Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro

Yanfeng Wang[1,3] , Wenling Wu[1,2], Zhiyuan Guo[1,3], and Xiaoli Yu[1,3]

[1] Trusted Computing and Information Assurance Laboratory, Institute of Software,
Chinese Academy of Sciences, Beijing 100190, P.R. China
[2] State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing 100190, P.R. China
[3] Graduate University of Chinese Academy of Sciences, Beijing 100049, P.R. China
{wwl,wangyanfeng}@tca.iscas.ac.cn

Abstract. Zorro is an AES-like lightweight block cipher proposed in CHES 2013, which only uses 4 S-boxes per round. The designers showed the resistance of the cipher against various attacks and concluded the cipher has a large security margin. Recently, Guo et. al [1] have given a key recovery attack on full-round Zorro by using the internal differential characteristics. However, the attack only works for 2^{64} out of 2^{128} keys. In this paper, the secret key selected randomly from the whole key space can be recovered much faster than the brute-force attack. We first observe that the fourth power of the MDS matrix used in Zorro(or AES) equals to the identity matrix. Moveover, several iterated differential characteristics and iterated linear trails are found due to the interesting property. We select three characteristics with the largest probability to give the key recovery attack on Zorro and a linear trail with the largest correlation to show a linear distinguishing attack with $2^{105.3}$ known plaintexts. The results show that the security of Zorro against linear and differential cryptanalysis evaluated by designers is insufficient and the security margin of Zorro is not enough.

Keywords: Zorro, block cipher, differential cryptanalysis, linear distinguisher.

1 Introduction

Block ciphers are used as building blocks for many symmetric cryptographic primitives for encryption, authentication, pseudo-random number generation, and hash functions. Security of these primitives is evaluated in regard to known attacks against block ciphers. Among the different types of attacks, the statistical ones exploit non-uniform behavior of the data extracted from the cipher to distinguish the block cipher from random permutations. Differential cryptanalysis[2] and linear cryptanalysis[3] are the most prominent statistical attacks against block ciphers.

Differential cryptanalysis has been introduced in 1990 by Biham and Shamir in order to break the DES block cipher. This statistical cryptanalysis exploits

I. Boureanu, P. Owesarski, and S. Vaudenay (Eds.): ACNS 2014, LNCS 8479, pp. 308–323, 2014.

the existence of a differential, i.e., a pair $(\triangle_{in}, \triangle_{out})$ of differences such that for a given input difference $\triangle_{in}$, the output difference after encryption equals $\triangle_{out}$ with a high probability. For a b-bit random permutation, the probability is about 2^{-b}. The gap of the probability results in a distinguisher between the cipher and the random permutation, which is often extended to distinguish the correct key and the wrong keys. In 1993, the iterated differentials are proposed to analyze DES and s2-DES[4]. Since then, the differential cryptanalysis is always a hot topic of cryptanalysis[5,6,7]. The problem of estimating the data complexity, time complexity and success probability of a differential cryptanalysis is far from being simple. In 2011, [8] presented a general method (Algorithm 1) for finding an accurate number of samples to reach given error probabilities which can be applied to the differential cryptanalysis.

Linear cryptanalysis[9,10] is a known-plaintext attack proposed in 1993 by Matsui to break DES. It exploits the correlation between linear combinations of input bits and linear combinations of output bits of the block cipher. If the correlation between input and output equals C, the required amount of known plaintexts is about C^{-2} if we want to distinguish the block cipher from the random permutation with a high success probability.

The large development of low resource devices such as RFID tags and sensor nodes increases the need to provide security among such devices. The implementation costs should be taken into account when choosing security algorithms for resource-limited devices. Symmetric-key algorithms, especially block ciphers, still play an important role in the security of embedded systems. Recently, a lot of block ciphers and authenticated encryption ciphers suitable for these environments have been designed, such as PRESENT[11], KATAN & KTANTAN[12], PRINT[13], LBlock[14], FIDES[15], Piccolo[16], LED[17] etc.

Zorro[18] is a new lightweight block cipher proposed at CHES 2013. It is an AES-like block cipher and is designed to improve the side-channel resistance of AES[19]. The secret key is added to the state only after each 4 rounds as in the block cipher LED-64. The S-box layer of Zorro only applies four same S-boxes to the first row per round and the S-box is different from that of AES. Besides, the MC operation is the same as AES. The designers have evaluated the security of the cipher against various methods. For differential/linear cryptanalysis, authors found a balance between the number of inactive S-boxes and degrees of freedom for the differential (or linear) paths. Considering the average number of conditions imposed at each round, designers concluded that 14(or 16) rounds are the upper bound for building a classical differential(or linear) path. Finally, a 12-round meet-in-the-middle attack was shown as the best powerful attack on Zorro in the single key model. Recently, Guo et. al[1] have given a key recovery attack on full-round Zorro by using the internal differential characteristics, while it only works for 2^{64} keys of the whole key space.

In this paper, we revaluated the security of Zorro against differential cryptanalysis and linear cryptanalysis. As mentioned in [1], the main weakness of Zorro includes defining a new S-box and applying only four S-boxes to the first row per round. Besides, we observed that the fourth power of the MDS matrix of

Zorro(or AES) is equal to the identity matrix. Coincidentally, one step of Zorro consists of four rounds with four MDS matrix transformations. Interestingly, there exist several iterated differential characteristics with a high probability and iterated linear trails with a high correlation for one step of Zorro. Furthermore, we can recover the secret key of the full-round Zorro based on a 23-round differential characteristic with a time complexity of 2^{106} full-round Zorro encryptions. Interestingly, no matter how many plaintext-ciphertext pairs are given, the time complexity of filtering the right key is at least 2^{96} full-round encryptions based on the 23-round distinguisher. In order to clarify the special property of the structure used in Zorro, another TMTO attack based on a 22-round differential characteristic is also shown and it only costs about 2^{64} full-round Zorro encryptions to filter out the right key. Meanwhile, $1/C^2$ of some linear trails of full-round Zorro is also lower than the size of the plaintext space 2^{128}. Thus, we can obtain a full-round linear distinguisher for Zorro with $1/C^2$ known plaintexts. All in all, the above results have threatened the theoretical security of the full-round Zorro.

The remainder of this paper is organized as follows. Section 2 gives a brief description of Zorro block cipher. Section 3 proposes some iterated differential characteristics for one step of Zorro and shows two key recovery attacks on full-round Zorro. Section 4 presents a linear distinguisher of full-round Zorro based on the theory of correlation matrix. Finally, Section 5 concludes this paper.

2 A Brief Description of Zorro

The block cipher Zorro has 128-bit key and 128-bit state. It iterates 24 rounds and the 24 rounds are divided into 6 steps of 4 rounds each.

Encryption Algorithm. As in AES-128, the state in Zorro is regarded as 4×4 matrix of bytes, and one round consists of four distinct transformations: SB^*, AC, SR and MC. SB^* is the S-box layer where only 4 same S-boxes are applied to the 4 bytes of the first row in the state matrix. The S-box used in Zorro is different from the one of AES and the definition of S-box is referred to Appendix A. Next, AC is the addition of round constants in round i. Specifically, the four constants $(i, i, i, i{<<}3)$ are added to the four bytes of the first row. Finally, the last two transformations, SR and MC, are the AES's ShiftRows and MixColumns.

Key Schedule Algorithm. The key schedule algorithm of Zorro is similar to that of LED. Before the first and after each step, the master key is bitwisely added to the state and the same addition is done after the last step.

Let us focus on MC(MixColumn) used in Zorro which is a permutation operation on the state column by column. The matrix multiplication can be shown as:

$$M = \begin{pmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{pmatrix}, \qquad M^{-1} = \begin{pmatrix} 0E\ \mathrm{0B\ 0D\ 09} \\ \mathrm{09\ 0E\ 0B\ 0D} \\ 0D\ \mathrm{09\ 0E\ 0B} \\ 0B\ \mathrm{0D\ 09\ 0E} \end{pmatrix}.$$

Interestingly, the following equation is true:

$$M^4 = \begin{pmatrix} 01\ 00\ 00\ 00 \\ 00\ 01\ 00\ 00 \\ 00\ 00\ 01\ 00 \\ 00\ 00\ 00\ 01 \end{pmatrix}.$$

Combined with the fact that only 4 S-boxes are applied to the first row for every round, iterated differential characteristics and linear trails are found for four rounds(one step) of Zorro.

3 Differential Cryptanalysis of Full-Round Zorro

Differential cryptanalysis defines characteristics that describe possible evolvements of the differences through the cipher. For non-linear operations (such as S-boxes), it is possible to predict statistical information on the output difference given the input difference by generating the differential distribution table (DDT). If the expected difference for the intermediate data before the last few rounds is given, it may be possible to deduce the unknown key by a statistical analysis. The attack is a chosen plaintext attack that is performed in two phases: In the data collection phase the attacker requests encryption of a large number of pairs of plaintexts, where the differences of all the plaintext pairs are selected to have the input difference of the characteristic. In the data analysis phase the attacker then recovers the key from the collected ciphertexts.

Generally, the total probability of a differential characteristic is the product of the probabilities of each round assuming that the round functions are independent. For Zorro, the secret key is added to the data every four rounds. If we add one value to the input and one at the output of the step, 4 rounds of Zorro can be seen as a step that has no constants in the rounds[1]. As a result, the assumption that the step functions are independent is more rational than the one that round functions are independent for Zorro. In this section, we will present two key recovery attacks on full-round Zorro. The basic one uses a 23-round distinguisher to give an attack with a time complexity of 2^{106} and a memory complexity of 2^{32}. In order to clarify the special structure used in Zorro, another attack with a key searching time complexity of 2^{64} and a memory complexity of 2^{64} is also described.

3.1 Iterated Differential Characteristic

As mentioned by designers, the most damaging differential patterns are those that would exclude active bytes affected by non-linear operations. This kind of differential characteristic with probability 1 exists for at most two rounds. We extend one type of the differential pattern to 4 rounds by adding 4 active bytes. In order to keep the high differential probability for one step, we aim to build iterated differential trails taking advantage of the fact that $M^4 = I$. In order to reduce the searching cases and remove the influence of ShiftRow, we set the original four-byte differences in each row all equal and the first row all zero. The obtained active model is shown in Figure 1. The big squares represent states, small squares represent bytes, white bytes are the ones with zero difference, gray bytes are the ones with a non-zero difference and the letters in gray bytes present the values of difference. As shown in Figure 1, the probability of the path from #1 to #7 is always 1 as the S-boxes are all inactive. If the output differences of all the 4 active S-boxes in the fourth round are equal to the input differences, then the differences of #1 are equal to those of #9 because $M^4 = I$.

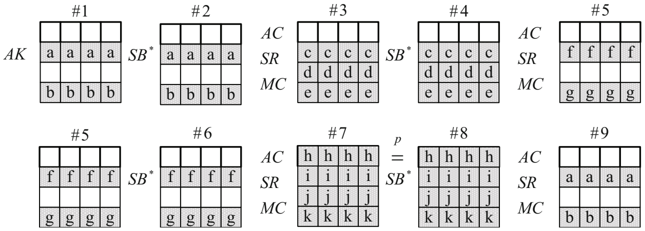


Fig. 1. Iterated differential characteristic of four rounds Zorro

Firstly, we find that 255 different values of (a, b) make the path from #1 to #7 with probability 1. After searching the differential distribution table (DDT) of the S-box used in Zorro, 101 original differences make the path from #7 to #9 possible. The probability of the differential characteristic from #1 to #9(four rounds) is determined by the value of (h, h) in DDT. Specifically, if the value of (h, h) in DDT of S-box is m, then there are m different solutions with the equation $S(x) \bigoplus S(x \bigoplus h) = h$. Thus, the probability of the differential characteristic p shown in Figure 1 is $(m/256)^4$. Obviously, the largest m means the highest probability of the characteristic. We find that the maximum m is equal to 6 and 3 options of h make the probability of the differential characteristic be $(6/256)^4 \approx 2^{-21.66}$. The corresponding values of differences expressed in decimal are shown in Table 1. Furthermore, if the state of #1 is replaced by #3, #5 or #7, we can obtain another three iterated differential characteristics with the same probability.

3.2 Basic Key Recovery Attack on Full-Round Zorro

In order to recover the secret key of Zorro, three iterated differential characteristics of 23-round Zorro are used to distinguish the right key and the wrong keys. With the assumption that the step functions of Zorro are independent, we can

Table 1. Three kinds of iterated differential characteristics on one step

NO	a	b	c	d	e	f	g	h	i	j	k
1	22	58	22	88	98	166	138	123	221	35	169
2	107	189	107	183	10	30	200	234	244	93	149
3	88	232	88	123	147	174	30	247	89	140	146

extend the iterated characteristics to 5 steps of Zorro. The probability becomes $2^{-21.66\times5}$=$2^{-108.3}$ which is much lower than 2^{-128} for the random permutation. Meanwhile, the 23-round differential characteristics shown in Figure 2 have the same probability $2^{-108.3}$ as the path from #1 to #7 with probability 1, where the values of a and b are referred to Table 1. With another assumption that the secret key is randomly chosen from the whole key space, we can give a key recovery attack on the full-round Zorro.

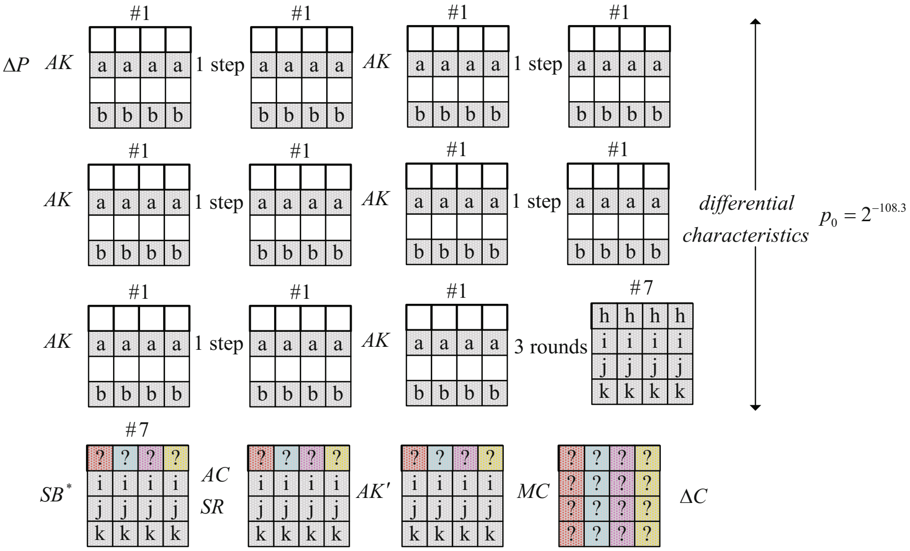
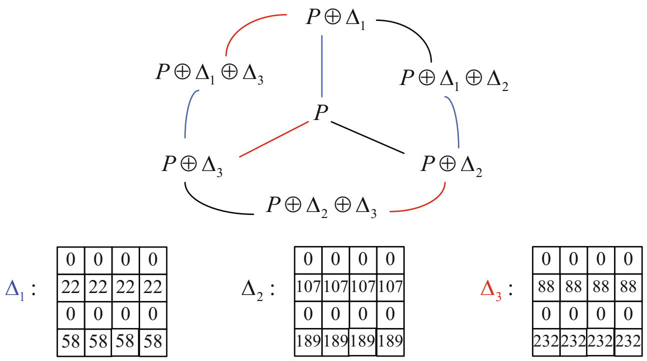


Fig. 2. Key recovery attack on full-round Zorro

Outline. In order to recover the secret key of Zorro efficiently, we combine 3 iterated differential trails to give a structure attack. If we denote the secret key by K, we can change the order of MC and AK in the last round by adding the equivalent key $K^{'} = MC^{-1}(K)$ before MC. Meanwhile, recovering the equivalent key means that the secret key is found. Note that it is impossible to distinguish equivalent keys that share the same values in the last three rows based on the above distinguisher. Therefore, we focus on the 4 bytes of the first row of K'. We first reduce the size of guessing key space from 2^{32} to 1 and then exhaustively search the remaining key candidates for the whole 128-bit key.

1. Choice of Plaintext Pairs
   The chosen plaintexts structure is shown as Figure 3. It is easy to see that in such a structure each difference appears three times. Thus, a total of 9 pairs are contained in a structure of 7 plaintexts. Choose n structures and ask all the $7n$ plaintexts for the corresponding ciphertexts, we can obtain $9n$ plaintext-ciphertext pairs.


$P \oplus \Delta_1$
$P \oplus \Delta_1 \oplus \Delta_3$
$P \oplus \Delta_1 \oplus \Delta_2$
P
$P \oplus \Delta_3$
$P \oplus \Delta_2$
$P \oplus \Delta_2 \oplus \Delta_3$

Δ_1:

0	0	0	0
22	22	22	22
0	0	0	0
58	58	58	58

Δ_2:

0	0	0	0
107	107	107	107
0	0	0	0
189	189	189	189

Δ_3:

0	0	0	0
88	88	88	88
0	0	0	0
232	232	232	232

Fig. 3. Chosen plaintexts structure

2. Filtration of Plaintext-Ciphertext Pairs
   Choose ciphertext pairs so that the differences of the input of 24-round satisfy the condition in #7. About 2^{32} among 2^{128} pairs can satisfy the differential condition. Therefore, it remains about $9n \times 2^{-96}$ plaintext-ciphertext pairs to distinguish the right key from wrong keys.
3. Reduction of Key Candidates in the First Row
   Guess the four bytes of the first row of $K^{'}(2^{32})$, and decrypt the remaining pairs to get the differences of the bytes which fall in the first row of the output of 23-round. If the differences satisfy the condition in the first row of the output of distinguisher, increase the corresponding counter of the guessed key.
4. Extraction from Key Candidates
   Up to now, $9n \times 2^{-96}$ plaintext-ciphertext pairs are left to distinguish the right key from wrong keys. The correct key is suggested with a probability of $2^{-108.3}/2^{-96} = 2^{-12.3}$ while it is about $2^{-128}/2^{-96} = 2^{-32}$ for the incorrect keys. Utilizing the probability differences between the correct key and incorrect keys, we can extract the correct key. We use the ranking paradigm to filter out the key in the first position as the right key candidate.
5. Recovery the Right Key
   Exhaustively test the remaining key candidates(2^{96} keys) to find the correct 128-bit key.

Complexities

1. Data Complexity
   As mentioned in the first step of attack, $7n$ chosen plaintexts are needed to process the attack.
2. Time Complexity
   One computational complexity is checking whether the differences of ciphertext pairs satisfy the differences of last three rows of #7 or not. It can be processed column by column. As is known to us, having known arbitrary 4 bytes in the input and output of MC in AES, the other 4 bytes can be determined. Thus, we can pre-compute all the 2^8 possible outputs of MC with knowing last three input bytes (i, j, k) and store them in a table with the last output byte as the index. Given the difference in the last byte of arbitrary column, the only possible differences in the other three bytes can be obtained from the table. Thus, a pair can be verified after looking up the table at most 4 times, which is much less than 1/4 one-round encryption. Checking all pairs spend about $9n \times 2^{-6.6}$ full-round Zorro encryptions.
   Another computational complexity is incrementing counters for correct key candidates from the tuples of guessed 32-bit keys and plaintext-ciphertext pairs. It is smaller than $9n \times 2^{-96} \times 2^{32}$ one round encryption. Finally, we need about 2^{96} full-round Zorro encryptions to exhaustively test the remaining key candidates.
3. Memory Complexity
   Since attackers must choose the correct key among the 32-bit keys, it is necessary for the attacker to have enough memory for each 2^{32} keys, which is independent of n.

Given the probabilities (p_0, p), the authors provided a general method for finding an accurate number of samples to reach given error probabilities in [8](Algorithm 1 shown in Appendix B), where p(resp. p_0) is the probability suggested for a wrong key(resp. for the right key). We first denote the type-I error probability (the probability to wrongfully discard the right key) with α and the type-II error probability (the probability to wrongfully accept a random key as the right key) with β. In our attack, we want to determine the number of sample $9n \times 2^{-96}$ with $p_0 = 2^{-12.3}$ and $p = 2^{-32}$. If $\alpha = 10\%$ and $\beta = 2^{-32}$, about $2^{16.85}$ samples($9n \times 2^{-96}$ pairs) can reduce 2^{32} keys to 1 candidate. That is to say, the data complexity of our attack is about $2^{112.5}$ chosen plaintexts. Therefore, the number of remaining key candidates for 128-bit key is about 2^{96} and we exhaustively check the key candidates to filter out the right key. All in all, the time complexity is about $2^{112.85} \times 2^{-6.6} + 2^{16.85} \times 2^{32} \times 1/24 + 2^{96} \approx 2^{106}$ full-round Zorro encryptions.

As mentioned before, it is impossible to distinguish the wrong keys that share the same values in the last three rows with the right key based on the above 23-round distinguisher. Thus, the number of key candidates after the distinguishing process is no less than 2^{96}. In other words, the time complexity for searching the right key is 2^{96} full-round encryptions at least no matter how many plaintext-ciphertext pairs are given. In order to reduce the time complexity of key filtering process, we will show a TMTO attack in the next section.

3.3 TMTO Key Recovery Attack on Full-Round Zorro

In this section, three iterated differential characteristics of 22-round Zorro are used to filter out the right key from the whole key space. The 22-round differential characteristics shown in Figure 4 also have the probability of $2^{-108.3}$, where the values of c, d and e are referred to Table 1. With the assumption that the secret key is randomly chosen from the whole key space, we can also give a full-round key recovery attack on Zorro with a less time complexity for key filtering process. We first consider 64-bit equivalent key and then use the ranking paradigm to filter out the correct one as the right 64-bit key candidate. Finally, exhaustively test the remaining 2^{64} key candidates to find the correct 128-bit key.

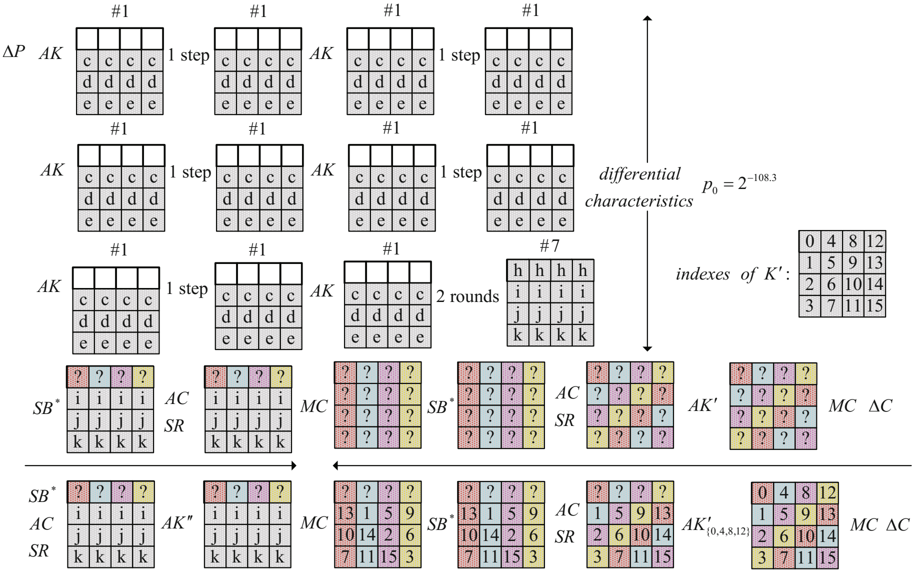


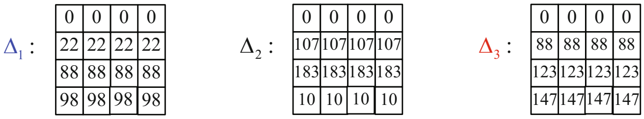
Fig. 4. TMTO key recovery attack on full-round Zorro

Outline. As before, we combine 3 iterated differential trails to give a structure attack and recover the equivalent key $K^{'} = MC^{-1}(K)$ before MC. We divide the 128-bit K' to 16 bytes and denote them as shown in Figure 4. As we know, the key addition can be removed through the linear function with the corresponding operation. Because the S-box layer of Zorro only consists of four S-boxes, we divide the 128-bit K' into two parts, the first row after and the last three rows before the 24-round S-box layer. The following three rows of K' can be removed before the 23-round MC operation and a new 128-bit key $K^{''}$ generated from 12 bytes of K' appears. Meanwhile,

$$K^{''} = \begin{pmatrix} K_0^{''} & K_4^{''} & K_8^{''} & K_{12}^{''} \\ K_1^{''} & K_5^{''} & K_9^{''} & K_{13}^{''} \\ K_2^{''} & K_6^{''} & K_{10}^{''} & K_{14}^{''} \\ K_3^{''} & K_7^{''} & K_{11}^{''} & K_{15}^{''} \end{pmatrix} = MC^{-1} \times \begin{pmatrix} 0 & 0 & 0 & 0 \\ K_{13}^{'} & K_1^{'} & K_5^{'} & K_9^{'} \\ K_{10}^{'} & K_{14}^{'} & K_2^{'} & K_6^{'} \\ K_7^{'} & K_{11}^{'} & K_{15}^{'} & K_3^{'} \end{pmatrix}.$$

The 128-bit $K^{''}$ is independent with $K_{\{0,4,8,12\}}^{'}$ and they together determine the equivalent 128-bit key K'. We can replace the operation $AK^{'}$ by respectively adding $K^{''}$ after the 23-round SR and adding $K_{\{0,4,8,12\}}^{'}$ after the 24-round SR. Similarly, it is impossible to distinguish the keys located in the last three rows of $K^{''}$ based on the distinguisher(Figure 4). As a result, we first use the plaintext-ciphertext pairs and distinguisher to filter out the correct 64-bit equivalent key($K_{\{0,4,8,12\}}^{'}$ and $\mathrm{K}_{\{0,4,8,12\}}^{''}$). Finally, exhaustively test the remaining 2^{64} key candidates to find the right 128-bit key.

1. Choice of Plaintext Pairs
   The chosen plaintexts structure is similar to that of the basic attack. Three kinds of differences are used to construct each structure and their values are given in Figure 5. Thus, we can obtain $9n$ differential pairs with $7n$ plaintext-ciphertext pairs.



Δ_1 :

0	0	0	0
22	22	22	22
88	88	88	88
98	98	98	98

Δ_2 :

0	0	0	0
107	107	107	107
183	183	183	183
10	10	10	10

Δ_3 :

0	0	0	0
88	88	88	88
123	123	123	123
147	147	147	147

Fig. 5. Three differences in the chosen plaintexts structure

2. Filtration of Plaintext-Ciphertext Pairs
   Considering one column MC transformation used in AES, if we have known arbitrary four bytes among the 8 bytes in the input and output, then the other four bytes can be determined with probability 1. Given a ciphertext pair, we can obtain the differences in the last three rows after the MC operation of 23-round with probability 1. Meanwhile, if a pair may suggest some keys, then the differences in the last three rows before the MC are equal to that of the output of the distinguisher. As a result, 6 bytes are known for each column and the matching between four columns occurs with a probability of $2^{-16\times 4} = 2^{-64}$. Choose ciphertext pairs that the differences of the last three rows successfully match between the MC operation of the 23-round. About 1 among 2^{64} pairs can satisfy the above condition. Therefore, it remains about $9n \times 2^{-64}$ plaintext-ciphertext pairs to distinguish the right 64-bit key from wrong keys.

3. Reduction of Key Candidates

   To reduce the time complexity, we compute the values of suggested keys from the remaining ciphertext pairs instead of exhaustively guessing the corresponding keys. The procedure can be described as follows:

   (a) Given a remaining pair, we can easily get the differences before and after the MC of 23-round as explained above. Thus, the input and the output differences of the S-box layers in the 23-round and 24-round are known.
   (b) After looking up the difference table of S-box, we can obtain the corresponding input and output values of the 8 S-boxes(4 in 23-round and 4 in 24-round).
   (c) Up to now, we have known the output values of the 4 S-boxes in the 24-round. Furthermore, we can easily get the suggested values $K^{'}_{\{0,4,8,12\}}$. On the average, only one key is suggested because given the input and output difference of the S-box in Zorro, one solution is averagely obtained.
   (d) Meanwhile, we have known all the values before the 23-round MC and the values in the first row after the 23-round SR. Easily, the possible values of $K^{''}_{\{0,4,8,12\}}$ are also obtained.
   (e) Increase the corresponding counters of the computed 64-bit keys.

   The above steps are repeated at most $9n \times 2^{-64}$ times. If there exists impossible input-output difference pair of S-box in Step (b), skip the following three steps and go to the next remaining pair.
4. Extraction from Key Candidates

   There are $9n \times 2^{-64}$ plaintext-ciphertext pairs to distinguish the right 64-bit key from wrong keys. The incorrect key is suggested with a probability of $2^{-128}/2^{-64} = 2^{-64}$ while it is about $2^{-108.3}/2^{-64} = 2^{-44.3}$ for the right key. We also use the ranking paradigm to filter out the correct key.
5. Recovery the Right Key

   Exhaustively test the remaining 2^{64} key candidates to find the correct 128-bit key.

Similarly, we want to determine the number of samples $9n \times 2^{-64}$ with $p_0 = 2^{-44.3}$ and $p = 2^{-64}$. If $\alpha = 10\%$ and $\beta = 2^{-64}$, about $2^{49.81}$ samples($9n \times 2^{-64}$ pairs) can reduce 2^{64} keys to 1 candidates. That is to say, the data complexity of our attack is about $2^{113.5}$ chosen plaintexts. To clarify the special structure of Zorro, we only focus on the time complexity for searching the right key after filtering out wrong pairs. For a remaining pair, the suggested 64-bit keys can be computed by looking up table 8 times. All in all, it costs much smaller than $9n \times 2^{-64}$ one round encryption to reduce the key space to 2^{64}. Finally, we need about 2^{64} full-round Zorro encryptions to exhaustively test the remaining key candidates. Thus, the time complexity of searching keys is about $2^{49.81} \times 1/24 + 2^{64} \approx 2^{64}$ full-round Zorro encryptions with 2^{64} memory.

4 Linear Distinguishing Attack on Full-Round Zorro

Consider an n-bit block cipher F and let the input of the function be $x \in F_2^n$. A linear approximation (u, v) with an input mask u and an output mask v has probability

$$p(u,v) = Pr_{x \in F_2^n}(u \cdot x \oplus v \cdot F(x) = 0).$$

The value $C_F(u,v) = 2p(u,v) - 1$ is called the correlation of linear approximation (u, v).

Consider a mapping $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$ given as a key-alternating iterative block cipher, i.e. $F = F_r \circ F_{r-1} \circ ... \circ F_1$. A linear trail consists of an input mask u and output mask v and a vector $U = (u_1, ..., u_{r-1})$ with $u_i \in \mathbb{F}_2^n$. The correlation of the trail is defined as

$$C_F(u, v, U) = C_{F_1}(u, u_1) C_{F_2}(u_1, u_2) ... C_{F_{r-1}}(u_{r-2}, u_{r-1}) C_{F_r}(u_{r-1}, v).$$

In contrary to the piling-up lemma[3], no assumption of any kind has to be made for this equation to hold. The characteristics of the correlation matrices of some special boolean functions are summarized as follows[19]:

Lemma 1 *(XOR with a Constant): Consider the function that consists of the bitwise XOR with a constant vector k: $F(x) = x \oplus k$, the correlation matrix is a diagonal matrix with*

$$C_F(u,u) = (-1)^{u^T k}.$$

Lemma 2 *(Linear functions): Consider a linear function $F(x) = Mx$, with M an $m \times n$ binary matrix. The elements of the corresponding correlation matrix are given by*

$$C_F(u,v) = \delta(M^T v \oplus u),$$

where

$$\delta(w) = \begin{cases} 1, & when\ w = 0 \\ 0, & when\ w \neq 0 \end{cases}.$$

Lemma 3 *(Bricklayer Functions): Consider a bricklayer function $y = F(x)$ that is defined by the following component functions: $y_{(i)} = F_{(i)}(x_{(i)})$ for $1 \leq i \leq l$. For every component function $F_{(i)}$ there is a corresponding correlation matrix denoted by $C_{F_{(i)}}$. The elements of the correlation matrix of F are given by*

$$C_F(u,v) = \prod_i C_{F_{(i)}}(u_{(i)}, v_{(i)}),$$

where $u = (u_{(1)}, u_{(2)}, ..., u_{(l)})$ and $v = (v_{(1)}, v_{(2)}, ..., v_{(l)})$.

In this section, we will give a linear distinguishing attack for full-round Zorro according to the above three rules. F represents the 24-round Zorro, and F_i represents the corresponding i-th step function. Note that the fact $M^4 = I$ implies that $(M^T)^4 = I$, where M^T means the transpose of matrix M.

4.1 Iterated Linear Trail

There exists some iterated linear trails for 4 rounds of Zorro and the pattern can also be shown as Figure 1, where the gray bytes are the ones with a non-zero mask. We compute the correlation of the linear trail using the theory of the correlation matrix with $u = v = u_i (i \leq 6)$. There are 255 different (a, b)

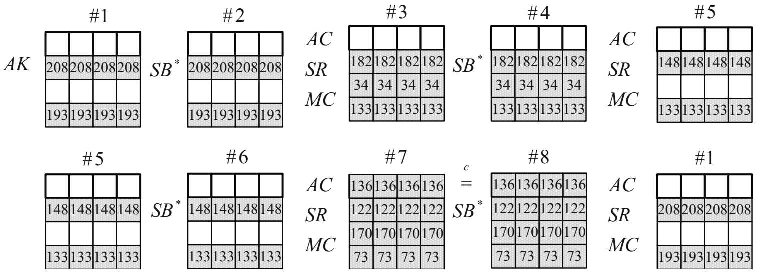


Fig. 6. Iterated linear trail of one-step Zorro

which result in the path from #1 to #7 with the absolute of correlation to be 1. After searching the linear approximation table(LAT) of the S-box used in Zorro, only 210 original linear masks make the path from #7 to #8 with a non-zero correlation. The largest linear correlation occurs when $a = 208$ and $b = 193$ and the absolute value of the corresponding correlation $|c| = (28/128)^4 \approx 2^{-8.77}$. If we change the relative location of #1 with #3, #5 or #7, $|c|$ remains equal. Meanwhile, if the input mask and the output mask of one step are both $(0, 0, 0, 0, 208, 208, 208, 208, 0, 0, 0, 0, 193, 193, 193, 193)$, the linear trail is determined as Figure 6.

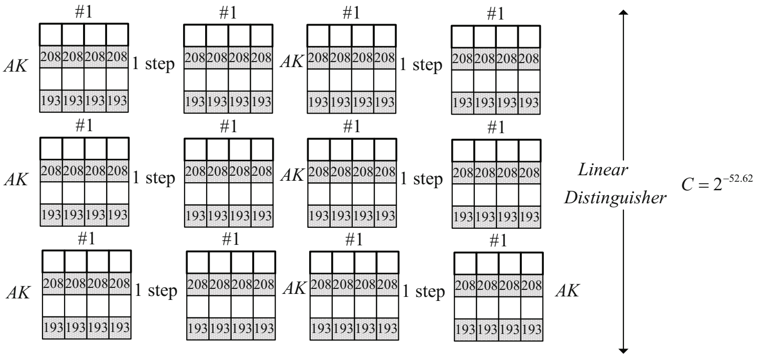


Fig. 7. Linear distinguisher on full-round of Zorro

4.2 Linear Distinguisher of the Full-Round Zorro

If we fix the input linear mask of every step to be the pattern of #1 with $a = 208$ and $b = 193$, we can get a linear trail of full-round Zorro. The absolute value of the correlation of the linear trail can be computed as $|C| = 2^{-8.77\times 6} = 2^{-52.62}$ without any assumption. Thus we can distinguish the full-round Zorro from random permutation by using $1/C^2 \approx 2^{105.3}$ known plaintexts and the distinguisher is shown as Figure 7.

5 Conclusion

In this paper, we evaluated the security of Zorro against differential cryptanalysis and linear cryptanalysis. Two different key recovery attacks were described in Section 3. The basic one recovered the secret key with a data complexity of $2^{112.4}$ chosen plaintexts, a time complexity of 2^{106} full-round Zorro encryptions and a memory complexity of 2^{32}. The TMTO attack required $2^{113.9}$ chosen plaintexts, a key filtering complexity of 2^{64} full-round Zorro encryptions and 2^{64} memory. Meanwhile, we gave a linear distinguishing attack on the full-round Zorro with $2^{105.3}$ known plaintexts.

For convenience, we fix that the differences of four bytes in each row are all the same. If we exhaustively search the characteristics covering three rounds with probability 1, we may obtain some trails for one step of Zorro with a probability higher than $2^{-21.66}$. Thus the complexity of our key recovery attacks can be improved. The similar cases may occur for the linear distinguishing attack. In summary, the results show that only four S-boxes located in the first row and an iterated structure as AES produce a theoretical weak block cipher. Designers should carefully reduce the non-linear operations when designing a lightweight block cipher based on AES block cipher.

Acknowledgments. We thank the anonymous reviewers for their useful comments that help to improve the paper. The research presented in this paper is supported by the National Basic Research Program of China (No. 2013CB338002) and National Natural Science Foundation of China (No. 61272476, No.61232009 and No. 61202420).

References

1. Guo, J., Nikolic, I., Peyrin, T., Wang, L.: Cryptanalysis of Zorro. Cryptology ePrint Archive, Report 2013/713 (2013), `http://eprint.iacr.org/`
2. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
3. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
4. Knudsen, L.R.: Iterative characteristics of DES and s2-DES. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 497–511. Springer, Heidelberg (1993)

5. Knudsen, L.: Truncated and higher order differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer, Heidelberg (1995)
6. Sugita, M., Kobara, K., Imai, H.: Security of reduced version of the block cipher Camellia against truncated and impossible differential cryptanalysis. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 193–207. Springer, Heidelberg (2001)
7. Wang, M.: Differential cryptanalysis of reduced-round Present. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 40–49. Springer, Heidelberg (2008)
8. Blondeau, C., Gérard, B., Tillich, J.P.: Accurate estimates of the data complexity and success probability for various cryptanalyses. Designs, Codes and Cryptography 59(1-3), 3–34 (2011)
9. Biham, E.: On Matsui's linear cryptanalysis. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 341–355. Springer, Heidelberg (1995)
10. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 432–450. Springer, Heidelberg (2004)
11. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: Present: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
12. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — A family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
13. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINT cipher: A block cipher for ic-printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)
14. Wu, W., Zhang, L.: LBlock: A lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)
15. Bilgin, B., Bogdanov, A., Knežević, M., Mendel, F., Wang, Q.: FIDES: Lightweight authenticated cipher with side-channel resistance for constrained hardware. In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 142–158. Springer, Heidelberg (2013)
16. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011)
17. 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)
18. Gérard, B., Grosso, V., Naya-Plasencia, M., Standaert, F.-X.: Block ciphers that are easier to mask: How far can we go? In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 383–399. Springer, Heidelberg (2013)
19. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer-Verlag New York, Inc., Secaucus (2002)

Appendix A: S-box of Zorro

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	B2	E5	5E	FD	5F	C5	50	BC	DC	4A	FA	88	28	D8	E0	D1
10	B5	D0	3C	B0	99	C1	E8	E2	13	59	A7	FB	71	34	31	F1
20	9F	3A	CE	6E	A8	A4	B4	7E	1F	B7	51	1D	38	9D	46	69
30	53	E	42	1B	F	11	68	CA	AA	6	F0	BD	26	6F	0	D9
40	62	F3	15	60	F2	3D	7F	35	63	2D	67	93	1C	91	F9	9C
50	66	2A	81	20	95	F8	E3	4D	5A	6D	24	7B	B9	EF	DF	DA
60	58	A9	92	76	2E	B3	39	C	29	CD	43	FE	AB	F5	94	23
70	16	80	C0	12	4C	E9	48	19	8	AE	41	70	84	14	A2	D5
80	B8	33	65	BA	ED	17	CF	96	1E	3B	B	C2	C8	B6	BB	8B
90	A1	54	75	C4	10	5D	D6	25	97	E6	FC	49	F7	52	18	86
A0	8D	CB	E1	BF	D7	8E	37	BE	82	CC	64	90	7C	32	8F	4B
B0	AC	1A	EA	D3	F4	6B	2C	FF	55	A	45	9	89	1	30	2B
C0	D2	77	87	72	EB	36	DE	9E	8C	DB	6C	9B	5	2	4E	AF
D0	4	AD	74	C3	EE	A6	F6	C7	7D	40	D4	D	3E	5B	EC	78
E0	A0	B1	44	73	47	5C	98	21	22	61	3F	C6	7A	56	DD	E7
F0	85	C9	8A	57	27	7	9A	3	A3	83	E4	6A	A5	2F	79	4F

Appendix B: Computation of the exact number of samples required for a statistical attack

Input: Given error probabilities (α, β) and probabilities (p_0, p).

Output: N and τ : the minimum number of samples and the corresponding relative threshold to reach error probabilities less than (α, β).

```
Set $\tau_{min}$ to $p$ and $\tau_{max}$ to $p_0$.
repeat
    Set $\tau$ to $(\tau_{min} + \tau_{max})/2$.
    Compute $N_{nd}$ such that $\forall N > N_{nd}$, $G_{nd}(N, \tau) \leq \alpha$.
    Compute $N_{fa}$ such that $\forall N > N_{fa}$, $G_{fa}(N, \tau) \leq \beta$.
    if $N_{nd} > N_{fa}$ then
        $\tau_{max} = \tau$.
    else
        $\tau_{min} = \tau$.
    end if
until $N_{nd} = N_f a$.
Return $N = N_{nd} = N_{fa}$ and $\tau$.
```