr.org/`.
14. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. Key Recovery Attacks on 3-round Even-Mansour, 8-step LED-128, and Full AES2. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 337–356. Springer, 2013.
15. Shimon Even and Yishay Mansour. A Construction of a Cipher From a Single Pseudorandom Permutation. In Hideki Imai, Ronald L. Rivest, and Tsutomu Matsumoto, editors, *ASIACRYPT 1991*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 1991.
16. Soichi Furuya. Slide Attacks with a Known-Plaintext Cryptanalysis. In Mitsuru Matsui, editor, *ICISC 2002*, volume 2288 of *Lecture Notes in Computer Science*, pages 214–225. Springer, 2002.
17. Benoît Gérard, Vincent Grosso, María Naya-Plasencia, and François-Xavier Standaert. Block Ciphers That Are Easier to Mask: How Far Can We Go? In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 383–399. Springer, 2013.
18. Michael Gorski, Stefan Lucks, and Thomas Peyrin. Slide Attacks on a Class of Hash Functions. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 2008.
19. Jian Guo, Ivica Nikolic, Thomas Peyrin, and Lei Wang. Cryptanalysis of Zorro. Cryptology ePrint Archive, Report 2013/713, 2013. `http://eprint.iacr.org/`.
20. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED Block Cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011.
21. Takanori Isobe and Kyoji Shibutani. Security Analysis of the Lightweight Block Ciphers XTEA, LED and Piccolo. In Willy Susilo, Yi Mu, and Jennifer Seberry, editors, *ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2012.
22. John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Yongfei Han, Tatsuaki Okamoto, and Sihan Qing, editors, *ICICS 1997*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.
23. Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. PRINTcipher: A Block Cipher for IC-Printing. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.
24. Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2011.

25. Mitsuru Matsui. Linear Cryptoanalysis Method for DES Cipher. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 386–397. Springer, 1994.
26. Mitsuru Matsui. On Correlation Between the Order of S-boxes and the Strength of DES. In Alfredo De Santis, editor, *EUROCRYPT 1994*, volume 950 of *Lecture Notes in Computer Science*, pages 366–375. Springer, 1994.
27. Ivica Nikolic, Lei Wang, and Shuang Wu. Cryptanalysis of Round-Reduced LED. In Shiho Moriai, editor, *FSE 2013*, Lecture Notes in Computer Science, to appear.
28. Paul Onions. On the Strength of Simply-Iterated Feistel Ciphers with Whitening Keys. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 63–69. Springer, 2001.
29. Raphael Chung-Wei Phan. Advanced Slide Attacks Revisited: Realigning Slide on DES. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 263–276. Springer, 2005.
30. Markku-Juhani Olavi Saarinen. Cryptanalysis of Block Ciphers Based on SHA-1 and MD5. In Thomas Johansson, editor, *FSE 2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 36–44. Springer, 2003.
31. Hadi Soleimany, Céline Blondeau, Xiaoli Yu, Wenling Wu, Kaisa Nyberg, Huiling Zhang, Lei Zhang, and Yanfeng Wang. Reflection Cryptanalysis of PRINCE-like Ciphers. In Shiho Moriai, editor, *FSE 2013*, Lecture Notes in Computer Science. Springer, to appear.

A Appendix

A.1 Characteristic used for cryptanalysis of LED-64

State	Difference	Probability
$\Delta_{in} = X_5^I \oplus P'$	0250060b33010700	
$X_5^A \oplus X_1'^A$	0150000b30010100	
$X_5^S \oplus X_1'^S$	07c0000860070900	2^{-12}
$X_5^R \oplus X_1'^R$	07c0008007600090	
$X_5^M \oplus X_1'^M$	0100051007000500	
$X_6^A \oplus X_2'^A$	0600001000000000	
$X_6^S \oplus X_2'^S$	0c0000d000000000	2^{-5}
$X_6^R \oplus X_2'^R$	0c000d0000000000	
$X_6^M \oplus X_2'^M$	0800020007000200	
$X_7^A \oplus X_3'^A$	0f00000000000000	
$X_7^S \oplus X_3'^S$	0100000000000000	2^{-2}
$X_7^R \oplus X_3'^R$	0100000000000000	
$X_7^M \oplus X_3'^M$	040008000b000200	
$X_8^A \oplus X_4'^A$	02000c000d000600	
$X_8^S \oplus X_4'^S$	0500050002000b00	2^{-8}
$X_8^R \oplus X_4'^R$	05005000000200b0	
$\Delta_{out} = X_8^M \oplus X_4'^M$	5754defa31c7aa9d	

A.2 Characteristic used for cryptanalysis of Zorro

State	Difference	Probability
$\Delta_{in} = X_5^I \oplus P'$	00000000d52c6f72120a92b50c8c2eee	
$X_5^S \oplus X_1'^S$	00000000d52c6f72120a92b50c8c2eee	1
$X_5^A \oplus X_1^A$	04040420d52c6f72120a92b50c8c2eee	
$X_5^R \oplus X_1'^R$	040404202c6f72d592b5120aee0c8c2e	
$X_5^M \oplus X_1'^M$	000000001f125aa13e0edd9375ce6fe3	
$X_6^S \oplus X_2'^S$	000000001f125aa13e0edd9375ce6fe3	1
$X_6^A \oplus X_2'^A$	040404201f125aa13e0edd9375ce6fe3	
$X_6^R \oplus X_2'^R$	04040420125aa11fdd933e0ee375ce6f	
$X_6^M \oplus X_2'^M$	00000000bf6bd16389fc90921e2f14af	
$X_7^S \oplus X_3'^S$	00000000bf6bd16389fc90921e2f14af	1
$X_7^A \oplus X_3'^A$	04040420bf6bd16389fc90921e2f14af	
$X_7^R \oplus X_3'^R$	040404206bd163bf909289fcaf1e2f14	
$X_7^M \oplus X_3'^M$	8aec0b72d60e6d4ebec81f40b273b80b	
$X_8^S \oplus X_4'^S$	0fa38ee5d60e6d4ebec81f40b273b80b	$2^{-24.41}$
$X_8^A \oplus X_4'^A$	03af8285d60e6d4ebec81f40b273b80b	
$X_8^R \oplus X_4'^R$	03af82850e6d4ed61f40bec80bb273b8	
$X_8^M \oplus X_4'^M$	000000003507b4c92e8f3e0b02b88be1	
$X_9^S \oplus X_5'^S$	000000003507b4c92e8f3e0b02b88be1	1
$X_9^A \oplus X_5'^A$	0c0c0c603507b4c92e8f3e0b02b88be1	
$X_9^R \oplus X_5'^R$	0c0c0c6007b4c9353e0b2e8fe102b88b	
$X_9^M \oplus X_5'^M$	ced6ce9ba1604f0b4fa84ad6f4af9817	
$X_{10}^S \oplus X_6'^S$	fff8ff04a1604f0b4fa84ad6f4af9817	2^{-26}
$X_{10}^A \oplus X_6'^A$	f3f4f364a1604f0b4fa84ad6f4af9817	
$X_{10}^R \oplus X_6'^R$	f3f4f364604f0ba14ad64fa817f4af98	
$X_{10}^M \oplus X_6'^M$	00000000faff9b463e0b8c3d0a6d0f8e	
$X_{11}^S \oplus X_7'^S$	00000000faff9b463e0b8c3d0a6d0f8e	1
$X_{11}^A \oplus X_7'^A$	0c0c0c60faff9b463e0b8c3d0a6d0f8e	
$X_{11}^R \oplus X_7'^R$	0c0c0c60ff9b46fa8c3d3e0b8e0a6d0f	
$X_{11}^M \oplus X_7'^M$	009981d1e86caf9d79f3819d60a6b64f	
$X_{12}^S \oplus X_8'^S$	0082b813e86caf9d79f3819d60a6b64f	2^{-21}
$X_{12}^A \oplus X_8'^A$	0486bc33e86caf9d79f3819d60a6b64f	
$X_{12}^R \oplus X_8'^R$	0486bc336caf9de8819d79f34f60a6b6	
$X_{12}^M \oplus X_8'^M$	720000000b1fb040a0a822e77f636c39	
$X_{13}^S \oplus X_9'^S$	190000000b1fb040a0a822e77f636c39	2^{-6}
$X_{13}^A \oplus X_9'^A$	1d0404200b1fb040a0a822e77f636c39	
$X_{13}^R \oplus X_9'^R$	1d0404201fb0400b22e7a0a8397f636c	
$X_{13}^M \oplus X_9'^M$	005b0b997c321cb90de0bad468a52a1b	
$X_{14}^S \oplus X_{10}'^S$	004838077c321cb90de0bad468a52a1b	2^{-19}
$X_{14}^A \oplus X_{10}'^A$	044c3c277c321cb90de0bad468a52a1b	
$X_{14}^R \oplus X_{10}'^R$	044c3c27321cb97cbad40de01b68a52a	
$X_{14}^M \oplus X_{10}'^M$	ff000000ae7be7ce745b6bfeb2cca1a1	
$X_{15}^S \oplus X_{11}'^S$	aa000000ae7be7ce745b6bfeb2cca1a1	2^{-7}
$X_{15}^A \oplus X_{11}'^A$	ae040420ae7be7ce745b6bfeb2cca1a1	
$X_{15}^R \oplus X_{11}'^R$	ae0404207be7ceae6bfe745ba1b2cca1	
$X_{15}^M \oplus X_{11}'^M$	0076f953447ad32bfbc96dc0a06a35cc	
$X_{16}^S \oplus X_{12}'^S$	000b84ed447ad32bfbc96dc0a06a35cc	$2^{-15.83}$
$X_{16}^A \oplus X_{12}'^A$	1c17980d447ad32bfbc96dc0a06a35cc	
$X_{16}^R \oplus X_{12}'^R$	1c17980d7ad32b446dc0fbc9cca06a35	
$\Delta_{out} = X_{16}^M \oplus X_{12}'^M$	1720c72a9351b2f0f3a4e09fb071b7f0	