The ISC Int'l Journal of Information Security

ISeCure

January 2014, Volume 6, Number 1 (pp. 23–34)

http://www.isecure-journal.org

Total Break of Zorro Using Linear and Differential Attacks

Shahram Rasoolzadeh [1,2], Zahra Ahmadian [1,2,*], Mahmoud Salmasizadeh [2], and Mohammad Reza Aref [1]

[1] *Information Systems and Security Lab (ISSL), Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

[2] *Electronic Research Institute, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 24 April 2014
Revised: 15 July 2014
Accepted: 20 August 2014
Published Online: 24 August 2014

Keywords:
Differential Attack, Lightweight Block Cipher, Linear Attack, Zorro.

ABSTRACT

An AES-like lightweight block cipher, namely Zorro, was proposed in CHES 2013. While it has a 16-byte state, it uses only 4 S-Boxes per round. This weak nonlinearity was widely criticized, insofar as it has been directly exploited in all the attacks on Zorro reported by now, including the weak key, reduced round, and even full round attacks. In this paper, using some properties discovered by Wang *et al.* we present new differential and linear attacks on Zorro, both of which recover the full secret key with practical complexities. These attacks are based on very efficient distinguishers that have only two active S-Boxes per four rounds. The time complexity of our differential and linear attacks are $2^{55.40}$ and $2^{45.44}$ and the data complexity are $2^{55.15}$ chosen plaintexts and $2^{45.44}$ known plaintexts, respectively. The results clearly show that the block cipher Zorro does not have enough security against differential and linear attacks.

1 Introduction

Block ciphers are the most widely-studied primitives in the area of symmetric cryptography. Among different types of attacks, differential cryptanalysis [1] and linear cryptanalysis [2] can be regarded as two of the oldest and most important statistical methods to analyze the security of the block ciphers.

Zorro is a newly proposed lightweight block cipher whose design is based on AES [4]. It is basically designed with the aim of increasing the resistance against side-channel attacks, while still remaining a lightweight block cipher. In spite of its 16-byte state, the SubByte layer of Zorro uses only 4 similar S-Boxes in the first row, which are different from AES S-Boxes. Similar to LED-64 [5], key addition layer in Zorro is applied only after each four rounds. Instead, an Add Constant layer is used in every round with round-dependent constants. Besides, Shift Row and Mix Column layers are exactly the same as AES ones.

For both differential and linear cryptanalysis, designers of Zorro have evaluated the security of the cipher and found a balance between the number of inactive S-Boxes and the number of freedom degrees for differential or linear paths. The designers concluded that 14 and 16 rounds are upper bounds for any non-trivial differential or linear characteristics, respectively. Furthermore, they show that in the single key model of Zorro, a 12 round meet-in-the-middle attack is the most powerful attack. Therefore, to meet the security requirements, they choose 24 rounds for Zorro [4].

The main idea in designing Zorro was using the partial nonlinear layers: only 4 S-Boxes for a 16-byte state. That's why Zorro has attracted the attentions

* Corresponding author.
Email addresses: sh_rasoolzadeh@ee.sharif.edu (Sh. Rasoolzadeh), ahmadian@ee.sharif.edu (Z. Ahmadian), salmasi@sharif.edu (M. Salmasizadeh), aref@sharif.edu (M. R. Aref).

ISSN: 2008-2045

of many cryptanalysts during the past year which resulted in some attacks even on the full version of the cipher. The first one, proposed by Guo *et al.* is a key recovery attack on the full-round version of the algorithm, but it works only for 2^{64} weak keys of the whole key space 2^{128} [6]. This attack exploits this unique property of Zorro twice in a two-stage attack: finding an equivalent description that does not have constants in the rounds, and then, launching an internal differential attack.

In the next attack, Wang *et al.* presented a differential key recovery attack and a linear distinguisher for full-round Zorro [7]. They observed an interesting property for the Zorro's MixColumn: the forth power of the mixcolumn matrix is equal to the identity matrix. Using this property of Zorro along with its weak nonlinearity, they found differential and linear distinguishers for Zorro in which only four S-Boxes are activated per four rounds. The resulted differential cryptanalysis can recover the randomly chosen key with a time complexity of 2^{108} and data complexity of $2^{112.4}$ chosen plaintexts, and linear distinguisher use $2^{105.3}$ known plaintexts to successfully distinguish it from the random permutation.

Also, Soleimany proposed a probabilistic variation of slide attack and applied it to 16 rounds of Zorro (out of 24 rounds) [8]. This attack challenges the key schedule approach in Zorro (and also LED [5]) in which all subkeys are equal to the master key of the algorithm, and this similarity is compensated by use of round-dependent constants. Probabilistic slide attack shows that this strategy does not necessarily make the cipher secure against the self-similarity attacks. Their attack requires $2^{123.62}$ known plaintexts with the time complexity of $2^{123.8}$ encryption or $2^{121.59}$ known plaintexts with time complexity of $2^{124.23}$ encryption.

Finally, Bar-On *et al.* briefly reported their new results on Zorro in FSE 2014 rump session which are an improvement of Wang's differential and linear attacks [9]. As they stated, the gain of their attack is not in the probability of distinguishers, since the new distinguishers still have two active S-Boxes per two rounds (i.e. one S-Box per round in average which is similar to that of Wang's attack). Instead, they achieved some improvements in the key recovery phase. Consequently, a differential attack with time and data complexity of 2^{98} and 2^{95}, and a linear attack with time and data complexity of 2^{88} and $2^{83.3}$ can be obtained. As we explain more in the next subsection, they could improve their work further and achieved more efficient distinguishers.

1.1 Our Contributions

In this paper, we break the full-round version of Zorro by using differential and linear cryptanalysis. Alongside the weak nonlinearity of Zorro (i.e. the limited number of S-Boxes in each round), we use the fact discovered in [7] that the forth power of MDS matrix is equal to the identity matrix. We propose very efficient iterated differential characteristics and linear trails that have only two active S-Boxes per four rounds. Using the 23, 22 and 21-round differential characteristics and linear trails, we can propose key recovery attacks for any randomly chosen secret key of full-round Zorro. Differential cryptanalysis has a time complexity of $2^{55.40}$ full round encryption and data complexity of $2^{55.15}$ chosen plaintexts. Also linear cryptanalysis has a time complexity of $2^{45.44}$ full round encryption and data complexity of $2^{45.44}$ known plaintexts. The memory complexity of both differential cryptanalysis and linear cryptanalysis is 2^{17}. Table 1 summarizes the complexities of existing attacks and ours. Our results show that the theoretical security of the full-round Zorro evaluated by designers does not hold up in practice.

We have also simulated our attacks on round-reduced variants of Zorro (up to 16 rounds for differential attack and 20 rounds for linear attack). The simulations results show that the attack complexities and success rate completely coincides the theoretically expected values.

Very recently, some days after that we archived our results on IACR ePrint Archive, Bar-On *et al.* published their improved attacks on Zorro in IACR ePrint Archive [10], that made use of different differential characteristics and linear trails from what they previously announced in FSE 2014 Rump session. It must be mentioned that their linear attack has same time, data and memory complexities as ours, because of using the same linear trails and same key recovery method. Also, their differential attack uses the same differential characteristics as ours. But, by using an improved key recovery method, their differential attack has better time and data complexities.

1.2 Outline

This paper is organized as follows: Section 2 defines some definitions and abbreviations used in the paper. Section 3 presents a brief description of Zorro. Section 4 represents the outline of the differential attack on full-round Zorro with all details and evaluates its complexities. Furthermore, the outline and details of linear attack and evaluation of its complexities are presented in Section 5. Section 6 shows results and the complexity of our practical attacks to Zorro. Finally, Section 7 concludes this paper.

Table 1. Summary of cryptanalytic results on Zorro

Attack Type	Rounds attacked	Time	Data	Memory	Ref.
Differential	Full-round*	$2^{54.3}$	$2^{54.3}$ CP	$2^{54.3}$	[6]
Statistical Slide	16 (out of 24)	$2^{123.8}$	$2^{123.62}$ CP	-	[8]
Statistical Slide	16 (out of 24)	$2^{124.23}$	$2^{121.59}$ CP	-	[8]
Linear (Distinguisher)	Full-round	$2^{105.3}$	$2^{105.3}$ CP	-	[7]
Differential	Full-round	2^{108}	$2^{112.4}$ CP	2^{32}	[7]
Differential	Full-round	2^{98}	2^{95} CP	-	[9]
Linear	Full-round	2^{88}	$2^{83.3}$ KP	2^{80}	[9]
Differential	**Full-round**	$\mathbf{2^{55.40}}$	$\mathbf{2^{55.15}}$ **CP**	$\mathbf{2^{17}}$	**Sec. 4**
Linear	**Full-round**	$\mathbf{2^{45.44}}$	$\mathbf{2^{45.44}}$ **KP**	$\mathbf{2^{17}}$	**Sec. 5**
Differential	Full-round	$2^{45.40}$	$2^{44.40}$ CP	2^{19}**	[10]
Linear	Full-round	2^{45}	2^{45} CP	2^{17}	[10]

* This attack works only for 2^{64} keys of the whole key space 2^{128}.

** The memory complexity is estimated 2^{10} in [10]. However, they need to save DDT with its inputs for every index in searching level for key recovery method.

CP: Chosen Plaintext, KP: Known Plaintext.

2 Definitions and Notations

The main notation and definitions used in the paper are listed as bellow.

- $\mathbf{DP(\alpha \rightarrow \beta)}$ Differential probability of Zorro S-Box with input difference α and output difference β.
- $\mathbf{P_{nr}}$ Differential probability for a n-round differential characteristic of Zorro.
- $\mathbf{P_{PRP}}$ Differential probability for a pseudo random permutation.
- $\mathbf{C(\alpha, \beta)}$ The linear correlation of Zorro S-Box with input mask α and output mask β.
- $\mathbf{c_{nr}}$ Linear correlation for a n-round linear trail of Zorro.
- $\mathbf{c_{PRP}}$ Linear correlation for a pseudo random permutation.

Differential Distribution Table (DDT): For an S-Box, the Differential Distribution Table is a table which the rows represent ΔX values and the columns indicate ΔY values, and each element of the table represents the number of occurrences of the corresponding output difference ΔY value given the input difference ΔX.

Linear Approximation Table (LAT): For an S-Box, the Linear Approximation Table is a table which the rows represent Γ_X values and the columns demonstrate Γ_Y values, and each element of the table represents the number of matches between the linear equation represented as sum of the input bits specified by Γ_X and the sum of the output bits specified by Γ_Y minus 2^{n-1}, where n shows the number of bits for input of S-Box. Hence, dividing an element value by 2^n gives the correlation for the particular linear combination of input and output bits.

Signal to Noise Ratio (SNR or S/N): The Ratio between the number of right pairs and the average count in counting scheme of differential attack is called signal to noise ratio of counting scheme and is denoted by S/N.

3 A Brief Description of Zorro

The block cipher Zorro has a 128-bit key and a 128-bit block size. It has 24 rounds which is divided into 6 steps of 4 rounds each.

As in AES, the internal state in Zorro is a 4×4 matrix of bytes, and every round consists of four transformations:

(1) **SB*** is the S-Box layer where 4 similar S-Boxes, which are different from AES S-Boxes, are applied to the 4 bytes of the first row in the state matrix.

(2) **AC** is adding (XORing) the round constant to the state martix. Specifically, in round i, the four constants $(i, i, i, i \ll 3)$ are XORed to the four bytes of the first row of state matrix. By $\ll$ we mean left shift.

(3) **SR** is similar to AES ShiftRow.
(4) **MC** is similar to AES MixColumn.

The key schedule of Zorro is similar to that of LED block cipher [5]. Before the first and after each step (i.e. each four rounds), the master key is XORed to the state.

As Wang *et al.* argued in [7], by focusing on MC layer used in Zorro, we will see an exclusive feature of this layer. The forth power of MC matrix equals the identity matrix.

$$M = \begin{pmatrix} 02\,03\,01\,01 \\ 01\,02\,03\,01 \\ 01\,01\,02\,03 \\ 03\,01\,01\,02 \end{pmatrix} \Rightarrow M^4 = \begin{pmatrix} 01\,00\,00\,00 \\ 00\,01\,00\,00 \\ 00\,00\,01\,00 \\ 00\,00\,00\,01 \end{pmatrix} \quad (1)$$

Since only 4 S-Boxes are applied to the first row in each round, combined with this feature of MC matrix, iterated differential characteristics and linear trails are found for one step of Zorro.

4 Differential Cryptanalysis

In this section, we first find some iterated differential characteristics for one step of Zorro, which have a high probability. The independence of round functions is a conventional assumption in differential (and linear) cryptanalysis of block ciphers [1, 2]. For Zorro, the secret key is XORed to the state every four rounds. Furthermore, 4 rounds of Zorro can be seen as a step that has no constants in the rounds, if we add one constant to the input and one to the output of the step [6]. Thus, the assumption that the step functions are independent is more rational and realistic than the one which the round functions are independent. Using this assumption, we will construct three groups of distinguishers for 23, 22 and 21 rounds of Zorro. The first distinguisher is used in the first phase of the key recovery attack to reduce the key space of 2^{128} to 2^{96}. Having recovered 32 linear relations between bits of the key in the first phase, we use the second and third distinguishers in the next two phases to recover 64 more relations. Finally, the remaining candidates of key can be retrieved by an exhaustive search.

4.1 Iterated Differential Characteristic

Our strategy to find an efficient iterated differential characteristic for one step of Zorro with the minimum number of active S-Boxes is to exploit the maximum flexibility in the input difference. This is as follows:

- Set the difference of the first row equal to zero to prevent the S-Boxes of the first round being active.
- Set the differences of the third and fourth columns equal to that of the first and second ones, respectively. This bypasses the influence of SR transformation and makes the MC property (1) valid for a 4-round Zorro.
- Do not impose any more conditions on the remaining six bytes now and let their dependency be utilized in minimizing the number of active S-Boxes in the next rounds.

We can extend this input difference to four rounds with only two active S-Boxes as shown in Figure 1. In this figure, the AC transformation is omitted, since it does not have any effect on the differentials. The active S-Boxes are shown in gray whose difference value is written inside. For attaining such a differential characteristic, some conditions in MC transformations between states (#3, #4), (#6, #7), (#12, #1), as well as two conditions for SB^* transformation between states (#10, #11) must be satisfied. All these conditions are presented in detail in Appendix B, which results in the following representation of all the variables based on A and B.

$C = A \oplus B$	$D = A \oplus B$	$E = 2A \oplus B$
$F = A \oplus 2B$	$G = 2A \oplus 3B$	$H = 3A \oplus 2B$
$I = A \oplus 5B$	$J = 5A \oplus B$	$K = 3A \oplus 4B$
$L = 4A \oplus 3B$	$M = A \oplus 8B$	$N = 8A \oplus B$
$O = 13(A \oplus B)$	$P = 13(A \oplus B)$	$Q = 10A \oplus B$
$R = A \oplus 10B$	$S = 20A \oplus 4B$	$T = 4A \oplus 20B$
$U = 6A \oplus 31B$	$V = 31A \oplus 6B$	$W = 17A \oplus 5B$
$X = 5A \oplus 17B$	$Y = 7A \oplus 24B$	$Z = 24A \oplus 7B$

Now, we focus on the SB^* transformation of the fourth round. We need that for all the four active S-Boxes, each output difference equals its own input difference. Suppose this happens with the probability of p. Then,

$$p = DP(S \to S)^2 \times DP(T \to T)^2 \quad (2)$$

We will try to maximize p. Also, we still have 2 degrees of freedom, A and B. So, we can set one of S or T to zero and confine the number of active S-Boxes to two, per four rounds. Let

$$\begin{cases} S = 0 \Rightarrow B = 5A \\ \quad or \\ T = 0 \Rightarrow A = 5B \end{cases} \quad (3)$$

Hence, for the best probability of the proposed 4-round differential characteristic

$$P_{4r} = \max_{1 \le x \le 255} DP(x \to x)^2 \quad (4)$$

According to DDT of S-Box, the maximum probability is equal to $P_{4r} = (6/256)^2 = 2^{-10.83}$ and there are

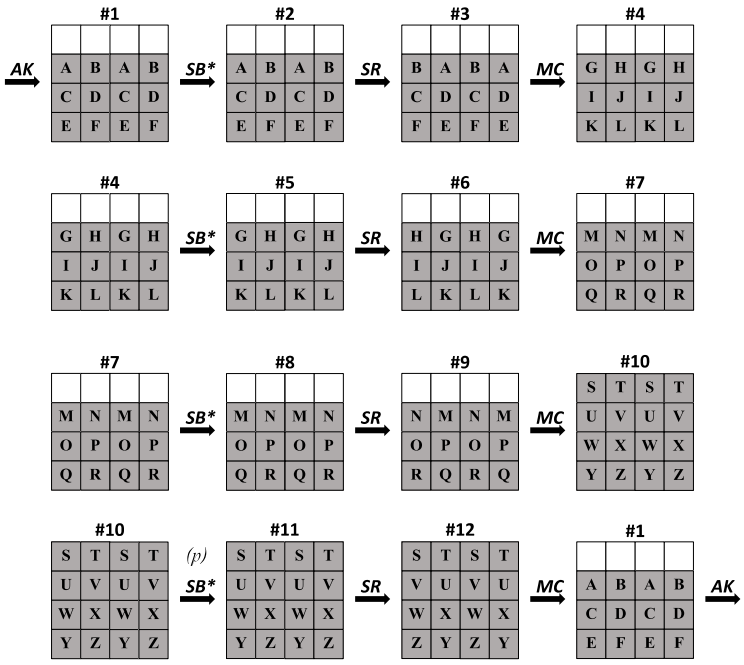

Figure 1. Iterated differential characteristic of one step of Zorro.

three choices for x to achieve this value. Considering the two cases of $S = 0$ or $T = 0$, there would be, in total, six options for the input difference to construct a differential characteristic with this maximum probability. These six differential characteristics are listed in Table 2, in which every row shows the difference values $A, ..., Z$ corresponding to one characteristic. Furthermore, similar to [7], we can replace the difference of state #1 by that of #4, #7 or #10, to get new sets of iterated differential characteristics.

Table 2. Six iterated differential characteristics for one step

Number	A	B	C	D	E	F	G	H	I	J	K	L	M
1	136	158	22	22	149	175	178	164	88	0	205	178	20
2	158	136	22	22	175	149	164	178	0	88	178	205	178
3	92	55	107	107	143	50	225	138	183	0	56	225	255
4	55	92	107	107	50	143	138	225	0	183	225	56	225
5	22	78	88	88	98	138	254	166	123	0	25	254	80
6	78	22	88	88	138	98	166	254	0	123	254	25	254

Number	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	178	254	254	185	51	0	123	85	136	0	35	42	131
2	20	254	254	51	185	123	0	136	85	35	0	131	42
3	225	169	169	89	145	0	234	168	92	0	93	113	228
4	255	169	169	145	89	234	0	92	168	93	0	228	113
5	254	213	213	210	204	0	247	79	22	0	140	168	58
6	80	213	213	204	210	247	0	22	79	140	0	58	168

4.2 Key recovery

The full key recovery attack on full-round Zorro proceeds in three phase. In each phases, we recover 32 linear relations between bits of the secret key.

4.2.1 Phase 1. Recovering 32 Relations Between Bits of Key.

Using each of the six 4-round iterated differentials introduced in Table 2, we can construct a 23-round (= 5 steps + 3 rounds) differential characteristics with probability of

$$P_{23r} = (P_{4r})^5 \times P_{3r} = 2^{-10.83\times 5} \times 1 = 2^{-54.15} \quad (5)$$

Note that, the last three rounds of this characteristics have no cost in probability, i.e. $P_{3r} = 1$. Since P_{23r} is too far from that of a Pseudo Random Permutation, $P_{PRP} = 2^{-128}$, such a 23-round distinguisher can be successfully used to distinguish the correct key from the wrong key in a 24-round attack, as Biham *et al.* thoroughly discussed in [1] for key recovery attack on DES.

In the following, we explain a key recovery attack on full round Zorro which extracts 32 bits information of the secret key K. Similar to [7], a structural attack which merges all the six differential characteristics simultaneously requires less data here. We also change the order of MC and AK in the last round where the equivalent key $K' = MC^{-1}(K)$ is added before MC. In fact, this attack recovers 32 bits of the first row of K', each of which, is a linear function of K, in two

(potentially simultaneous) procedures: In the first one, we find the second and fourth bytes of first row by using iterated differential characteristics respected to No. 1, 3 and 5 of Table 2; In the other one, the first and third bytes are recovered respected to No. 2, 4 and 6 of Table 2. At the end, we will come up with 2^{96} key candidates for the whole 128-bit key.

Step 1. Choosing the Plaintext Pairs
Our Attack is a structural chosen plaintext attack, where we choose some structures and all the plaintexts in every structure are queried from the encryption oracle to get the corresponding ciphertexts. Suppose that we construct M structures which, in total, give N differential pairs with the difference according to #1. The precise relation between M and N can be found in Appendix A and discussed more in Section 4.3.

Step 2. Filtering the Ciphertext Pairs
Partially decrypt all the N ciphertext pairs generated in Step 1 to get their corresponding difference in the output of SB^* of round 24. Keep only those pairs that satisfy the condition in the third row of #10 as well as the two zero differences in the first row (see Figure 2). For a pseudo random permutation, this happens with the probability of 2^{-112}. Whereas, for Zorro this probability is $2^{-54.15}$. Therefore, about $N \times 2^{-54.15}$ pairs of data remain, which can be used to distinguish the right key from the wrong keys.

Step 3. Recovering 16 bits of K'
Guess the two bytes of the first row of K' corresponding to those two active S-Boxes, and partially decrypt the remaining pairs to get their differences in the first row of the input of round 24. If it is consistent with that of #10, increase the corresponding counter of the guessed key. There are $N \times 2^{-54.15}$ differential pairs to distinguish the right key from the wrong keys. An incorrect key is suggested with a probability of 2^{-16} while it is sufficiently high for the right key. Since for each triple of subkeys and a ciphertext pair, which satisfy condition of Step 2, there are in average 7.8^2 key candidates for the desired input/output differences, so S/N ratio for this attack is about $2^{16}/7.8^2 \simeq 2^{10}$ which is significantly high and guarantees that the right key will be suggested with a high probability [1]. Utilizing the probability differences between the correct key and incorrect keys, we can extract the correct candidates for secret key. By this procedure we find two bytes of K' in the first row. A similar procedure can be repeated for the other two active S-Boxes to find the other two bytes in the first row.

4.2.2 Phases 2 & 3. Recovering the 96 Remaining Key Bits.

If we replace the state of #1 by #4 or #7 in Figure 1, we will come up with another 6 iterated differential characteristics, which can be used to construct 22 or 21-round differential characteristics with the same probability of $P_{22r} = P_{21r} = 2^{-54.15}$. So, we need the same number of differential pairs (N) to distinguish the right key from the wrong keys.

The steps of Phase 2 are similar to that of Phase 1 with two minor differences: In Step 2, the ciphertexts differences are filtered based on their partially decrypted values in the output of SB^* transformation in round 23 (rather than 24). Thanks to the 32 bits of K' retrieved in Phase 1, this can be performed. In Step 3, We need to guess 16 bits of K'', where $K'' = MC^{-1}(SR^{-1}(K' \text{ with all bits 0 in the first row}))$.

In this phase, we partially decrypt all the ciphertexts in the structure for one round. But, in AC layer, in addition to round constant, we add bitwisely the first row of K' which was found in Phase 1, and continue the rest of the attack similar to Phase 1. We guess all the 2^{16} keys involved in active S-Boxes, and repeat this procedure once more to get the other 2^{16} key bits. So, we can finally find 32 bits of the first row of K''.

Also in Phase 3, we make use of 21-round differentials and find the third 32 bits of K''', where $K''' = MC^{-1}(SR^{-1}(K'' \{ \text{ with all bits 0 in the first row } \}))$. We do similar to Phase 2, except that at first all the ciphertexts in the structure are partially decrypted for two rounds, and in AC layers, in addition to round constant, we add the first row of K' in round 23, and the first row of K'' in round 22.

Finally, by using the information retrieved from K', K'' and K''', we end up with only 2^{32} candidates for the 128-bit secret key K. With an exhaustive search on these 2^{32} key, we can find the whole 128 bits of secret key.

4.3 Complexities

(1) Data Complexity
For both attacks procedures presented in Phase 1, we need in total $2N$ differential pairs. According to Appendix A, we have $x = 6$ hence, each structure has 2^6 plaintexts and $2N = 6 \times 2^5 M$ where M is the number of structures. So the Data complexity of this phase would be $D_1 = {}^2/_3 \times N \simeq 2^{53.57}$.
The other two phases require also $D_2 = D_3 \simeq 2^{53.57}$ chosen data, so for the full key recovery attack we need about $D = 3 \times 2^{53.57} \simeq 2^{55.15}$ chosen plaintexts.

(2) Time Complexity

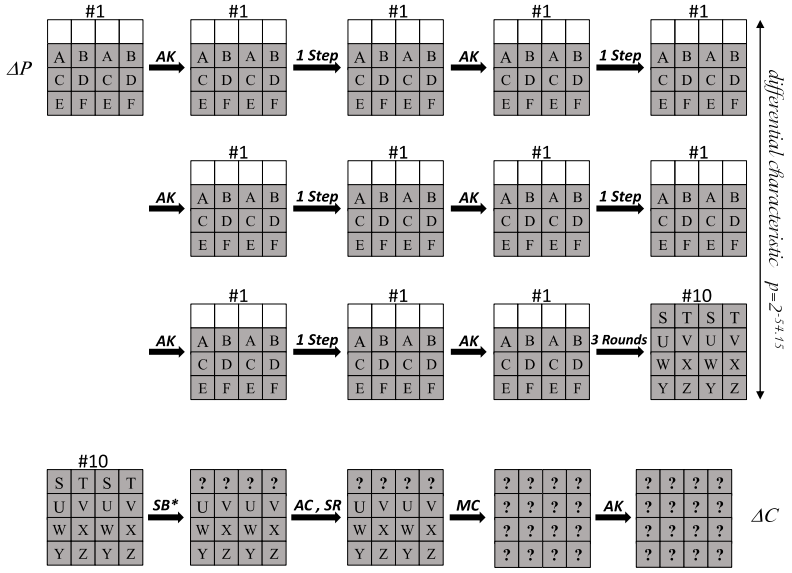

Figure 2. Differential characteristics on 23-round Zorro

For Phase 1, in Step 1 we need to produce the ciphertext for chosen plaintexts that it takes D_1 full-round Zorro, and in Step 2, we need to partially decrypt each remaining pair for less than one round. Therefore, it takes about $N \times 2^{-54.15} \times 1/24$ full-round Zorro encryption. Step 3 requires less than one round encryption for $N \times 2^{-54.15} \times 2^{16}$ times. Thus, the time complexity for finding 32 bits of K' is about

$$T_1 = D_1 + 2 \times N \times 1/24 \times (1 + 2^{-54.15} \times 2^{16}) \simeq D_1 + 1/12 \times N$$

full-round Zorro encryption. As described in [1] and [3], for a differential attack with differential characteristics with probability of p, about c/p differential pairs are needed to distinguish the right key from the wrong keys, where c is a small constant. These all results that N is smaller than $2^{54.15}$ and time complexity is about $T_1 = 2^{53.74}$ full-round Zorro encryption.

Similar to what explained for Phase 1, for the other two phases we have:

$$T_2 = D_2 + N \times 1/24 \times (1 + 2 \times (1 + 2^{-54.15} \times 2^{16})) \simeq D_2 + 1/8 \times N$$
$$T_3 = D_3 + N \times 1/24 \times (2 + 2 \times (1 + 2^{-54.15} \times 2^{16})) \simeq D_3 + 1/6 \times N$$

All in all, the time complexity for the key recovery attack on full-round Zorro would be $T = T_1 + T_2 + T_3 + 2^{32} = D + 3/8 \times N = 2^{55.40}$

(3) Memory Complexity

The memory required for all the three phases of the attack is used to keep the counters of the two 16-bit keys. For the simultaneous attack procedures in three phases, it is $M = 2 \times 2^{16} = 2^{17}$ counters. Note that, the memory required for keeping each structure pairs is negligible. So, the memory complexity is independent of N.

5 Linear Cryptanalysis

The procedure of linear attack is very similar to that of differential attack, presented in Section 4. We first try to find iterated linear trails with a high correlation for one step of the algorithm. Then, we make use of this trail to construct 23, 22 and 21-round linear distinguishers, which are used for a key recovery attack on the full-round Zorro.

5.1 Iterated Linear Trail

Same as the way of finding iterated differential characteristics in Section 4.1, we can find iterated linear trails for Zorro. There exists some iterated linear trials for one step of Zorro, whose patterns are identical to that of differential characteristics given in Figure 1, where the gray bytes are the ones with a non-zero mask. For satisfying conditions of MixColumn transformation between states of (#3, #4) , (#6, #7) and (#12, #1), we use 3 lemmas about the correlation matrixes of boolean functions in [11]. All these conditions are presented in detail in Appendix C, which results in the following representation of all the variables based on Q and R.

$A = 10Q \oplus R$	$B = Q \oplus 10R$	$C = 13Q \oplus R$
$D = 13Q \oplus R$	$E = Q \oplus 8R$	$F = 8Q \oplus R$
$G = 3Q \oplus 4R$	$H = 4Q \oplus 3R$	$I = Q \oplus 5R$
$J = 5Q \oplus R$	$K = 2Q \oplus 3R$	$L = 3Q \oplus 2R$
$M = 2Q \oplus R$	$N = Q \oplus 2R$	$O = Q \oplus R$
$P = Q \oplus R$	$S = 20Q \oplus 4R$	$T = 4Q \oplus 20R$
$U = 7Q \oplus 24R$	$V = 24Q \oplus 7R$	$W = 17Q \oplus 5R$
$X = 5Q \oplus 17R$	$Y = 6Q \oplus 31R$	$Z = 31Q \oplus 6R$

Since the only nonlinear parts involved in this trail are the active S-Boxes of state #10, the absolute correlation $|c|$ of this four round trail is

$$|c| = C(S,S)^2 \times C(T,T)^2 \tag{6}$$

Again, we have 2 degrees of freedom, Q and R to maximize $|c|$. So we can set one of S or T to zero.

$$\begin{cases} S = 0 \Rightarrow R = 5Q \\ \quad or \\ T = 0 \Rightarrow Q = 5R \end{cases} \tag{7}$$

which in two cases yields

$$|c_{4r}| = \max_{1 \le x \le 255} C(x,x)^2. \tag{8}$$

After searching the LAT of Zorro S-box, the largest linear correlation occurs when $x = 136$. With this setting the absolute of the corresponding correlation would be $|c_{4r}| = (28/128)^2 \simeq 2^{-4.39}$. Also, we can find new linear trails with the same correlation, if we change the relative location of #1 with #4, #7 or #10. In Table 3, each row shows the mask values $A, ..., Z$ corresponding to one of the above-mentioned linear trail.

Table 3. Two iterated linear trails for one step

Number	A	B	C	D	E	F	G	H	I	J	K	L	M
1	177	97	227	227	191	126	130	126	34	0	126	251	160
2	97	177	227	227	126	191	126	130	0	34	251	126	52

Number	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	52	133	133	234	234	0	136	95	37	0	170	163	234
2	152	133	133	234	234	136	0	37	95	170	0	234	163

5.2 Key Recovery

Similar to that of differential attack, the full key recovery attack on full-round Zorro proceeds in three phase. In each phases, we recover 32 linear relations between bits of the secret key.

5.2.1 Phase 1. Recovering the 32 Bits of Key.

Using each of the two 4-round iterated linear trails in Table 3, we can construct a 23-round (= 5 steps + 3 rounds) linear trail with the correlation of

$$|c_{23r}| = |c_{4r}|^5 \times |c_{3r}| = 2^{-4.39 \times 5} = 2^{-21.93} \tag{9}$$

This 23-round linear trail is similar to the 23-round differential chararcteristic given in Figure 2 Since $|c_{23r}|$ is much larger than that of a Pseudo Random Permutation, $|c_{PRP}| = 0$, such a 23-round distinguisher can be successfully used to distinguish the correct key from the wrong key in a 24-round attack, as discussed thoroughly by Matsui in [2] for cryptanalysis of DES.

In the following, we explain a key recovery attack on full round Zorro which extracts 32 bits of the first row of K', in two sequential procedures: First, we find the second and fourth bytes of the first row of K' by using iterated linear trails respected to No. 1 of Table 3. Then, first and third bytes of key respected to No. 2 of Table 3 gets found.

With the assumption that the secret key is randomly chosen from the whole key space, the amount of plaintext/ciphertext pairs required for this attack would be $N_L = 1/|c_{23r}|^2 \simeq 2^{43.85}$ as discussed in [2] and [3]. The steps of this phase of attack are as follows:

Step 1. Data Collection
Ask the corresponding ciphertexts of N_L randomly generated plaintexts from the encryption oracle.

Step 2. Data Processing
Compute

$$\alpha = \Gamma_{\#1} \cdot P \oplus \Gamma_{\#10, rows\ 2,3,4} \cdot C'_{rows\ 2,3,4} \tag{10}$$

where P is the plaintext, C' is the one-round partially decrypted ciphertext, $\cdot$ represent the dot product, and $\Gamma_{\#n}$ is the linear mask for state $\#n$ in No.1 linear trail given in Table 3.

Step 3. Recovering the second and fourth bytes of K'
Guess the second and fourth bytes of K', partially decrypt the ciphertext to get the first row of C' for every 2^{16} guesses. Compute

$$\beta = \Gamma_{\#10, row\ 1} \cdot C'_{row\ 1} \tag{11}$$

If $\alpha = \beta$, increase the counter of the corresponding guessed key.

Step 4. Recovering the first and third bytes of K'
Repeat Steps 2 and 3 for these two bytes of key.

At the end of this procedure, all the four bytes of K''s first row are introduced.

In Step 3 we use a matrix with size of 256×256, and index of (i, j) matrix shows the sum of mask for S-Box input which its output equals to bitwisely sum of i and j. For each active S-Box we take the x-th

row of the matrix, that x is equal to output of S-Box in partially decrypted ciphertext. We have only two active S-Boxes, So, the first arrow is for 8 bits of key for the first active S-Box, and the second arrow is for 8 bits of key for the second active S-Box. In each arrow j'th bit shows sum of mask for S-Box input which output is j bitwisely added to partially decrypted ciphertext. With this method, we can check for all 2^{16} keys, whether β equals to α or not, with a negligible time for each pliantext-ciphertext pairs.

5.2.2 Phases 2 & 3. Recovering the 96 Remaining Key Bits.

Look like full-key recovery attack in Phase 2 and 3 of differential cryptanalysis , we use 22 and 21-round linear distinguishers with $c_{22r} = c_{21r} = 2^{-21.93}$, which works with an amount of $N_L = 2^{43.85}$ known plaintexts. After reducing the key candidates to 2^{32}, we perform an exhaustive search on the key candidates to get the secret key.

5.2.3 Complexities

(1) Data Complexity
As mentioned before, for each phase we need about $N_L \simeq 2^{43.85}$ known plaintexts.

(2) Time Complexity
We actually separated Steps 2 and 3 to avoid some unnecessary repetitions in attack computations in practice. Though this two steps have a negligible time in total, compared to Step 1, which must produce ciphertext for a random plaintext. So, time complexity for any of these phases equals to $T_1 = T_2 = T_3 = N_L \simeq 2^{45.44}$.

(3) Memory Complexity
Since, the procedure of recovering the two 16 bits of first row of K' are performed in parallel, it is necessary to have enough memory for each 2×2^{16} keys, which is independent of N_L. Another memory complexity is to saving 256×256 bits matrices. All needed memory is equal to 2^{17} counters.

All in all, the time, data and memory complexity for the proposed key recovery attack on full-round Zorro are $2^{45.44}, 2^{45.44}$, and 2^{17}, respectively.

6 Practical Results

We have experimentally verified the efficiency of the proposed attacks by simulating some variants by a C++ code. As described in Section 3 and Section 4, the complete key is recovered in 3 phases, in each phase we find 32 linear equations, and then find the right key from 2^{32} remaining candidates with an exhaustive search. We precisely implemented the 3 phases of the attack, excluding the exhaustive search of the last 2^{32} remaining candidates.

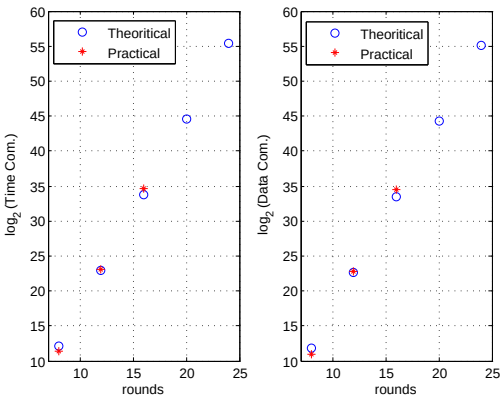

Figure 3. Theoretical and practical results for differential cryptanalysis

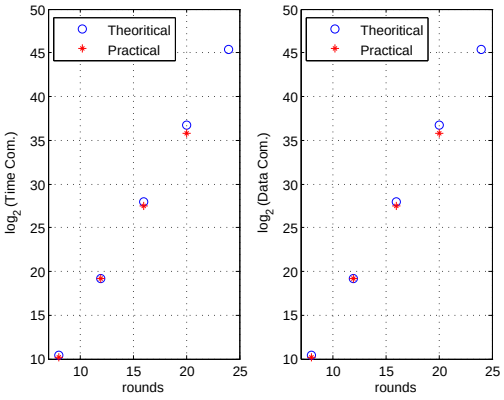

Figure 4. Theoretical and Practical results for linear cryptanalysis

In particular, this attack can be well regarded as the first successful practical attack on full-round Zorro. We used a PC with an Intel(R) Core(TM) i7 CPU Q740 at 1.73GHz, and with 4GB of RAM. Our results show that the proposed attack on round-reduced variants of Zorro works and recovers the correct key as expected theoretically. In Figure 3 and Figure 4, we report some results of our program for both differential and linear attacks on reduced to $r = 8$, 12, 16 and 20 rounds of Zorro. In our calculations, the time for running a r-round Zorro is taken as the unit of time complexity for a r-round attack. Figure 3 compares the theoretical and practical results of our differential attack, while Figure 4 is for linear attack.

7 Conclusion and Future Work

In this paper, we presented an approach to break the full-round version of Zorro by using differential and linear cryptanalysis with practical complexities. These attacks work for all the key space and make use of 23, 22 and 21-round differential characteristics or linear trails. While differential cryptanalysis has a time complexity

of $2^{55.40}$ full round encryption and data complexity of $2^{55.15}$ chosen plaintexts, linear cryptanalysis has a time complexity of $2^{45.44}$ full round encryption and data complexity of $2^{45.44}$ known plaintexts. Some reduced-round variants of both attacks have been simulated which absolutely validates the theoretically estimated complexities.

As far as we know, this is the first practical attack on full-round Zorro, which along with the previous cryptanalyses shows that the partial nonlinearity in the design of Zorro has obviously sacrificed the security for efficiency.

8 Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable and constructive comments. This work was partially supported by Iranian National Science Foundation (INSF) under contract no. 92/32575 and INSF cryptography chair and by the office of Vice-President for the Science and Technology, I. R. Iran.

References

[1] Eli Biham and Adi Shamir. *Differential cryptanalysis of DES-like cryptosystems.* In Alfred J. Menezes and Scott A. Vanstone, editors, Advances in Cryptology-CRYPTO 1990, volume 537 of Lecture Notes in Computer Science, pages 221. Springer Berlin Heidelberg, 1991.

[2] Mitsuru Matsui. *Linear cryptanalysis method for DES cipher.* In Tor Helleseth, editor, Advances in Cryptology-EUROCRYPT 1993, volume 765 of Lecture Notes in Computer Science, pages 386-397. Springer Berlin Heidelberg, 1994.

[3] Howard M. Heys, *A Tutorial on Linear and Differential Cryptanalysis.* Technical Report CORR 2001-17, Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, Mar. 2001.

[4] Benoit Gerard, Vincent Grosso, Maria Naya-Plasencia, and Francois-Xavier Standaert. *Block ciphers that are easier to mask: How far can we go?* In Guido Bertoni and Jean-Sbastien Coron, editors, Cryptographic Hardware and Embedded Systems (CHES) 2013, volume 8086 of Lecture Notes in Computer Science, pages 383-399. Springer Berlin Heidelberg, 2013.

[5] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. *The LED block cipher.* In Bart Preneel and Tsuyoshi Takagi, editors, Cryptographic Hardware and Embedded Systems-CHES 2011, volume 6917 of Lecture Notes in Computer Science, pages 326-341. Springer Berlin Heidelberg, 2011.

[6] Jian Guo, Ivica Nikolic, Thomas Peyrin, and Lei Wang. *Cryptanalysis of Zorro.* Cryptology ePrint Archive, Report 2013/713, 2013. http://eprint.iacr.org/

[7] Yanfeng Wang, Wenling Wu, Zhiyuan Guo, and Xiaoli Yu. *Differential Cryptanalysis and Linear Distinguisher of Full-Round Zorro.* Cryptology ePrint Archive, Report 2013/713, 2013. http://eprint.iacr.org/

[8] Hadi Soleimany. *Probabilistic Slide Cryptanalysis and Its Applications to LED-64 and Zorro.* In the proceeding of the 21st International Workshop on Fast Software Encryption.

[9] Achiya Bar-On, Itai Dinur, Orr Dunkelman, Nathan Keller, Virginie Lallemand, Maria Naya-Plasencia, Boaz Tsaban and Adi Shamir. *New Results on Zorro.* In the rump session of the 21st International Workshop on Fast Software Encryption. http://fse.2014.rump.cr.yp.to/

[10] Achiya Bar-On, Itai Dinur, Orr Dunkelman, Virginie Lallemand and Boaz Tsaban. *Improved Analysis of Zorro-Like Ciphers.* Cryptology ePrint Archive, Report 2014/228, 2014. http://eprint.iacr.org/

[11] Joan Daemen and Vincent Rijmen. *The Design of Rijndael.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

Appendix A. Structural Chosen Plaintext

Assume that we have x differential characteristics and we are going to choose minimum number of plaintexts that provide enough pairs for these x differential characteristics. Let's define a graph in which the vertexes are the plaintexts and the edges are the valid differential pairs. For any node we have x edges and the number of nodes are 2^x. So, we have $x \times 2^{x-1}$ differential plaintext pairs, in total. Thus, the ratio of the chosen plaintexts to the differential plaintext pairs in a stracture is $2/x$. This method is an extension of what was proposed in [1] for generating data.

Appendix B. Differential Characteristic Conditions

The conditions that must be satisfied for the differential characteristic are formulated as below.

The condition for MC transformation between states (#3, #4) results:

$$\begin{cases} E = 3A \oplus D \\ F = 3B \oplus C \end{cases} \Rightarrow \begin{cases} G = B \oplus 2C \\ H = A \oplus 2D \\ I = 4B \oplus C \\ J = 4A \oplus D \\ K = 7B \oplus 3C \\ L = 7A \oplus 3D \end{cases} \quad (12)$$

The condition for MC transformation between states (#6, #7) results:

$$\begin{cases} K = 3G \oplus J \\ L = 3H \oplus I \end{cases} \Rightarrow \begin{cases} M = H \oplus 2I \\ N = G \oplus 2J \\ O = 4H \oplus I \\ P = 4G \oplus J \\ Q = 7H \oplus 3I \\ R = 7G \oplus 3J \end{cases} \quad (13)$$

Also after MC transformation between states (#9, #10), we have:

$$\begin{aligned} S &= 3N \oplus O \oplus R & T &= 3M \oplus P \oplus Q \\ U &= 2N \oplus 3O \oplus R, & V &= 2M \oplus 3P \oplus Q \\ W &= N \oplus 2O \oplus 3R, & X &= M \oplus 2P \oplus 3Q \\ Y &= N \oplus O \oplus 2R, & Z &= M \oplus P \oplus 2Q \end{aligned} \quad (14)$$

Hence, after combining mentioned conditions with each other and some simplifications, we can represent all the variables based on A and B:

$$\begin{aligned} C &= D = A \oplus B \\ E &= 2A \oplus B & F &= A \oplus 2B \\ G &= 2A \oplus 3B & H &= 3A \oplus 2B \\ I &= A \oplus 5B & J &= 5A \oplus B \\ K &= 3A \oplus 4B & L &= 4A \oplus 3B \\ M &= A \oplus 8B & N &= 8A \oplus B \\ O &= P = 13(A \oplus B) \\ Q &= 10A \oplus B & R &= A \oplus 10B \\ S &= 20A \oplus 4B & T &= 4A \oplus 20B \\ U &= 6A \oplus 31B & V &= 31A \oplus 6B \\ W &= 17A \oplus 5B & X &= 5A \oplus 17B \\ Y &= 7A \oplus 24B & Z &= 24A \oplus 7B \end{aligned} \quad (15)$$

Using equations in (15) we see that condition for MC transformation between states (#12, #1) are automatically satisfied.

Appendix C. Linear Trail Conditions

The conditions that must be satisfied for the linear trail are formulated as below.

The condition for MC transformation between states (#3, #4) results:

$$\begin{cases} G = I \oplus 3K \\ H = J \oplus 3L \end{cases} \Rightarrow \begin{cases} A = 3J \oplus 7L \\ B = 3I \oplus 7K \\ C = I \oplus 4K \\ D = J \oplus 4L \\ E = 2J \oplus L \\ F = 2I \oplus K \end{cases} \quad (16)$$

The condition for MC transformation between states (#6, #7) results:

$$\begin{cases} M = O \oplus 3Q \\ N = P \oplus 3R \end{cases} \Rightarrow \begin{cases} G = 3P \oplus 7R \\ H = 3O \oplus 7Q \\ I = O \oplus 4Q \\ J = P \oplus 4R \\ K = 2P \oplus R \\ L = 2O \oplus Q \end{cases} \quad (17)$$

Also after MC transformation between states (#9, #10), we have:

$$\begin{cases} W = 2S \oplus U \oplus 3Y \\ X = 2T \oplus V \oplus 3Z \end{cases} \Rightarrow \begin{cases} M = T \oplus 3V \oplus 2Z \\ N = S \oplus 3U \oplus 2Y \\ O = 5S \oplus U \oplus 7Y \\ P = 5T \oplus V \oplus 7Z \\ Q = 7T \oplus 2V \oplus 7Z \\ R = 7S \oplus 2U \oplus 7Y \end{cases} \quad (18)$$

Hence, after combining mentioned conditions with each other and some simplifications, we can represent all the variables based on Q and R:

$A = 10Q \oplus R$ $B = Q \oplus 10R$

$C = D = 13\ Q \oplus R$

$E = Q \oplus 8R$, $F = 8Q \oplus R$

$G = 3Q \oplus 4R$, $H = 4Q \oplus 3R$

$I = Q \oplus 5R$, $J = 5Q \oplus R$

$K = 2Q \oplus 3R$, $L = 3Q \oplus 2R$

$M = 2Q \oplus R$, $N = Q \oplus 2R$

$O = P = Q \oplus R$

$S = 20Q \oplus 4R$, $T = 4Q \oplus 20R$

$U = 7Q \oplus 24R$, $V = 24Q \oplus 7R$

$W = 17Q \oplus 5R$, $X = 5Q \oplus 17R$

$Y = 6Q \oplus 31R$, $Z = 31Q \oplus 6R$

Using equations in (8) we see that condition for MC transformation between states (#12, #1) are automatically satisfied.

Shahram Rasoolzadeh received his B.S. degree from University of Tabriz, Tabriz, Iran, in 2013, in electrical engineering (communications). He is currently working toward his M.S. degree in Cryptography at Electrical Engineering department of Sharif University of Technology, Tehran, Iran. He serves as a member of Electronic Research Institute and Information Systems and Security Lab. (ISSL) at the Electrical Engineering Department of Sharif University of Technology. His research interests include Cryptography, Network Security, and Signal Processing.

Zahra Ahmadian received the B.S. degree in electrical engineering (communications and electronics) from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2006, and the M.S. degree in electrical engineering (secure communications) from Sharif University of Technology, Tehran, Iran, in 2008. She is currently working toward the Ph.D. degree in electrical engineering (communication systems) at Sharif University of Technology. Her special fields of interest include Wireless Security and Cryptology with an emphasis on Cryptanalysis.

Mahmoud Salmasizadeh received the B.S. and M.S. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1972 and 1989, respectively. He also received the Ph.D. degree in information technology from Queensland University of Technology, Australia, in 1997. Currently, he is an associate professor in the Electronics Research Institute and adjunct associate professor in the Electrical Engineering Department, Sharif University of Technology. His research interests include Design and Cryptanalysis of cryptographic algorithms and protocols, E-commerce Security, and Information Theoretic Secrecy. He is a founding member of Iranian Society of Cryptology.

Mohammad Reza Aref received the B.S. degree in 1975 from the University of Tehran, Iran, and the M.S. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in electrical engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of Electrical Engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 290 technical papers in communications, information theory and cryptography in international journals and conferences proceedings. At the same time, during his academic activities, he has been involved in different political positions. First Vice President of I.R. Iran, Vice President of I.R. Iran and Head of Management and Planning Organization, Minister of ICT of I.R. Iran, and Chancellor of University of Tehran, are the most recent ones. His current research interests include areas of Communication Theory, Information Theory, and Cryptography.