Linear Fault Analysis of Block Ciphers

Zhiqiang Liu[1,⋆], Dawu Gu[1], Ya Liu[1], and Wei Li[2]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, P.R. China
{ilu_zq,dwgu,liuya0611}@sjtu.edu.cn
[2] School of Computer Science and Technology,
Donghua University, Shanghai 201620, P.R. China
liwei.cs.cn@gmail.com

Abstract. Differential fault analysis (DFA) has already been applied to attack many block ciphers with the help of inducing some faults at the last few rounds of block ciphers. Currently, a general countermeasure against DFA is to protect the last few rounds of block ciphers by means of redundancy. In this paper, we present a new fault attack on block ciphers called linear fault analysis (LFA), in which linear characteristics for some consecutive rounds of a block cipher will be utilized instead of exploiting differential distributions of S-Boxes within the block cipher in DFA. Basically, the new approach can handle the case that faults are induced several rounds earlier compared to DFA, thus leading to a threat to the protected implementations (against DFA) of block ciphers. For the purpose of illustration, we mount an effective attack on SERPENT by adopting LFA and achieve a good cryptanalytic result on SERPENT. We hope that our work enriches the picture on the applicability of fault attacks to block ciphers and could be beneficial to the security evaluation of block ciphers.

Keywords: Differential Fault Analysis, Linear Fault Analysis, Block Ciphers, SERPENT.

1 Introduction

In recent years, side channel attacks [1] have become important and efficient cryptanalytic tools in analyzing various cryptographic devices. These attacks exploit easily accessible information such as power consumption, running time, input-output behavior under malfunctions, and so on, and then evaluate such leaked information with the help of statistical methods. To some extent, side channel attacks are often more powerful than classical approaches such as differential cryptanalysis [2], linear cryptanalysis [3], related-key cryptanalysis [4], integral cryptanalysis [5], algebraic attack [6], and so on.

⋆ This work has been supported by National Natural Science Foundation of China (No. 61073150 and No. 61003278), the Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, and the Fundamental Research Funds for the Central Universities.

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 241–256, 2012.

As one of side channel attacks, fault analysis is a class of implementation attacks that disturb cryptographic computations so as to recover secret keys. To speak specifically, when an encryption is executed under faulty condition, an error occurs at some intermediate state, which results in a faulty output. The faulty output is then used as leaked information to help retrieve secret key. Since the fault analysis was first introduced in 1997 by Boneh *et al* [7], various methods of fault analysis have been proposed and studied. Among them, differential fault analysis(DFA) [8] can be regarded as the most effective cryptanalytic method against block ciphers. In fact, DFA derives information about the secret key of a block cipher by using differences between correct and faulty ciphertexts. Generally, an attacker gets faulty ciphertexts by giving external impact on a device with voltage variation, glitch, laser, etc [9]. To date, much research work has been devoted to DFA on a variety of block ciphers such as DES [8,10], AES [11,12,13,14,15,16,17,18], IDEA [19], CLEFIA [20], SMS4 and MacGuffin [21], ARIA [22] and Camellia [23]. Such work demonstrates the vulnerability of block ciphers towards DFA and the subsequent need of including countermeasures in embedded implementations of block ciphers. Moreover, some extensions to DFA have been presented in [24,25,26] in order to make fault attack more efficient.

As a matter of fact, most of DFA techniques target the last few rounds of a block cipher, i.e., faults will be triggered at the last few rounds of the cipher so as to induce information leakage. Consequently, the general countermeasure against DFA is to protect the last few rounds of the cipher by means of redundancy. However, the implementation of the countermeasure against DFA is more costly and less efficient along with the number of protected rounds increasing, thus for a block cipher, the practical implementations used to thwart DFA will cover as less protected rounds as possible.

In this paper, we propose a new fault attack on block ciphers called linear fault analysis (LFA), in which linear characteristics for some consecutive rounds of a block cipher will be utilized instead of exploiting differential distributions of S-Boxes within the block cipher in DFA. Generally, the new approach can deal with the case that faults are injected several rounds earlier compared to DFA as long as suitable linear approximations exist. Thus based on our new method, one may mount an effective attack on a block cipher even if the cipher has already been protected against DFA. Furthermore, in order to demonstrate the validity of LFA, we apply it to analyze the block cipher SERPENT which is a candidate of Advanced Encryption Standard and rated just behind the AES Rijndael. To the best of our knowledge, there isn't any known DFA attack on SERPENT which can be done by inducing faults at the round earlier than the penultimate round of the cipher, as a result, the countermeasure against DFA on SERPENT can be implemented by protecting the last two rounds of the cipher. However, we present an effective attack on the protected implementation of SERPENT by means of LFA. As a new extension of fault attack, LFA may be beneficial to the security evaluation of block ciphers.

The rest of the paper is organized as follows. Section 2 introduces the notations used throughout this paper and shows the possibility and rationality of

fault injection briefly. Section 3 presents our new fault attack on block ciphers, that is, linear fault analysis. Section 4 applies our novel approach to mount an effective attack on the DFA-proof implementation of SERPENT. Finally, Section 5 summarizes the paper.

2 Preliminaries

The following notations are used throughout the paper.

- $\oplus$ denotes bitwise exclusive OR (XOR).
- $\cdot$ denotes bitwise inner product.
- $|x|$ denotes the absolute value of a real number x.
- $\circ$ denotes the composition operation.
- $\#S$ denotes the cardinality of a set S.
- 0x denotes the hexadecimal notation.

2.1 Fault Injection

In practice, many block ciphers have been implemented in cryptographic devices such as smart cards and RFID tags. Generally, we can assume that a cryptographic device with fixed secret key is under the control of the attacker who could use the device to encrypt (or decrypt) arbitrary plaintexts (or ciphertexts). Accordingly, the attacker is able to deliberately interfere the normal operation of the device with electromagnetic perturbations, voltage variations, clock glitches and lasers so as to induce faults [9].

With the development of fault injection techniques, many fault models have been established, among which single bit error model and single byte error model are the most well-studied and applicable. Actually, in order to trigger single bit error or single byte error at certain intermediate state within the encryption/decryption process of a block cipher, laser technique could be adopted since a laser with certain energy and wavelength could interfere fixed parts of the memory/registers without damaging them, resulting in single bit error or single byte error at some internal state accurately [27].

3 Linear Fault Analysis

We now present a new fault attack on block ciphers called linear fault analysis (LFA), in which linear characteristics for some consecutive rounds of a block cipher will be utilized instead of exploiting differential distributions of S-Boxes within the block cipher in DFA.

3.1 Fault Model and Assumption

Our attack is applicable in the single bit error model as well as single byte error model. To speak specifically, for a given block cipher, the attacker has the capability to choose plaintexts to encrypt, and during the encryption process, he can

repeatedly induce single bit error or single byte error at the input of some certain round of the block cipher so as to obtain the corresponding right and faulty ciphertexts. Note that the values and positions (within the impacted round) of the faults injected by the attacker are unknown and randomly distributed.

3.2 Principle of Linear Fault Analysis

In the following we will demonstrate the principle of linear fault analysis (LFA) under the condition of single bit error model. Firstly, we will introduce the definition of linearly active input set with respect to a linear approximation and give a claim related to linear fault analysis.

Definition 1. Let E_1 be a block cipher and $\Gamma_P \cdot P \oplus \Gamma_C \cdot C = \Gamma_K \cdot K$ (also denoted as $\Gamma_P \to \Gamma_C$) be a linear approximation for E_1. Let $S_{\Gamma_P \to \Gamma_C}$ be a set consisting of all bits of P involving in the item $\Gamma_P \cdot P$. Then $S_{\Gamma_P \to \Gamma_C}$ is denoted as the linearly active input set with respect to the linear characteristic $\Gamma_P \to \Gamma_C$.

Claim 1. Let E be a block cipher and decompose the cipher into $E = E_1 \circ E_0$, where E_0 represents the first part of the cipher and E_1 represents the last part. Let $\Gamma_P \cdot P \oplus \Gamma_C \cdot C = \Gamma_K \cdot K$ be a linear approximation for E_1 with probability $1/2 + \varepsilon$ and $S_{\Gamma_P \to \Gamma_C}$ be the linearly active input set with respect to the linear characteristic $\Gamma_P \to \Gamma_C$. Suppose that an attacker has the ability to induce single bit error at the input of E_1 repeatedly and the error bits don't belong to the set $S_{\Gamma_P \to \Gamma_C}$, then an effective distinguisher $\Gamma_C \cdot C_1 \oplus \Gamma_C \cdot C_2 = 0$ for the cipher E with probability $1/2 + 2\varepsilon^2$ can be derived by the attacker.

Next we will show the reasonability of Claim 1 in detail. First of all, we find that the effect of injecting single bit error at the input of E_1 repeatedly with error bits not in the set $S_{\Gamma_P \to \Gamma_C}$, is somewhat like constructing a particular differential-linear distinguisher that is composed of a special truncated differential characteristic *unknown input difference* $\to \nabla$ for E_0 with probability 1 and a linear characteristic $\Gamma_P \to \Gamma_C$ for E_1 with probability $1/2 + \varepsilon$ such that $\nabla \cdot \Gamma_P = 0$. For any pair consisting of a right ciphertext C_1 under E and the corresponding faulty ciphertext C_2 derived under the above condition, we have that both

$$\Gamma_P \cdot E_1^{-1}(C_1) \oplus \Gamma_C \cdot C_1 = \Gamma_K \cdot K$$

and

$$\Gamma_P \cdot E_1^{-1}(C_2) \oplus \Gamma_C \cdot C_2 = \Gamma_K \cdot K$$

hold with probability $1/2+\varepsilon$. Assume that these two equations are uncorrelated, and take into account the condition that the error bit induced at the input of E_1 is not in the set $S_{\Gamma_P \to \Gamma_C}$, we immediately obtain that

$$\Gamma_C \cdot C_1 \oplus \Gamma_C \cdot C_2 = 0$$

holds with probability

$$(1/2 + \varepsilon)^2 + (1/2 - \varepsilon)^2 = 1/2 + 2\varepsilon^2.$$

Thus the distinguisher by checking the parity of $\Gamma_C \cdot C_1 \oplus \Gamma_C \cdot C_2$ can be utilized to distinguish the cipher E from a random permutation since for such a ciphertext pair (C_1, C_2), the equation $\Gamma_C \cdot C_1 \oplus \Gamma_C \cdot C_2 = 0$ holds with probability $1/2$ for a random permutation. □

Based on the result given in the above Claim as well as the Algorithm 2 proposed in [3], we can mount a key recovery attack on $E' = E_2 \circ E = E_2 \circ E_1 \circ E_0$ (where E_2 represents the last round of the cipher E') by guessing part of the last round subkey of E'. The general attack procedure can be described as follows:

Step 1. Given the linear characteristic $\Gamma_P \rightarrow \Gamma_C$ for E_1, collect N pairs of ciphertexts, each pair consisting of a right ciphertext C_1^i under E' and the corresponding faulty ciphertext C_2^i derived by injecting single bit error at any position of the input of E_1, where $1 \leq i \leq N$.

Step 2. Let K_g denote the bits of the last round subkey which are related to the item $\Gamma_C \cdot E_2^{-1}(C_j^i)$, i.e., $\Gamma_C \cdot E_2^{-1}(C_j^i)$ can be obtained by performing a partial decryption of the ciphertext C_j^i with the guessed value of K_g, where $1 \leq i \leq N, 1 \leq j \leq 2$. Then for each possible value of K_g, do the following:
(1). Initialize a counter T_{K_g} firstly.
(2). For each ciphertext pair (C_1^i, C_2^i), implement the partial decryptions of C_1^i and C_2^i respectively and compute the parity of $\Gamma_C \cdot E_2^{-1}(C_1^i) \oplus \Gamma_C \cdot E_2^{-1}(C_2^i)$. If the parity is 0, increase the relevant counter T_{K_g} by 1, and decrease by 1 otherwise.
(3). Store the value of K_g as well as the value of the corresponding $|T_{K_g}|$.

Step 3. For all possible values of K_g, compare the stored values and take the value of K_g as the correct key information if the value of the corresponding $|T_{K_g}|$ is maximal.
Note that in the above key recovery attack, the single bit error can be triggered **at any position of the input of** E_1. Here we want to discuss the rationality of the key recovery attack. On the one hand, if the guessed value of K_g is correct, then for any ciphertext pair (C_1^i, C_2^i) in which C_2^i is derived by inducing single bit error at the input of E_1 such that the error bit is not in the set $S_{\Gamma_P \rightarrow \Gamma_C}$, the equation $\Gamma_C \cdot E_2^{-1}(C_1^i) \oplus \Gamma_C \cdot E_2^{-1}(C_2^i) = 0$ holds with probability $1/2 + 2\varepsilon^2$, and for any ciphertext pair (C_1^i, C_2^i) where C_2^i is obtained by injecting single bit error at the input of E_1 such that the error bit belongs to the set $S_{\Gamma_P \rightarrow \Gamma_C}$, the equation $\Gamma_C \cdot E_2^{-1}(C_1^i) \oplus \Gamma_C \cdot E_2^{-1}(C_2^i) = 1$ holds with probability $1/2 + 2\varepsilon^2$. Thus in the case that the guessed value of K_g is correct, we can estimate $|T_{K_g}|$ by

$$|T_{K_g}| \approx N \times |1 - 2 \times \#S_{\Gamma_P \rightarrow \Gamma_C}/n| \times 4\varepsilon^2,$$

where n is the block size of the cipher E', and $\#S_{\Gamma_P \rightarrow \Gamma_C}$ is not equal to $n/2$. On the other hand, if the guessed value of K_g is wrong, according to the Wrong-Key Randomization Hypothesis given in [28], it's assumed that the wrong guess of K_g results in a random-looking parity of $\Gamma_C \cdot E_2^{-1}(C_1^i) \oplus \Gamma_C \cdot E_2^{-1}(C_2^i)$. Consequently,

the value of $|T_{K_g}|$ approximates to 0 in this case. So it is feasible to distinguish the correct value of K_g from all wrong guesses of K_g by applying the above key recovery attack if given sufficient ciphertext pairs (C_1^i, C_2^i), where C_2^i is gained by triggering single bit error at any position of the input of E_1. Following the technique introduced in [3], the number of ciphertext pairs required in our key recovery attack can be estimated as

$$c_N \times \frac{1}{|1 - 2 \times \#S_{\Gamma_P \to \Gamma_C}/n|} \times \frac{1}{4\varepsilon^4} ,$$

where the coefficient c_N, which is closely related to the number of guessed subkey bits and the desired success rate of our attack, can be measured by using the approach given in [29].

Furthermore, if an adversary has the capability to induce single bit error at the input of E_1 repeatedly as well as the ability to get several linear characteristics $\Gamma_P^i \to \Gamma_C^i$ $(1 \le i \le m)$ for E_1, where the calculations of $\Gamma_C^i \cdot E_2^{-1}$ and $\Gamma_C^j \cdot E_2^{-1}$ $(i \ne j)$ influence different bits of the last round subkey of E', then the adversary can mount the above key recovery attack m times so as to recover more bits of the last round subkey (note that ciphertext pairs could be multiplexed partially or entirely among these attacks). After the adversary derives enough bits of the last round subkey, he can guess the left unknown bits of the subkey by means of exhaustive search if needed, and then the last round of E' can be stripped. Repeat the above procedure until the adversary can recover the secret key of the cipher E'.

Regarding the linear fault analysis under the condition of single byte error model, similar result can be derived by the same means as above.

4 A Key Recovery Attack on SERPENT by Using LFA

In order to illustrate the effectiveness of LFA, we mount a key recovery attack on the block cipher SERPENT by using LFA in this section. Since there isn't any known DFA attack on SERPENT which can be done by inducing faults at the round earlier than the penultimate round of the cipher so far, the general countermeasure against DFA on SERPENT could be implemented by protecting the last two rounds of the cipher if taking into account the cost and efficiency of the implementation. However, our effective attack shows that LFA could be a threat to the protected implementation of SERPENT.

4.1 A Brief Description of SERPENT

The SERPENT block cipher was proposed by Anderson *et al* in 1998 [30]. As a candidate of Advanced Encryption Standard, it was rated just behind the AES Rijndael. SERPENT has a classical SPN structure with 32 rounds and 128-bit block size. It accepts keys of 128, 192 and 256 bits and consists of the following operations:

• an initial permutation IP;
• 32 rounds, each consisting of a key mixing operation, a passage through 32 S-boxes and a linear transformation (except the last round, where the linear transformation is replaced by an additional key mixing operation);
• a final permutation FP.

In our description we adopt the notations of [30] in the bitsliced version. The intermediate value just before the round i (i.e., the $(i+1)$-th round) is denoted by B_i (a 128-bit value), where $0 \leq i \leq 31$. Each B_i is composed of four 32-bit words X_0, X_1, X_2 and X_3, where bit j of X_k is the bit $4j+k$ of the 128-bit value B_i $(0 \leq j \leq 31,\ 0 \leq k \leq 3)$. The four bits, bit j of X_3, X_2, X_1 and X_0, consist of the nibble j (i.e., the $(j+1)$-th nibble), with the bit from X_3 as the most significant bit.

SERPENT uses 8 distinct 4-bit to 4-bit S-boxes S_i $(0 \leq i \leq 7)$ successively along the rounds and consequently, each S-box is used in exactly four different rounds (i.e., S_0 is used in round 0, S_1 is used in round 1, ..., after S_7 is used in round 7, S_0 will be adopted again in round 8, then S_1 in round 9, and so on).

As for each round function R_i $(0 \leq i \leq 31)$, a single S-box will be used 32 times in parallel. For instance, R_0 uses 32 copies of S_0, and the $(j+1)$-th copy of S_0 takes the nibble j as the input and then outputs the value according to the S-box, where $0 \leq j \leq 31$.

The cipher can be formally described as follows:
• $B_0 \leftarrow P$,
• $B_{i+1} \leftarrow R_i(B_i)\ \ 0 \leq i \leq 31$,
• $C \leftarrow B_{32}$,

where P, C denote plaintext and ciphertext respectively, and round function R_i can be expressed as below:

$$R_i(X) = LT(\hat{S}_i(X \oplus K_i)) \quad i = 0, \ldots, 30,$$
$$R_i(X) = \hat{S}_i(X \oplus K_i) \oplus K_{32} \quad i = 31,$$

where $\hat{S}_i$ denotes the application of the S-box $S_{(i \bmod 8)}$ 32 times in parallel, LT denotes the linear transformation, and K_i denotes the subkey of round i (note that both K_{31} and K_{32} are the subkeys of round 31). Please refer to [30] for detailed information about the 8 S-boxes, the linear transformation and the key schedule algorithm.

4.2 Attacking SERPENT

We now present a key recovery attack on SERPENT under the condition of single bit error model, and a similar attack can be mounted on SERPENT for the case of single byte error model. Firstly, we construct twelve 2-round linear characteristics $\Gamma_P^i \rightarrow \Gamma_C^i$ $(1 \leq i \leq 12)$ for the rounds from round 29 to round 30 of SERPENT, and assume that single bit error can be injected at the input of the round 29 repeatedly and randomly, then by applying the method given in Section 3 twelve times, 128 bits of K_{32} can be retrieved from the attack.

In order to describe the linear characteristics adopted in our attack, 32 hexadecimal digits will be used to denote the masks corresponding to 32 nibbles respectively, where the $(j+1)$-th hexadecimal digit (numbered from right to

left) corresponds to the nibble j, $0 \leq j \leq 31$. Please refer to Appendix for the depiction of the twelve 2-round linear characteristics used in our attack.

According to the approach given in Section 3, we could derive twelve distinguishers for the 31 rounds from round 0 to round 30 of SERPENT as below:

$$\Gamma_C^i \cdot SERPENT_{lr}^{-1}(C_1) \oplus \Gamma_C^i \cdot SERPENT_{lr}^{-1}(C_2) = 0, \quad 1 \leq i \leq 12, \qquad (1)$$

where $SERPENT_{lr}^{-1}$ means the inverse of the last round of SERPENT, C_1 is a right ciphertext under SERPENT and C_2 is the corresponding faulty ciphertext obtained by inducing single bit error at the input of the round 29 of SERPENT. Moreover, for the case $1 \leq i \leq 9$, the i-th equation in (1) holds with probability $1/2+2^{-9}$, and for the case $10 \leq i \leq 12$, the i-th equation in (1) holds with probability $1/2 + 2^{-7}$. Thus a key recovery attack can be mounted on SERPENT based on the above twelve distinguishers. Following gives the detailed description of the attack in three phases.

Phase 1. For the i-th ($1 \leq i \leq 3$) distinguisher given in equations (1), do the following:

Step 1. Collect N_i pairs of ciphertexts, each pair consisting of the right ciphertext C_1 under SERPENT and the corresponding faulty ciphertext C_2^j ($1 \leq j \leq N_i$) derived by randomly injecting single bit error at the input of the round 29 of SERPENT.

Step 2. Let K_g denote the 8 bits of K_{32} which are relevant to the two active nibbles (i.e., S-boxes) influenced by the distinguisher. Initialize 2^8 counters $\{T_l\}_{0 \leq l \leq 2^8-1}$ (the size of each counter could be set to $\lceil \log_2^{N_i} \rceil$ bits), where T_l corresponds to l which represents the possible value of the 8 bits of C_2^j entering the above two active nibbles. For each faulty ciphertext C_2^j, increase the counter T_l by 1 if the corresponding 8-bit value of the C_2^j is equal to l. Then for each possible value of K_g, do the following:

(a). Initialize a counter T_{K_g} with the counter size being $\lceil \log_2^{N_i} \rceil$ bits and implement the partial decryption of C_1.

(b). For each value of l, decrypt the above two active nibbles and then calculate the parity in terms of the distinguisher. If the parity is 0, increase the counter T_{K_g} by the value of T_l, and decrease by the value of T_l otherwise.

(c). Store the value of K_g as well as the value of the corresponding $|T_{K_g}|$.

Step 3. For all possible values of K_g, compare the stored values and take the value of K_g as the correct key information if the value of the corresponding $|T_{K_g}|$ is maximal.

Thus the 24 bits of K_{32} related to the nibbles 20, 21, 22, 25, 26 and 27 could be recovered in this phase.

Phase 2. For the i-th ($4 \leq i \leq 6$) distinguisher given in equations (1), the

attack steps are the same as those in Phase 1 except the step 2 which is described as below.

Step 2. Let K_g denote the 12 bits of K_{32} which are relevant to the three active nibbles (i.e., S-boxes) impacted by the distinguisher. Initialize 2^{12} counters $\{T_l\}_{0\le l\le 2^{12}-1}$ (the size of each counter could be set to $\lceil \log_2^{N_i} \rceil$ bits), where T_l corresponds to l which represents the possible value of the 12 bits of C_2^j entering the above three active nibbles. For each faulty ciphertext C_2^j, increase the counter T_l by 1 if the corresponding 12-bit value of the C_2^j is equal to l. Then for each possible value of K_g, do the following:

(a). Initialize a counter T_{K_g} with the counter size being $\lceil \log_2^{N_i} \rceil$ bits and implement the partial decryption of C_1.

(b). For each value of l, decrypt the above three active nibbles and then calculate the parity in terms of the distinguisher. If the parity is 0, increase the counter T_{K_g} by the value of T_l, and decrease by the value of T_l otherwise.

(c). Store the value of K_g as well as the value of the corresponding $|T_{K_g}|$.

Accordingly, the 36 bits of K_{32} corresponding to the nibbles 0, 1, 2, 3, 4, 5, 10, 11 and 12 could be obtained in this phase.

Phase 3. After the above 60 bits of K_{32} have been retrieved, we can mount attacks on SERPENT sequentially in terms of the 7th, 8th, 9th, 10th, 11th and 12th distinguishers given in equations (1), and these attacks are similar to that in Phase 1. Finally we can get all the 128 bits of K_{32}.

After that, we rewrite the encryption algorithm of SERPENT in an equivalent way by swapping the order of the linear transformation in round 30 and the key mixing operation (with K_{31}) in round 31, then for the modified cipher, the part after the XOR operation with $LT^{-1}(K_{31})$ can be stripped. Furthermore, we mount an attack on the reduced-round cipher similarly to the above and recover the 128 bits of K_{31}. Thus following the key schedule algorithm of SERPENT, we can derive the secret key from K_{31} and K_{32}.

For the attack in terms of the i-th ($1 \le i \le 12$) distinguisher given in equations (1), the necessary number of ciphertext pairs, each pair consisting of a right ciphertext under SERPENT and the corresponding faulty ciphertext derived by triggering single bit error at the input of the round 29 of SERPENT randomly, could be estimated as

$$c_{N_i} \times \frac{1}{1-2\times 6/128} \times \frac{1}{(2^{-9})^2}, \quad 1 \le i \le 9,$$

or

$$c_{N_i} \times \frac{1}{1-2\times 3/128} \times \frac{1}{(2^{-7})^2}, \quad 10 \le i \le 12,$$

where the coefficient c_{N_i} is closely related to the number of guessed subkey bits and the desired success rate of the attack. Thus for the attack corresponding to the i-th distinguisher, according to the Theorem 2 proposed in [29], $2^4 \times \frac{32}{29} \times 2^{18} \approx 2^{22.14}$ and $2^{3.8} \times \frac{64}{61} \times 2^{14} \approx 2^{17.87}$ ciphertext pairs are needed in the cases $1 \le i \le 9$ and $10 \le i \le 12$ respectively so as to achieve a high success probability

of 1 approximately. Note that the ciphertext pairs could be multiplexed in our key recovery attack on SERPENT, consequently, the data complexity of our attack with success probability of about 1 can be estimated as $2 \times 2^{22.14} = 2^{23.14}$ ciphertext pairs or $2^{23.14}$ faulty ciphertexts (taking the attack for recovering K_{31} into account as well).

The time complexity of our attack is dominated mainly by the decryptions of the active nibbles in the attack based on the 7th distinguisher of equations (1). As a result, the time complexity of our attack is around $2 \times 2^{22.14} \times 2^{16} \times \frac{6}{32 \times 32} \approx 2^{31.73}$ SERPENT encryptions (taking the attack for retrieving K_{31} into account as well).

The memory complexity of our attack is primarily owing to storing the value of $|T_{K_g}|$ in the attack based on the 7th distinguisher of equations (1) as well as keeping the required faulty ciphertexts. Considering the fact that the attacks for obtaining K_{32} and K_{31} are implemented sequentially, the memory complexity of our attack can be approximated as $(23 \times 2^{16} + 128 \times 2^{22.14})/8 \approx 2^{26.14}$ bytes.

4.3 Experiments and Results

We use a PC with i3 M380 processor(2.53 GHz) and 4G DDR memory to do the experiments of our key recovery attack on SERPENT. The software platform of the experiments is Visual C++, and fault inductions are simulated in this platform. Under this condition we implement 100 experiments of our attack on SERPENT with randomly generated secret keys.

The main procedure of each experiment is as follows. At first, a correct ciphertext is obtained by encrypting a given plaintext under SERPENT with a secret key. Secondly, we trigger single bit error at the input of round 29 of SERPENT randomly and repeatedly to derive $2^{22.14}$ faulty ciphertexts and then retrieve the 128 bits of K_{32} by the means presented in Section 4.2. After that, we inject single bit error at the input of round 28 of SERPENT randomly and repeatedly to generate $2^{22.14}$ faulty ciphertexts and then obtain the 128 bits of K_{31} by the means similar to the above. Finally, the secret key is recovered from K_{32} and K_{31} with the help of the key schedule algorithm of SERPENT.

Among all the 100 experiments, there are 92 experiments such that the recovered secret keys are equal to the corresponding correct ones. Consequently, our experimental results match the theoretical analysis given in Section 4.2 well.

5 Conclusion

In this paper, we have proposed a new fault attack on block ciphers called linear fault analysis (LFA), in which linear characteristics for some consecutive rounds of a block cipher will be utilized instead of exploiting differential distributions of S-Boxes within the block cipher in DFA. Generally, our new approach can deal with the case that faults are triggered several rounds earlier compared to DFA as long as suitable linear approximations exist, as a result, one may mount an effective attack on a block cipher by applying LFA even if the cipher has already been protected against DFA.

In order to demonstrate the validity of LFA, we have applied it to analyze the block cipher SERPENT. Basically, the countermeasure against DFA on SERPENT can be implemented by protecting the last two rounds of the cipher since there isn't any known DFA attack on SERPENT so far which can be done by inducing faults at the round earlier than the penultimate round of the cipher. However, with the help of LFA, we have presented an effective attack on the protected implementation of SERPENT. Although the attack has a data complexity which seems impractical for real cryptographic devices, it does show that LFA could be a potential threat to the protected implementations (against DFA) of block ciphers. Moreover, it is expected that further results could be derived by applying linear hulls and non-linear approximations in LFA.

Finally, the implementation of redundancy (a simple and widely used countermeasure against fault attack) is more costly and less efficient along with the number of protected rounds increasing, thus for a block cipher, the number of protected rounds must be chosen very carefully in order to prevent security flaws as well as keep the corresponding implementation economical and efficient. We hope that our work could be helpful in determining this number.

References

1. Kelsey, J., Schneier, B., Wagner, D., Hall, C.: Side Channel Cryptanalysis of Product Ciphers. In: Quisquater, J.-J., Deswarte, Y., Meadows, C., Gollmann, D. (eds.) ESORICS 1998. LNCS, vol. 1485, pp. 97–110. Springer, Heidelberg (1998)
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. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 398–409. Springer, Heidelberg (1994)
5. Knudsen, L., Wagner, D.: Integral Cryptanalysis (Extended Abstract). In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 112–127. Springer, Heidelberg (2002)
6. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002)
7. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the Importance of Checking Cryptographic Protocols for Faults. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997)
8. Biham, E., Shamir, A.: Differential Fault Analysis of Secret Key Cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997)
9. Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The Sorcerer's Apprentice Guide to Fault Attacks. In: FDTC 2004 in Association with DSN 2004, pp. 330–342 (2004)
10. Rivain, M.: Differential Fault Analysis on DES Middle Rounds. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 457–469. Springer, Heidelberg (2009)

11. Blömer, J., Seifert, J.-P.: Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 162–181. Springer, Heidelberg (2003)
12. Chen, C.-N., Yen, S.-M.: Differential Fault Analysis on AES Key Schedule and Some Countermeasures. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 118–129. Springer, Heidelberg (2003)
13. Dusart, P., Letourneux, G., Vivolo, O.: Differential Fault Analysis on A.E.S. In: Zhou, J., Yung, M., Han, Y. (eds.) ACNS 2003. LNCS, vol. 2846, pp. 293–306. Springer, Heidelberg (2003)
14. Giraud, C.: DFA on AES. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) AES 2005. LNCS, vol. 3373, pp. 27–41. Springer, Heidelberg (2005)
15. Kim, C.H., Quisquater, J.-J.: New Differential Fault Analysis on AES Key Schedule: Two Faults Are Enough. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 48–60. Springer, Heidelberg (2008)
16. Piret, G., Quisquater, J.-J.: A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 77–88. Springer, Heidelberg (2003)
17. Takahashi, J., Fukunaga, T., Yamakoshi, K.: DFA Mechanism on the AES Key Schedule. In: FDTC 2007, pp. 62–74 (2007)
18. Kim, C.H.: Improved Differential Fault Analysis on AES Key Schedule. IEEE Transactions on Information Forensics and Security 7(1), 41–50 (2012)
19. Clavier, C., Gierlichs, B., Verbauwhede, I.: Fault Analysis Study of IDEA. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 274–287. Springer, Heidelberg (2008)
20. Chen, H., Wu, W., Feng, D.: Differential Fault Analysis on CLEFIA. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 284–295. Springer, Heidelberg (2007)
21. Li, W., Gu, D., Wang, Y.: Differential Fault Analysis on the Contracting UFN Structure, with Application to SMS4 and MacGuffin. Journal of Systems and Software 82(2), 346–354 (2009)
22. Li, W., Gu, D., Li, J.: Differential Fault Analysis on the ARIA Algorithm. Information Sciences 10(178), 3727–3737 (2008)
23. Zhou, Y., Wu, W., Xu, N., Feng, D.: Differential Fault Attack on Camellia. Chinese Journal of Electronics 18(1), 13–19 (2009)
24. Phan, R.C.-W., Yen, S.-M.: Amplifying Side-Channel Attacks with Techniques from Block Cipher Cryptanalysis. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 135–150. Springer, Heidelberg (2006)
25. Kim, C.H.: Efficient Methods for Exploiting Faults Induced at AES Middle Rounds, `http://eprint.iacr.org/2011/349`
26. Derbez, P., Fouque, P.-A., Leresteux, D.: Meet-in-the-Middle and Impossible Differential Fault Analysis on AES. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 274–291. Springer, Heidelberg (2011)
27. Dutertre, J.M., Mirbaha, A.P., Naccache, D., Ribotta, A.L., Tria, A.: Reproducible Single-Byte Laser Fault Injection. In: PASTIS 2010 (2010)
28. Harpes, C., Kramer, G.G., Massey, J.L.: A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-Up Lemma. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 24–38. Springer, Heidelberg (1995)

29. Selçuk, A.A.: On Probability of Success in Linear and Differential Cryptanalysis. Journal of Cryptology 21(1), 131–147 (2008)
30. Anderson, R., Biham, E., Knudsen, L.R.: SERPENT: A Proposal for the Advanced Encryption Standard, NIST AES Proposal (1998)

Appendix: The Linear Characteristics Used in Section 4.2

The 2-round linear characteristic $(\Gamma_P^1 \to \Gamma_C^1)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^1 = \quad & \text{0x00E0000000000000000000000000000E} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x00400000000000000000000000000008} \\
& \qquad \downarrow LT \\
& \text{0x00000000000000000000000080000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x00000000000000000000000010000000} \\
& \qquad \downarrow LT \\
& \text{0x00000010000A0000000000000000000000} \quad = \Gamma_C^1,
\end{aligned}
$$

The 2-round linear characteristic $(\Gamma_P^2 \to \Gamma_C^2)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^2 = \quad & \text{0x0E0000000000000000000000000000E0} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x04000000000000000000000000000080} \\
& \qquad \downarrow LT \\
& \text{0x00000000000000000000008000000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x00000000000000000000001000000000} \\
& \qquad \downarrow LT \\
& \text{0x0000010000A0000000000000000000000} \quad = \Gamma_C^2,
\end{aligned}
$$

The 2-round linear characteristic $(\Gamma_P^3 \to \Gamma_C^3)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^3 = \quad & \text{0xE0000000000000000000000000000E00} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x40000000000000000000000000000800} \\
& \qquad \downarrow LT \\
& \text{0x00000000000000000000008000000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x00000000000000000000001000000000} \\
& \qquad \downarrow LT \\
& \text{0x000010000A0000000000000000000000} \quad = \Gamma_C^3,
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^4 \to \Gamma_C^4$) with $p = 1/2 - 2^{-5}$

$$
\begin{aligned}
\Gamma_P^4 = \quad & \text{0x00E0000000000000000000000000000E} \\
& \qquad \downarrow S_5 \qquad\qquad\qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0040000000000000000000000000000008} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000000080000000} \\
& \qquad \downarrow S_6 \qquad\qquad\qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x0000000000000000000000040000000} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000080000002004} \quad = \Gamma_C^4,
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^5 \to \Gamma_C^5$) with $p = 1/2 - 2^{-5}$

$$
\begin{aligned}
\Gamma_P^5 = \quad & \text{0x0E0000000000000000000000000000E0} \\
& \qquad \downarrow S_5 \qquad\qquad\qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0400000000000000000000000000080} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000008000000000} \\
& \qquad \downarrow S_6 \qquad\qquad\qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x0000000000000000000004000000000} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000800000020040} \quad = \Gamma_C^5,
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^6 \to \Gamma_C^6$) with $p = 1/2 - 2^{-5}$

$$
\begin{aligned}
\Gamma_P^6 = \quad & \text{0xE00000000000000000000000000E00} \\
& \qquad \downarrow S_5 \qquad\qquad\qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x4000000000000000000000000000800} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000008000000000} \\
& \qquad \downarrow S_6 \qquad\qquad\qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x0000000000000000000004000000000} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000008000000200400} \quad = \Gamma_C^6,
\end{aligned}
$$

The 2-round linear characteristic $(\Gamma_P^7 \to \Gamma_C^7)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^7 = \quad & \text{0x00E0000000000000000000000000000E} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0040000000000000000000000000008} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000000080000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0000000000000000000000080000000} \\
& \qquad \downarrow LT \\
& \text{0x0040000000000000080000010800A0002} \quad = \Gamma_C^7,
\end{aligned}
$$

The 2-round linear characteristic $(\Gamma_P^8 \to \Gamma_C^8)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^8 = \quad & \text{0x0E0000000000000000000000000000E0} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0400000000000000000000000000080} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000000800000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0000000000000000000000800000000} \\
& \qquad \downarrow LT \\
& \text{0x0400000000000000080000010800A00020} \quad = \Gamma_C^8,
\end{aligned}
$$

The 2-round linear characteristic $(\Gamma_P^9 \to \Gamma_C^9)$ with $p = 1/2 + 2^{-5}$

$$
\begin{aligned}
\Gamma_P^9 = \quad & \text{0xE00000000000000000000000000E00} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x4000000000000000000000000000800} \\
& \qquad \downarrow LT \\
& \text{0x0000000000000000000008000000000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 + 2^{-3} \\
& \text{0x0000000000000000000008000000000} \\
& \qquad \downarrow LT \\
& \text{0x4000000000000080000010800A000200} \quad = \Gamma_C^9,
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^{10} \to \Gamma_C^{10}$) with $p = 1/2 + 2^{-4}$

$$
\begin{aligned}
\Gamma_P^{10} = \ & \text{0x00000000000000000000000000000000E} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 - 2^{-2} \\
& \text{0x00000000000000000000000000000004} \\
& \qquad \downarrow LT \\
& \text{0x00000040000000000000000000008000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x000000D0000000000000000000B000} \\
& \qquad \downarrow LT \\
& \text{0x000810800A30060A400018010A12A002} \ = \Gamma_C^{10},
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^{11} \to \Gamma_C^{11}$) with $p = 1/2 + 2^{-4}$

$$
\begin{aligned}
\Gamma_P^{11} = \ & \text{0x00000000000000000000000000000E0} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 - 2^{-2} \\
& \text{0x00000000000000000000000000000040} \\
& \qquad \downarrow LT \\
& \text{0x00004000000000000000000000080000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x00000D000000000000000000B0000} \\
& \qquad \downarrow LT \\
& \text{0x00810800A30060A400018010A12A0020} \ = \Gamma_C^{11},
\end{aligned}
$$

The 2-round linear characteristic ($\Gamma_P^{12} \to \Gamma_C^{12}$) with $p = 1/2 + 2^{-4}$

$$
\begin{aligned}
\Gamma_P^{12} = \ & \text{0x00000000000000000000000000000E00} \\
& \qquad \downarrow S_5 \qquad\qquad \Pr = 1/2 - 2^{-2} \\
& \text{0x00000000000000000000000000000400} \\
& \qquad \downarrow LT \\
& \text{0x00040000000000000000000000800000} \\
& \qquad \downarrow S_6 \qquad\qquad \Pr = 1/2 - 2^{-3} \\
& \text{0x0000D000000000000000000B00000} \\
& \qquad \downarrow LT \\
& \text{0x0810800A30060A400018010A12A00200} \ = \Gamma_C^{12}.
\end{aligned}
$$