Differential Fault Analysis on Block Cipher Piccolo

Kitae Jeong

Center for Information Security Technologies(CIST), Korea University, Korea
kite.jeong@gmail.com

Abstract. Piccolo is a 64-bit block cipher suitable for the constrained environments such as wireless sensor network environments. In this paper, we propose differential fault analysis on Piccolo. Based on a random byte fault model, our attack can recover the secret key of Piccolo-80 by using an exhaustive search of 2^{24} and six random byte fault injections on average. It can be simulated on a general PC within a few seconds. In the case of Piccolo-128, we require an exhaustive search of 2^{40} and eight random byte fault injections on average. This attack can be simulated on a general PC within one day. These results are the first known side-channel attack results on them.

Keywords: Block cipher, Piccolo, Differential fault analysis.

1 Introduction

Recently, the research on ultra-lightweight block ciphers suitable for the efficient implementation in constrained hardware environments such as RFID tags and sensor nodes has been studied. As a result, KATAN/KTANTAN [2], PRINTcipher [7], LED [4] and Piccolo [9] were proposed.

A 64-bit block cipher Piccolo proposed in CHES 2011 supports 80- and 128-bit secret keys. According to the length of the secret key, they are denoted by Piccolo-80 and Piccolo-128, respectively. The number of rounds of Piccolo-80 and Piccolo-128 is 25 and 31, respectively. The iterative structure of Piccolo is a variant of generalized Feistel network. Until now, several cryptanalytic results on them were proposed. First, the designers of them evaluated the security of Piccolo by various attacks and attacked Piccolo-80 to 17 rounds and Piccolo-128 to 21 rounds by using related-key attacks [9]. The best result of actual single-key attack is 3-Subset Meet-in-the-Middle(MITM) attacks on a 14-round reduced Piccolo-80 and a 21-round reduced Piccolo-128 without whitening keys. On the other hand, Wang et al. introduced a biclique cryptanalysis of the full round Piccolo-80 without postwhitening keys and a 28-round Piccolo-128 without prewhitening keys [11]. These attacks are respectively with data complexity of 2^{48} and 2^{24} chosen ciphertexts, and with time complexity of $2^{78.95}$ and $2^{126.79}$ encryptions. To our knowledge, there is no cryptanalytic results on them based on side-channel attacks.

Differential fault analysis (DFA), one of the side channel attacks, was first proposed by Biham and Shamir on DES in 1997 [1]. This attack exploits faults within the computation of a cryptographic algorithm to reveal the secret information. So far, DFAs on many block ciphers such as AES [10], ARIA [8], SEED [6], CLEFIA [3] and LED [5] have been proposed. It means that DFA poses a major threat to the security on block ciphers.

In this paper, we propose a differential fault analysis on Piccolo. Our attack is based on the random byte fault model. To recover the secret key of Piccolo-80, it is assumed that several random byte faults are injected to the input register of round 23. Thus, the number of possible fault positions is 8. We can compute the exact fault position by checking the corresponding ciphertext differences. As simulation results, this attack requires an exhaustive search of 2^{24} and six random byte faults on average. It can be simulated on a general PC within a few seconds. Similarly, we can the secret key of Piccolo-128 with an exhaustive search of 2^{40} and eight random byte fault injections on average. As simulation results, this attack can be simulated on a general PC within one day. These results are the first known side-channel attack results on them.

This paper is organized as follows. In Section 2, we briefly introduce the structure of Piccolo. Our attacks on Piccolo-80/128 are presented in Section 3 and Section 4, respectively. Finally, we give our conclusion in Section 5.

2 Description of Piccolo

In this section, we briefly present the structures of Piccolo-80 and Piccolo-128. Throughout this paper, the following notations are used.

- $P = (P_0, P_1, P_2, P_3)$: a 64-bit plaintext.
- $C = (C_0, C_1, C_2, C_3)$: a 64-bit ciphertext.
- $I_r = (I_{r,0}, I_{r,1}, I_{r,2}, I_{r,3})$: a 64-bit input value of round r.
- (rk_{2r}, rk_{2r+1}): a round key of round r.
- (wk_0, wk_1, wk_2, wk_3): a whitening key.

Piccolo-80/128 is a 64-bit block cipher and supports 80- and 128-bit secret keys. As shown in Figure 1, the structure of Piccolo-80/128 is a variant of generalized Feistel network. Here, the number of rounds r is 25 for Piccolo-80 and 31 for Piccolo-128. First, with a 64-bit plaintext $P = (P_0, P_1, P_2, P_3)$ and a prewhitening key (wk_0, wk_1), the input value $I_0 = (I_{0,0}, I_{0,1}, I_{0,2}, I_{0,3})$ of round 0 is computed as follows.

$$I_{0,0} = P_0 \oplus wk_0,\ I_{0,1} = P_1,\ I_{0,2} = P_2 \oplus wk_1,\ I_{0,3} = P_3.$$

To generate I_{i+1} from I_i ($i = 0, \cdots, r-2$), each round is made up of a function $F : \{0,1\}^{16} \rightarrow \{0,1\}^{16}$ and a round permutation $RP : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$. Figure 2 presents the structure of F function. Since our attack do not use the property of 4×4 S-box S and 4×4 matrix M, we omit the descriptions of them in this paper. See [9] for the detailed descriptions of them. As shown in Figure

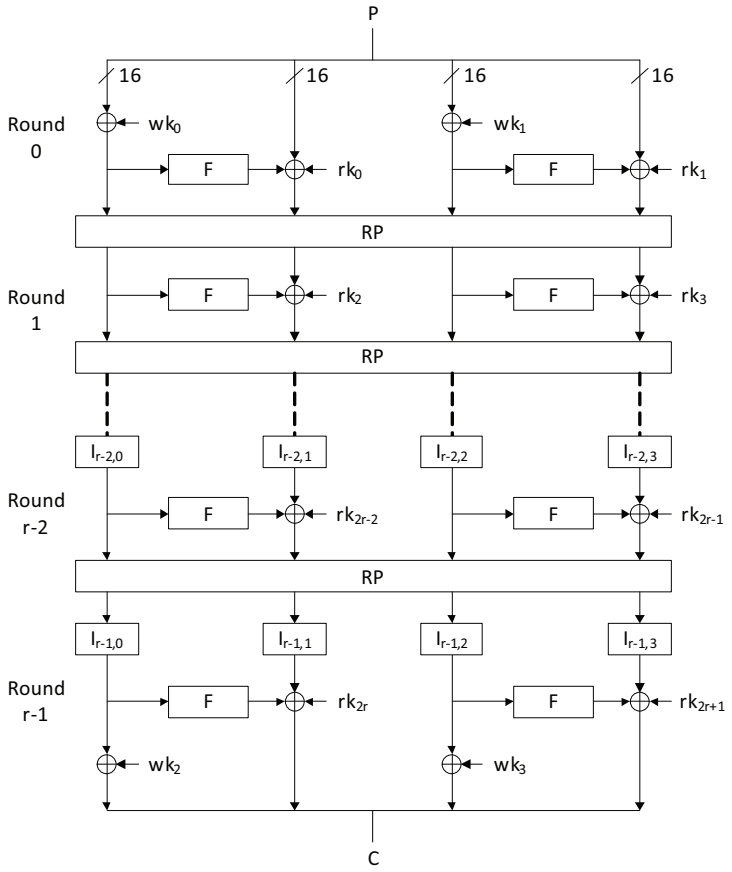

Fig. 1. The structure of Piccolo.

3, a round permutation RP takes a 64-bit input value $X = (x_0, x_1, x_2, x_3)$ and generates a 64-bit output value $Y = (y_0, y_1, y_2, y_3)$. Here, a 16-bit x_i is divided into (x_i^L, x_i^R).

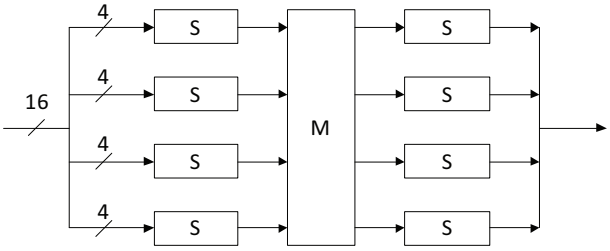

Fig. 2. F function of Piccolo.

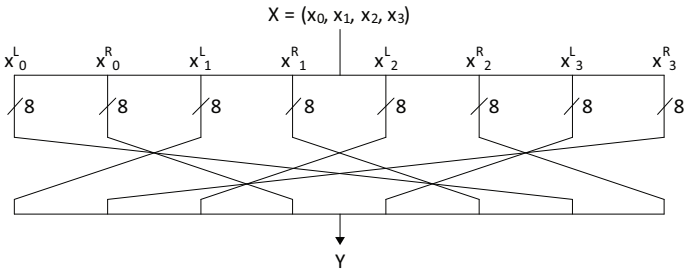

Fig. 3. Round permutation RP of Piccolo.

A 64-bit ciphertext $C = (C_0, C_1, C_2, C_3)$ is generated as follows.

$$C_0 = I_{r-1,0} \oplus wk_2, \quad C_1 = F(I_{r-1,0}) \oplus I_{r-1,1} \oplus rk_{2r},$$
$$C_2 = I_{r-1,2} \oplus wk_3, \quad C_3 = F(I_{r-1,2}) \oplus I_{r-1,3} \oplus rk_{2r+1}.$$

The keyschedule of Piccolo-80 is simple. First, the 80-bit secret key K is computed as follows. Here, $k_j = (k_j^L, k_j^R)$ $(j = 0, 1, 2, 3, 4)$.

$$K = (k_0, k_1, k_2, k_3, k_4).$$

Four whitening keys (wk_0, wk_1, wk_2, wk_3) and 25 round keys (rk_{2i}, rk_{2i+1}) are generated as follows $(i = 0, 1, \cdots, 24)$. Here, $(con_{2i}^{80}, con_{2i+1}^{80})$ is a 16-bit round constant. See See [9] for the detailed descriptions of them.

– Whitening key

$$wk_0 = k_0^L \| k_1^R, \quad wk_1 = k_1^L \| k_0^R,$$
$$wk_2 = k_4^L \| k_3^R, \quad wk_3 = k_3^L \| k_4^R.$$

– Round key

$$(rk_{2i}, rk_{2i+1}) = (con_{2i}^{80}, con_{2i+1}^{80}) \oplus \begin{cases} (k_2, k_3), (i \mod 5) \equiv 0 \text{ or } 2, \\ (k_0, k_1), (i \mod 5) \equiv 1 \text{ or } 4, \\ (k_4, k_4), (i \mod 5) \equiv 3. \end{cases}$$

Table 1. The partial secret key used in each round key

Piccolo-80		Piccolo-128	
Round i	Partial secret key	Round i	Partial secret key
0	(k_2, k_3)	0	(k_2, k_3)
1	(k_0, k_1)	1	(k_4, k_5)
$\vdots$	$\vdots$	$\vdots$	$\vdots$
22	(k_2, k_3)	28	(k_0, k_7)
23	(k_4, k_4)	29	(k_6, k_3)
24	(k_0, k_1)	30	(k_2, k_5)

Table 1 presents the partial secret key used in each round key. For example, the round key (rk_{48}, rk_{49}) of round 24 includes the partial secret key (k_0, k_1).

The keyschedule of Piccolo-128 is similar to that of Piccolo-80. By using the 128-bit secret key $K = (k_0, k_1, \cdots, k_7)$, four whitening keys and 31 round keys are generated as follows (See Table 1).

– Whitening key

$$wk_0 = k_0^L \| k_1^R, \quad wk_1 = k_1^L \| k_0^R,$$
$$wk_2 = k_4^L \| k_7^R, \quad wk_3 = k_7^L \| k_4^R.$$

– Round key $(i = 0, 1, \cdots, 61)$
  - if $((i+2) \mod 8 \equiv 0)$ then

$$(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7) = (k_2, k_1, k_6, k_7, k_0, k_3, k_4, k_5).$$

  - $rk_i = k_{(i+2) \mod 8} \oplus con_i^{128}$.

3 DFA on Piccolo-80

Our proposed fault assumption includes the following assumptions.

– The attacker has the capability to choose one plaintext to encrypt and obtain the corresponding right/faulty ciphertexts.
– The attacker can induce random byte faults to the input register of round 23.
– The location and value of faults are both unknown.

3.1 Computation of the exact fault position

First, we present the method to compute the exact fault position by using the difference between right/faulty ciphertext pairs (C, C^*). According to our fault assumption, a random byte fault can be induced to the input byte register $I^j_{23,i}$ of round 23 ($i = 0, 1, 2, 3$ and $j = L, R$). Thus, the number of all possible fault positions is 8. For the simplicity of notations, we denote each case by $E^j_{23,i}$. For example, $E^L_{23,0}$ means an event that a random byte fault is injected to $I^L_{23,0}$.

***E*$_{23,0}$** We assume that a random byte fault was injected to $I_{23,0}$, that is an event $E^L_{23,0}$ or $E^R_{23,0}$ was occurred. Fig. 4 presents the differential propagation under this assumption. In this figure, blue lines mean an event $E^L_{23,0}$ and red lines mean an event $E^R_{23,0}$.

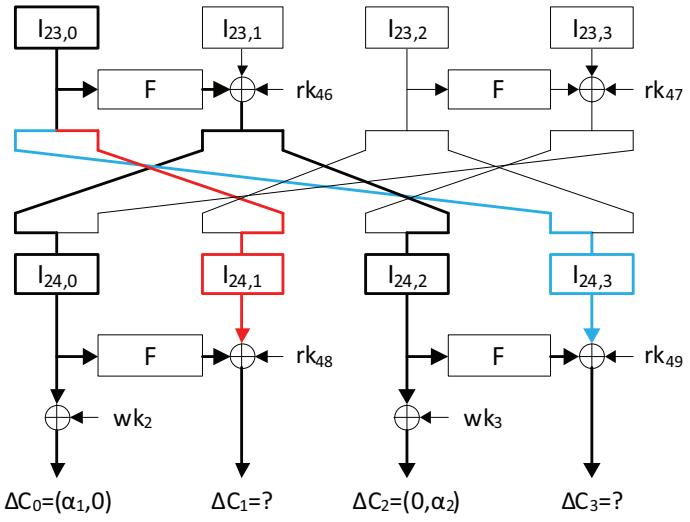

Fig. 4. Event $E_{23,0}$.

Under $E_{23,0}$, the difference $\Delta I_{23,0}$ for $I_{23,0}$ has the pattern of $(\alpha, 0)$ or $(0, \alpha)$. Here, α means a nonzero byte value. Then, according to the property of F function, the output difference of F function of round 23 has the pattern of (β, γ)

($\beta \neq 0, \gamma \neq 0$). As shown in Fig. 4, we cannot distinguish $E^L_{23,0}$ and $E^R_{23,0}$. Under two events, the difference ΔC between right/faulty ciphertext pairs (C, C^*) has the following pattern. Here, α_1 and α_2 are nonzero byte values and '?' means an unknown byte value.

– $E_{23,0}$: $\Delta C = (\alpha_1 \| 0,\ ? \| ?,\ 0 \| \alpha_2,\ ? \| ?)$.

$E^L_{23,1}$ and $E^R_{23,1}$ It is assumed that a random byte fault was injected to $I_{23,1}$, that is an event $E^L_{23,1}$ or $E^R_{23,1}$ was occurred. Fig. 5 presents the differential propagation under this assumption. In this figure, blue lines mean an event $E^L_{23,1}$ and red lines mean an event $E^R_{23,1}$.

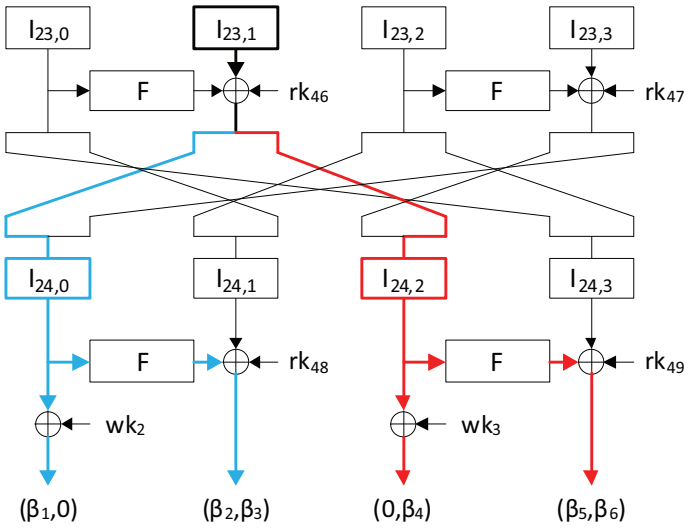

Fig. 5. Event $E_{23,1}$.

As depicted in Fig. 5, we can distinguish $E^L_{23,1}$ and $E^R_{23,1}$. In each case, the ciphertext difference has the following pattern. Here, β_j means a nonzero byte value ($j = 1, \cdots, 6$).

– $E^L_{23,1}$: $\Delta C = (\beta_1 \| 0,\ \beta_2 \| \beta_3,\ 0 \| 0,\ 0 \| 0)$.
– $E^R_{23,1}$: $\Delta C = (0 \| 0,\ 0 \| 0,\ 0 \| \beta_4,\ \beta_5 \| \beta_6)$.

$E_{23,2}$ It is assumed that an event $E^L_{23,2}$ or $E^R_{23,2}$ was occurred. That is, a random byte fault was injected to $I_{23,2}$. Fig. 6 presents the differential propagation under this assumption. In this figure, blue lines mean an event $E^L_{23,2}$ and red lines mean an event $E^R_{23,2}$.

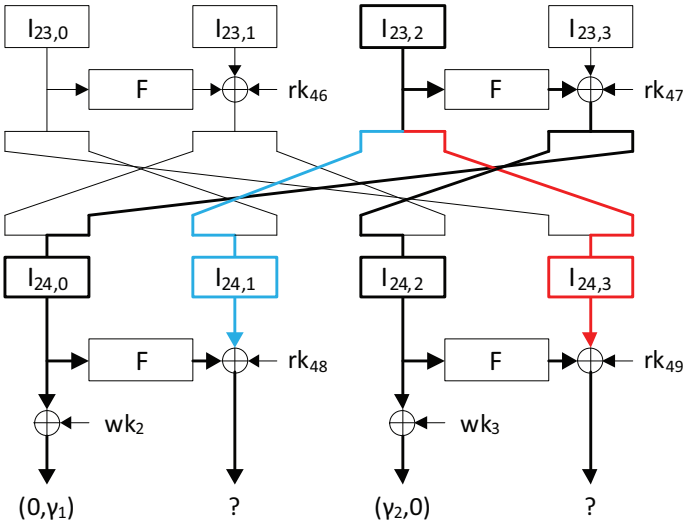

Fig. 6. Event $E_{23,2}$.

As depicted in Fig. 6, we cannot distinguish $E^L_{23,2}$ and $E^R_{23,2}$ similarly to $E_{23,0}$. In this case, the ciphertext difference has the following pattern. Here, γ_1 and γ_2 mean nonzero byte value and '?' means an unknown byte value..

– $E_{23,2}$: $\Delta C = (0\|\gamma_1,\ ?\|?,\ \gamma_2\|0,\ ?\|?)$.

$E^L_{23,3}$ and $E^R_{23,3}$ We assume that an event $E^L_{23,3}$ or $E^R_{23,3}$ was occurred. Fig. 7 presents the differential propagation under this assumption. In this figure, blue lines mean an event $E^L_{23,3}$ and red lines mean an event $E^R_{23,3}$.

As depicted in Fig. 7, we can distinguish $E^L_{23,3}$ and $E^R_{23,3}$ similarly to $E_{23,1}$. In each case, the ciphertext difference has the following pattern. Here, δ_j means a nonzero byte value ($j = 1, \cdots, 6$).

– $E^L_{23,3}$: $\Delta C = (0\|0,\ 0\|0,\ \delta_4\|0,\ \delta_5\|\delta_6)$.
– $E^R_{23,3}$: $\Delta C = (0\|\gamma_1,\ \gamma_2\|\gamma_3,\ 0\|0,\ 0\|0)$.

Under the above events, the ciphertext differences are summarized in Table 2. We can compute the exact fault position by using Table 2.

3.2 Computation of the secret key under each event

Second, we show how to compute candidates of the secret key of Piccolo-80 under each event. Recall that round keys are computed by using round constants and the secret key. Since round constants are known values, we can easily compute the partial secret key related to it when we get a round key. Thus, we do not consider round constants in this paper.

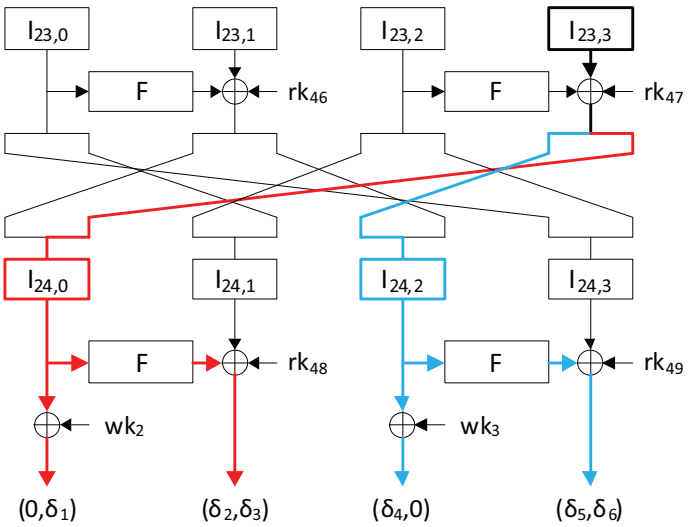

Fig. 7. Event $E_{23,3}$.

Table 2. Ciphertext differences for the positions of fault injections

Fault position	Ciphertext difference
$E_{23,0}$	$(\alpha_1\|\|0,\ ?\|\|?,\ 0\|\|\alpha_2,\ ?\|\|?)$
$E^L_{23,1}$	$(\beta_1\|\|0,\ \beta_2\|\|\beta_3,\ 0\|\|0,\ 0\|\|0)$
$E^R_{23,1}$	$(0\|\|0,\ 0\|\|0,\ 0\|\|\beta_4,\ \beta_5\|\|\beta_6)$
$E_{23,2}$	$(0\|\|\gamma_1,\ ?\|\|?,\ \gamma_2\|\|0,\ ?\|\|?)$
$E^L_{23,3}$	$(0\|\|0,\ 0\|\|0,\ \delta_4\|\|0,\ \delta_5\|\|\delta_6)$
$E^R_{23,3}$	$(0\|\|\gamma_1,\ \gamma_2\|\|\gamma_3,\ 0\|\|0,\ 0\|\|0)$

$\boldsymbol{E_{23,0}}$ Under $E_{23,0}$, we can obtain 2^{16} candidates of 48-bit (k_0^R, k_1^L, k_3, k_4). The attack procedure is as follows.

1. Guess 16-bit wk_2 $\left(= k_4^L \| k_3^R\right)$ and compute the output difference of the left F function in round 24 (see Fig. 4). Then check that the 8 most significant bits of this value are equal to the 8 most significant bits of ΔC_1. The probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k_3^R, k_4^L\right)$.
2. Guess 16-bit wk_3 $\left(= k_3^L \| k_4^R\right)$ and compute the output difference of the right F function in round 24 (see Fig. 4). Then check that the 8 least significant bits of this value are equal to the 8 least significant bits of ΔC_3. Since the filtering probability is 2^{-8}, we can compute 2^8 candidates of $\left(k_3^L, k_4^R\right)$.
3. Guess the 8 least significant bits of rk_{48} $\left(= con_{48}^{80} \oplus k_0\right)$ and the 8 most significant bits of rk_{49} $\left(= con_{49}^{80} \oplus k_1\right)$. By using 2^{16} candidates of (k_3, k_4) passing Step 1 and Step 2, check that the guessed value satisfy the input/output differences of the left F function in round 23 (see Fig. 4). The probability passing this test is 2^{-16}, we can get 2^{16} candidates of (k_0^R, k_1^L, k_3, k_4).

With the above attack procedure, we can obtain 2^{16} candidates of 48-bit (k_0^R, k_1^L, k_3, k_4) by using one random byte fault.

$\boldsymbol{E_{23,1}^L}$ **and** $\boldsymbol{E_{23,1}^R}$ Under $E_{23,1}^L$ or $E_{23,1}^R$, we can get the right (k_3^R, k_4^L) (under $E_{23,1}^L$) or (k_3^L, k_4^R) (under $E_{23,1}^R$), respectively. The attack procure is as follows.

- $E_{23,1}^L$
  - Guess 16-bit wk_2 $\left(= k_4^L \| k_3^R\right)$ and compute the output difference of the left F function in round 24 (see Fig. 5). Then check that this value is equal to ΔC_1. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k_3^R, k_4^L\right)$.
- $E_{23,1}^R$
  - Guess 16-bit wk_3 $\left(= k_3^L \| k_4^R\right)$ and compute the output difference of the right F function in round 24 (see Fig. 5). Then check that this value is equal to ΔC_3. Since the filtering probability is 2^{-16}, we can compute the right $\left(k_3^L, k_4^R\right)$.

$\boldsymbol{E_{23,2}}$ From the following attack procedure, we can obtain 2^{16} candidates of 48-bit (k_0^L, k_1^R, k_3, k_4) by using one random byte fault.

1. Guess 16-bit wk_2 $\left(= k_4^L \| k_3^R\right)$ and compute the output difference of the left F function in round 24 (see Fig. 6). Then check that the 8 least significant bits of this value are equal to the 8 least significant bits of ΔC_1. The probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k_3^R, k_4^L\right)$.
2. Guess 16-bit wk_3 $\left(= k_3^L \| k_4^R\right)$ and compute the output difference of the right F function in round 24 (see Fig. 6). Then check that the 8 most significant bits of this value are equal to the 8 most significant bits of ΔC_3. Since the filtering probability is 2^{-8}, we can compute 2^8 candidates of $\left(k_3^L, k_4^R\right)$.

3. Guess the 8 most significant bits of rk_{48} $\left(= con_{48}^{80} \oplus k_0\right)$ and the 8 least significant bits of rk_{49} $\left(= con_{49}^{80} \oplus k_1\right)$. By using 2^{16} candidates of (k_3, k_4) passing Step 1 and Step 2, check that the guessed value satisfy the input/output differences of the right F function in round 23 (see Fig. 6). The probability passing this test is 2^{-16}, we can get 2^{16} candidates of (k_0^L, k_1^R, k_3, k_4).

$E_{23,3}^L$ and $E_{23,3}^R$ Finally, under $E_{23,3}^L$ or $E_{23,3}^R$, we can get the right (k_3^L, k_4^R) (under $E_{23,3}^L$) or (k_3^R, k_4^L) (under $E_{23,3}^R$), respectively. The attack procure is as follows.

- $E_{23,3}^L$
  - Guess 16-bit wk_3 $\left(= k_3^L \| k_4^R\right)$ and compute the output difference of the right F function in round 24 (see Fig. 7). Then check that this value is equal to ΔC_3. Since the filtering probability is 2^{-16}, we can compute the right $\left(k_3^L \| k_4^R\right)$.
- $E_{23,3}^R$
  - Guess 16-bit wk_2 $\left(= k_4^L \| k_3^R\right)$ and compute the output difference of the left F function in round 24 (see Fig. 7). Then check that this value is equal to ΔC_1. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k_3^R \| k_4^L\right)$.

3.3 DFA on Piccolo-80

Now, we are ready to propose DFA on Piccolo-80. Our attack consists of the following two substpes: we first compute the exact fault position by using the ciphertext difference, and then the partial secret key of Piccolo-80 is obtained according to the computed fault position.

The attack procedure on Piccolo-80 is as follows.

1. [**Collection of right ciphertext**] Choose a plaintext P and obtain the corresponding right ciphertext $C = (C_0, C_1, C_2, C_3)$.
2. [**Collection of faulty ciphertext**] After inducing an i-th random byte fault to the input register $I_{23} = (I_{23,0}, I_{23,1}, I_{23,2}, I_{23,3})$ of round 23, obtain the corresponding faulty ciphertext $C^{i*} = (C_0^{i*}, C_1^{i*}, C_2^{i*}, C_3^{i*})$ $(i = 1, \cdots, n)$.
3. [**Computation of fault positions**] Compute ΔC^i by using (C, C^{i*}) and then compute the exact fault positions from Table 2.
4. [**Computation of candidates of** (k_0, k_1, k_3, k_4)] According to fault positions computed in Step 3, compute candidates of (k_0, k_1, k_3, k_4) by using the method in Section 3.2.
5. [**Recovery of the** 80**-bit secret key**] Guess 16-bit k_2 for each candidate of (k_0, k_1, k_3, k_4) and then recover the 80-bit secret key by using one trial encryption.

We simulated our attack on a general PC $10,000$ times. As simulation results, we can obtain about 2^8 candidates of (k_0, k_1, k_3, k_4) by using six fault injections on average. Thus, we do an exhaustive search for $2^{24} (= 2^8 \cdot 2^{16})$ candidates of

$(k_0, k_1, k_2, k_3, k_4)$. Since the filtering probability is 2^{-64}, the expected number of wrong secret keys passing our attack algorithm is $2^{-40}(= 2^{24} \cdot 2^{-64})$. It means that the possibility that a wrong key can pass our attack algorithm is very low. As simulation results, we can always recover the 80-bit secret key of Piccolo-80 within a few seconds by using six fault injections on average.

4 DFA on Piccolo-128

In this section, we propose DFA on Piccolo-128. Our attack on Piccolo-128 is similar to that on Piccolo-80. Our fault assumption is as follows.

- The attacker has the capability to choose one plaintext to encrypt and obtain the corresponding right/faulty ciphertexts.
- The attacker can induce random byte faults to the input register of round 28 and 29, respectively.
- The location and value of faults are both unknown.

4.1 The main idea

We first recover the right $wk_2(= k_4^L \| k_7^R)$ and $wk_3(= k_7^L \| k_4^R)$ by injecting the input register of round 29. The attack procedure of this step is similar to that on Piccolo-80. In detail, under events $E_{29,0}$, $E_{29,1}$, $E_{29,2}$ and $e_{29,3}$, we compute them. Note that we compute only (wk_2, wk_3). As simulation results, we can always obtain the right (wk_2, wk_3) by using only two random fault injections. Note that our attack on Piccolo-80 can compute about 2^8 candidates of (k_0, k_1, k_3, k_4) by using six fault injections on average.

Second, we assume that random byte faults are injected to the input register of round 28, that is $E_{28,0}$, $E_{28,1}$, $E_{28,2}$ and $E_{28,3}$. Thus, the number of all possible fault positions is 8. By using the recovered (wk_2, wk_3) in the previous step, we can distinguish them. According to each fault position, we can compute candidates of the partial secret key similarly to DFA on Piccolo-80.

4.2 $E_{28,0}$

It is assumed that we know the right (wk_2, wk_3). Under $E_{28,0}^L$ or $E_{28,0}^R$, the differential propagation is depicted in Fig. 8. In this figure, blue lines mean an event $E_{28,0}^L$ and red lines mean an event $E_{28,0}^R$.

Since we know (wk_2, wk_3), we can compute the input difference $(\Delta I_{30,0}, \Delta I_{30,1}, \Delta I_{30,2}, \Delta I_{30,3})$ of round 30. As shown in Fig. 8, we cannot distinguish $E_{28,0}^L$ and $E_{28,0}^R$. Under two events, $(\Delta I_{30,1}, \Delta I_{30,3})$ has the following pattern. Here, α_1 and α_2 are nonzero byte values.

- $E_{28,0}$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (0\|0,\ \alpha_1\|\alpha_2)$.

Under $E_{28,0}$, we can obtain 2^{16} candidates of 48-bit $\left(k_2, k_3^L, k_5, k_6^R\right)$. The attack procedure is as follows.

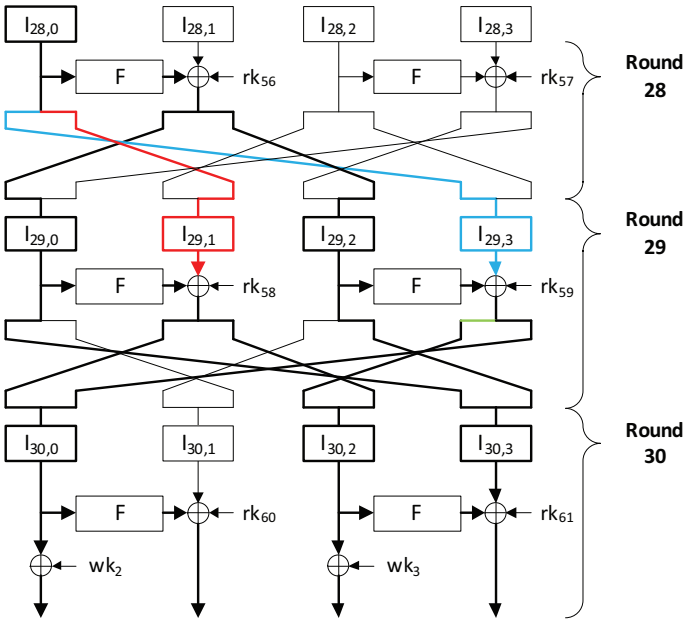

Fig. 8. Event $E_{28,0}$.

1. Guess 16-bit $(rk^R_{60}(= k^R_2 \oplus con^{128R}_{60}), rk^L_{61}(= k^L_5 \oplus con^{128L}_{61}))$ and compute the output difference of the left F function of round 29 (see Fig. 8). Then check that the 8 most significant bits of this value are equal to $\Delta I^L_{30,0}$. The probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k^R_2, k^L_5\right)$.
2. Guess 16-bit $(rk^L_{60}(= k^L_2 \oplus con^{128L}_{60}), rk^R_{61}(= k^R_5 \oplus con^{128R}_{61}))$ and compute the output difference of the right F function of round 29 (see Fig. 8). Then check that the 8 least significant bits of this value are equal to $\Delta I^R_{30,0}$. The probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k^L_2, k^R_5\right)$.
3. Guess 16-bit $(rk^R_{58}(= k^R_6 \oplus con^{128R}_{58}), rk^L_{59}(= k^L_3 \oplus con^{128L}_{59}))$ and compute the output difference of the left F function of round 28 for each candidate of (k_2, k_5) (see Fig. 8). Then check that this value is equal to $\Delta I_{30,3}$. The probability passing this test is 2^{-16}. Thus, we can obtain 2^{16} candidates of $\left(k_2, k^L_3, k_5, k^R_6\right)$.

4.3 $E^L_{28,1}$ and $E^R_{28,1}$

It is assumed that we know the right (wk_2, wk_3). Under $E^L_{28,1}$ or $E^R_{28,1}$, the differential propagation is shown in Fig. 9. In this figure, blue lines mean an event $E^L_{28,1}$ and red lines mean an event $E^R_{28,1}$. As depicted in Fig. 9, under each

event, $(\Delta I_{30,1}, \Delta I_{30,3})$ has the following pattern. Here, β_1 and β_2 are nonzero byte values.

- $E^L_{28,1}$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (0\|0,\ \beta_1\|0)$.
- $E^R_{28,1}$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (0\|0,\ 0\|\beta_2)$.

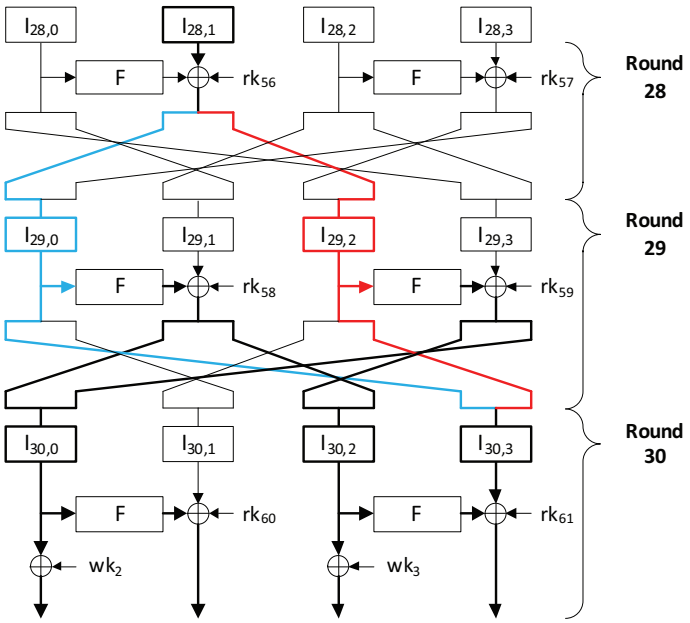

Fig. 9. Event $E_{28,1}$.

Under $E^L_{28,1}$ or $E^R_{28,1}$, we can get the right (k^R_2, k^L_5) (under $E^L_{28,1}$) or (k^L_2, k^R_5) (under $E^R_{28,1}$), respectively. The attack procure is as follows.

- $E^L_{28,1}$
  - Guess 16-bit $(rk^R_{60}(= k^R_2 \oplus con^{128R}_{60}), rk^L_{61}(= k^L_5 \oplus con^{128L}_{61}))$ and compute the output difference of the left F function of round 29 (see Fig. 9). Then check that this value is equal to $(\Delta I^L_{30,0} \| \Delta I^R_{30,2})$. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k^R_2, k^L_5\right)$.
- $E^R_{28,1}$
  - Guess 16-bit $(rk^L_{60}(= k^L_2 \oplus con^{128L}_{60}), rk^R_{61}(= k^R_5 \oplus con^{128R}_{61}))$ and compute the output difference of the right F function of round 29 (see Fig. 9). Then check that this value is equal to $(\Delta I^L_{30,2} \| \Delta I^R_{30,0})$. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k^L_2, k^R_5\right)$.

4.4 $E_{28,2}$

It is assumed that we know the right (wk_2, wk_3). Under $E^L_{28,2}$ or $E^R_{28,2}$, the differential propagation is shown in Fig. 10. In this figure, blue lines mean an event $E^L_{28,2}$ and red lines mean an event $E^R_{28,2}$. As depicted in Fig. 10, we cannot distinguish $E^L_{28,2}$ and $E^R_{28,2}$ similarly to $E_{28,0}$. Under each event, $(\Delta I_{30,1}, \Delta I_{30,3})$ has the following pattern. Here, γ_1 and γ_2 are nonzero byte values.

– $E_{28,2}$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (\gamma_1 \| \gamma_2,\ 0 \| 0)$.

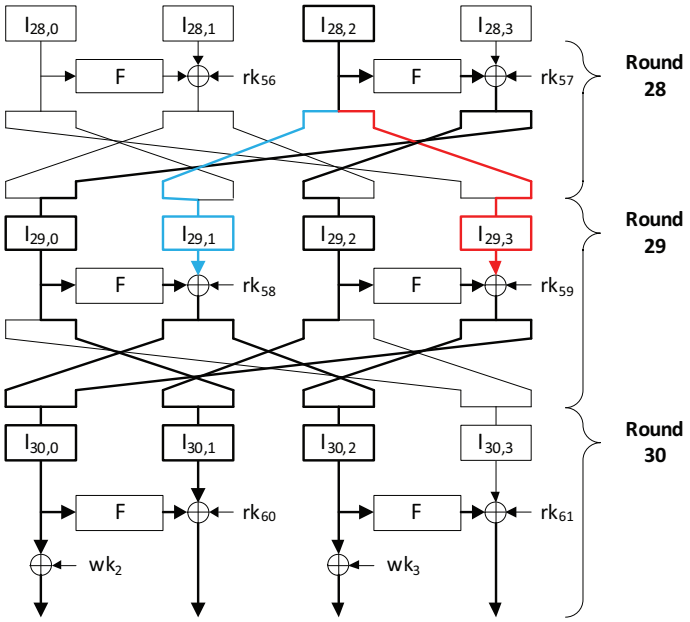

Fig. 10. Event $E_{28,2}$.

From the following attack procedure, we can obtain 2^{16} candidates of 48-bit (k_2, k_3^R, k_5, k_6^L) by using one random byte fault.

1. Guess 16-bit $(rk_{60}^R (= k_2^R \oplus con_{60}^{128R}), rk_{61}^L (= k_5^L \oplus con_{61}^{128L}))$ and compute the output difference of the left F function of round 29 (see Fig. 10). Then check that the 8 least significant bits of this value are equal to $\Delta I^R_{30,2}$. The probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k_2^R, k_5^L\right)$.
2. Guess 16-bit $(rk_{60}^L (= k_2^L \oplus con_{60}^{128L}), rk_{61}^R (= k_5^R \oplus con_{61}^{128R}))$ and compute the output difference of the right F function of round 29 (see Fig. 10). Then check that the 8 most significant bits of this value are equal to $\Delta I^L_{30,2}$. The

probability passing this test is 2^{-8}. Thus, we can obtain 2^8 candidates of $\left(k_2^L, k_5^R\right)$.

3. Guess 16-bit $(rk_{58}^L (= k_6^L \oplus con_{58}^{128L}), rk_{59}^R (= k_3^R \oplus con_{59}^{128R}))$ and compute the output difference of the right F function of round 28 for each candidate of (k_2, k_5) (see Fig. 10). Then check that this value is equal to $\Delta I_{30,1}$. The probability passing this test is 2^{-16}. Thus, we can obtain 2^{16} candidates of $\left(k_2, k_3^R, k_5, k_6^L\right)$.

4.5 $E_{28,3}^L$ and $E_{28,3}^R$

It is assumed that we know the right (wk_2, wk_3). Under $E_{28,3}^L$ or $E_{28,3}^R$, the differential propagation is shown in Fig. 11. In this figure, blue lines mean an event $E_{28,3}^L$ and red lines mean an event $E_{28,3}^R$. As depicted in Fig. 11, under each event, $(\Delta I_{30,1}, \Delta I_{30,3})$ has the following pattern. Here, δ_1 and δ_2 are nonzero byte values.

- $E_{28,3}^L$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (\delta_1 \| 0,\ 0 \| 0)$.
- $E_{28,3}^R$: $(\Delta I_{30,1}, \Delta I_{30,3}) = (0 \| \delta_2,\ 0 \| 0)$.

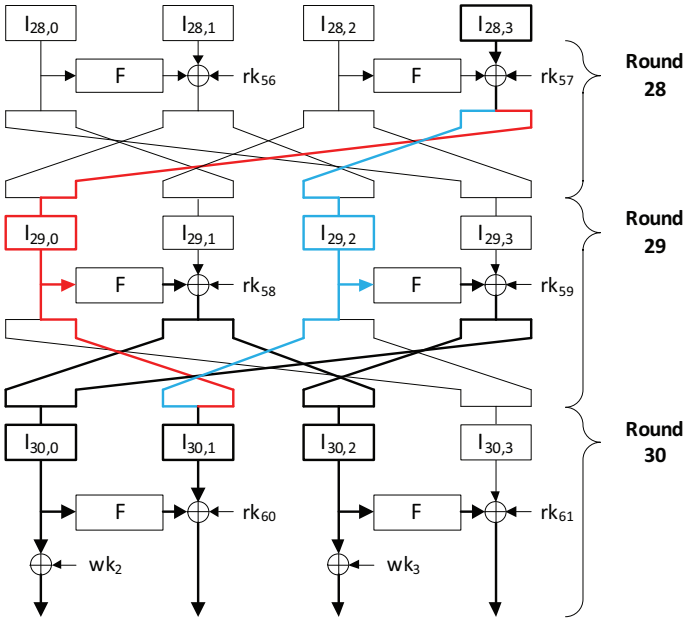

Fig. 11. Event $E_{28,3}$.

Under $E_{28,3}^L$ or $E_{28,3}^R$, we can get the right (k_2^L, k_5^R) (under $E_{28,3}^L$) or (k_2^R, k_5^L) (under $E_{28,3}^R$), respectively. The attack procure is as follows.

– $E^L_{28,3}$
  - Guess 16-bit $(rk^L_{60}(= k^L_2 \oplus con^{128L}_{60}), rk^R_{61}(= k^R_5 \oplus con^{128R}_{61}))$ and compute the output difference of the right F function of round 29 (see Fig. 11). Then check that this value is equal to $(\Delta I^L_{30,2} \| \Delta I^R_{30,0})$. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k^L_2, k^R_5\right)$.
– $E^R_{28,3}$
  - Guess 16-bit $(rk^R_{60}(= k^R_2 \oplus con^{128R}_{60}), rk^L_{61}(= k^L_5 \oplus con^{128L}_{61}))$ and compute the output difference of the left F function of round 29 (see Fig. 11). Then check that this value is equal to $(\Delta I^L_{30,0} \| \Delta I^R_{30,2})$. The probability passing this test is 2^{-16}. Thus, we can obtain the right $\left(k^R_2, k^L_5\right)$.

Under the above events, $(\Delta I_{30,1}, \Delta I_{30,3})$ are summarized in Table 3. Since we know (wk_2, wk_3), we can compute the exact fault position by using Table 3.

Table 3. Ciphertext differences for the positions of fault injections

Fault position	$(\Delta I_{30,1}, \Delta I_{30,3})$
$E_{28,0}$	$(0 \| 0,\ \alpha_1 \| \alpha_2)$
$E^L_{28,1}$	$(0 \| 0,\ \beta_1 \| 0)$
$E^R_{28,1}$	$(0 \| 0,\ 0 \| \beta_2)$
$E_{28,2}$	$(\gamma_1 \| \gamma_2,\ 0 \| 0)$
$E^L_{28,3}$	$(\delta_1 \| 0,\ 0 \| 0)$
$E^R_{28,3}$	$(0 \| \delta_2,\ 0 \| 0)$

4.6 DFA on Piccolo-128

The attack procedure on Piccolo-128 is as follows.

1. **[Collection of right ciphertext]** Choose a plaintext P and obtain the corresponding right ciphertext $C = (C_0, C_1, C_2, C_3)$.
2. **[Collection of faulty ciphertext]** After inducing an i-th random byte fault to the input register $I_{29} = (I_{29,0}, I_{29,1}, I_{29,2}, I_{29,3})$ of round 29, obtain the corresponding faulty ciphertext $C^{i*} = (C^{i*}_0, C^{i*}_1, C^{i*}_2, C^{i*}_3)$ $(i = 1, \cdots, n)$.
3. **[Recovery of (wk_2, wk_3)]** Recover the right (wk_2, wk_3) under events $E_{29,0}$, $E_{29,1}$, $E_{29,2}$ and $E_{29,3}$ similarly to $E_{23,0}$, $E_{23,1}$, $E_{23,2}$ and $E_{23,3}$ in DFA on Piccolo-80.
4. **[Collection of faulty ciphertext]** After inducing an i-th random byte fault to the input register $I_{28} = (I_{28,0}, I_{28,1}, I_{28,2}, I_{28,3})$ of round 28, obtain the corresponding faulty ciphertext $C^{i*} = (C^{i*}_0, C^{i*}_1, C^{i*}_2, C^{i*}_3)$ $(i = 1, \cdots, n)$.
5. **[Computation of candidates of $(k_2, k_3, k_4, k_5, k_6, k_7)$]** According to computed fault positions, compute candidates of $(k_2, k_3, k_4, k_5, k_6, k_7)$ by using the method in the previous subsections.

6. **[Recovery of the 128-bit secret key]** Guess 32-bit (k_0, k_1) for each candidate of $(k_2, k_3, k_4, k_5, k_6, k_7)$ and then recover the 128-bit secret key by using one trial encryption.

We simulated our attack on a general PC $10,000$ times. As simulation results, we can obtain about 2^8 candidates of (k_0, k_1, k_3, k_4) by using eight fault injections on average. Thus, we do an exhaustive search for $2^{40}(= 2^8 \cdot 2^{32})$ candidates of $(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7)$. Since the filtering probability is 2^{-64}, the expected number of wrong secret keys passing our attack algorithm is $2^{-24}(= 2^{40} \cdot 2^{-64})$. It means that the possibility that a wrong key can pass our attack algorithm is very low. As simulation results, we can always recover the 128-bit secret key of Piccolo-128 within one day by using eight fault injections on average.

5 Conclusion

In this paper, we have presented DFA on Piccolo. Our attack on Piccolo-80 is executed within a few seconds by using six random byte faults. And our attack on Piccolo-128 needs eight random byte faults and is executed within one day. They are first known side-channel attack results on Piccolo.

References

1. E. Biham and A. Shamir, *Differential fault analysis of secret key cryptosystems*, Crypto 1997, LNCS 1294, pp. 513-525, Springer, 1997.
2. C. Cannière, O. Dunkelman and M. Knežević, *KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers*, CHES 2009, LNCS 5747, pp. 272-288, Springer, 2009.
3. H. Chen, W. Wu and D. Feng, *Differential fault analysis on CLEFIA*, ICICS 2007, LNCS 4861, pp. 284-295, Springer, 2007.
4. J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, *The LED Block Cipher*, CHES 2011, LNCS 6917, pp. 326-341, Springer, 2011.
5. K. Jeong and C. Lee, *Differential Fault Analysis on Block Cipher LED-64*, Future Information Technology, Application, and Service, LNEE 164, pp. 747-755, Springer, 2012.
6. K. Jeong, Y. Lee, J. Sung and S. Hong, *Differential fault analysis on block cipher SEED*, Mathematical and Computer Modelling, Vol. 55, Issues 1-2, pp. 26-34, Elsevier, 2012.
7. L. Knudsen, G. Leander, A. Poschmann and M. Robshaw, *PRINTcipher: a block cipher for IC-printing*, CHES 2010, LNCS 6225, pp. 16-32, Springer, 2010.
8. W. Li, D. Gu and J. Li, *Differential fault analysis on the ARIA algorithm*, Information Sciences, Vol. 178, Issue 19, pp. 3727-3737, Elsevier, 2008.
9. K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, *Piccolo: an ultra-lightweight blockcipher*, CHES 2011, LNCS 6917, pp. 342-357, Springer, 2011.
10. M. Tunstall, D. Mukhopadhyay and S. Ali, *Differential fault analysis of the advanced encryption standard using a single fault*, WISTP 2011, LNCS 6633, pp. 224-233, Springer, 2011.
11. Y. Wang, W. Wu and X. Yu, *Biclique Cryptanalysis of Reduced-Round Piccolo Block Cipher*, ISPEC 2012, LNCS 7232, pp. 337-352, Springer, 2012.