Differential fault attack on KASUMI cipher used in GSM telephony

Article *in* Mathematical Problems in Engineering · June 2014

DOI: 10.1155/2014/251853

Xiaoyang Dong

Tsinghua University

3 PUBLICATIONS **13** CITATIONS

SEE PROFILE

Keting Jia

Tsinghua University

30 PUBLICATIONS **203** CITATIONS

SEE PROFILE

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 251853, 7 pages
http://dx.doi.org/10.1155/2014/251853

Research Article

Differential Fault Attack on KASUMI Cipher Used in GSM Telephony

Zongyue Wang,[1] Xiaoyang Dong,[1] Keting Jia,[2] and Jingyuan Zhao[1]

[1] *Key Laboratory of Cryptologic Technology and Information Security, Shandong University, Ministry of Education, Jinan 250100, China*
[2] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Zongyue Wang; zongyuewang@mail.sdu.edu.cn

Received 20 January 2014; Revised 19 May 2014; Accepted 19 May 2014; Published 5 June 2014

Academic Editor: Wang Xing-yuan

The confidentiality of GSM cellular telephony depends on the security of A5 family of cryptosystems. As an algorithm in this family survived from cryptanalysis, A5/3 is based on the block cipher KASUMI. This paper describes a novel differential fault attack on KAUSMI with a 64-bit key. Taking advantage of some mathematical observations on the FL, FO functions, and key schedule, only one 16-bit word fault is required to recover all information of the 64-bit key. The time complexity is only 2^{32} encryptions. We have practically simulated the attack on a PC which takes only a few minutes to recover all the key bits. The simulation also experimentally verifies the correctness and complexity.

1. Introduction

These years witness the rapid development of computer and network communication. The requirement of privacy and authentication in open access environment promote the developments of cryptography as well. Various cryptosystems have been proposed as well as some new studies [1, 2]. Fast encryption method can be used in real-time communications [3, 4]. GSM (Global System for Mobile Communications) is a widely used real-time communication system which is also the stander for mobile telephony. The confidentiality of GSM depends on the security of A5 family of cryptosystems. The first two members of this family, A5/1 and A5/2, are stream ciphers which were designed 20 years ago in an opaque process and were kept secret until they were reverse engineered in 1999 [5]. Since then, many cryptanalytic results on these two ciphers have been proposed. It becomes clear that A5/2 provides almost no security and A5/1 is too weak to prevent adversary from eavesdropping on GSM conversations [6, 7]. Even emulating a mobile phone to make calls and send text messages is possible [8, 9].

In response to these attacks, GSM association has vowed to switch to the much more secure A5/3 cipher since 2010. A5/3 is a cryptosystem based on block cipher KASUMI [10], which has eight Feistel rounds with a 64-bit block size. KASUMI accepts 128-bit key in the specification, but the key length needs to be reduced in some cases. In practice, A5/3 cryptosystem supports a 64~128-bit session key. We denote KASUMI with 64-bit key by KASUMI-64 and A5/3 with 64-bit session key by A5/3-64, respectively.

Lots of attacks on variants of KASUMI have been proposed in the past years with a variety of techniques [11–14]. Among them, Jia et al. give a result on KASUMI-64 with only 1152 chosen plaintexts and a time complexity of $2^{62.75}$ encryptions. Dunkelman et al. also show that they could derive the complete 128-bit key with data complexity of 2^{26}, 2^{32} encryptions, and 2^{30} bytes of memory under the related key setting [12].

Differential attack was proposed by Biham and Shamir to analyze DES [15]. This powerful method has been successfully applied to evaluate cryptosystems and ciphers in subsequent works [16–18]. Combined with side channel attack and engineering, differential fault analysis (DFA) is a well-known threat to cryptographic devices. Utilizing differential information between correct and faulty ciphertexts, DFA recovers key efficiently. Fault is injected by giving external impact on a device with voltage variation, glitch, laser, and so forth. Since the first DFA on DES proposed by Biham and Shamir [19], this technique has been successfully applied to many other block ciphers, for example, AES [20–23], CLEFIA [24, 25], SM4 [26], and ARIA [27].

In 2011, Jeong et al. proposed the first fault injection attack on A5/3-64 [28]. Their attack is based on the fault assumption in [29], which assumes that the implementation of a symmetric cipher in the PIC assembly language has the following format:

```
movlw       08 h
movwf       RoudCounter
RoundLabel
Call        RoundFunction
decfz       RoundCounter
goto        RoundLabel
```

The RAM variable (RoundCounter) is set to the round number. The adversary may decrease the number of rounds by injecting faults to RoundCounter. With about $2^{45.44}$ KASUMI encryptions and one fault on average, the author recovers a 64-bit session key. However, in many cases, this attack may fail. As a simple example, it is possible to implement block ciphers so that each round is called independently:

```
Call    RoundFunction
Call    RoundFunction
Call    RoundFunction
...
```

In this paper, a novel DFA on KASUMI with a 64-bit key is proposed. The method is also applicable to A5/3-64. Based on some mathematical observations on the FL, FO functions, and the key schedule, we show that only one 16-bit word fault is enough to perform an efficient key recovery with 2^{32} encryptions. We highlight that the attack is practical. The attacking procedure is simulated on a PC where the correct key is recovered in a few minutes. The simulation experimentally verifies the correctness and complexity. Compared with the attack proposed by Kitae Jeong, our method is more flexible and has lower time complexity.

The remainder of the paper is organized as follows. Section 2 gives a brief description of KASUMI. Section 3 shows some important observations useful to our DFA method. The detailed attack procedure is described in Section 4. In Section 5, we show some simulation results. Finally, we conclude this paper in Section 6.

2. Description of KASUMI

As depicted in Figure 1, KASUMI is a Feistel structure with 8 rounds. It works on a 64-bit block and uses a 128-bit key. Each round is made up of an FL function and an FO function. The order of the two functions depends on the round number: in odd numbered rounds the FL function precedes the FO function, whereas in even numbered rounds the FO function precedes the FL function.

FL is a simple key-dependent Boolean function, which accepts XL as well as round key KL as input and output YL (Figure 1(d)). XL, KL, and YL are all 32-bit words which can be divided into two halves. We denote the most significant half by subscript l and the other by subscript r. Subscript i is used to denote the ith round. Then the inputs of the FL function of the ith round are $XL_i = XL_{i,l} \parallel XL_{i,r}$, $KL_i = (KL_{i,1}, KL_{i,2})$ and the output is $YL_i = YL_{i,l} \parallel YL_{i,r}$ ("$\parallel$" is the concatenating operation). FL is defined as follows:

$$\begin{aligned} YL_{i,r} &= \left(\left(XL_{i,l} \wedge KL_{i,1}\right) \lll 1\right) \oplus XL_{i,r}, \\ YL_{i,l} &= \left(\left(YL_{i,r} \vee KL_{i,2}\right) \lll 1\right) \oplus XL_{i,l}, \end{aligned} \tag{1}$$

where the "$\wedge$" and "$\vee$" denote bitwise AND and OR, respectively. "$x \lll i$" implies that x rotates left by i bits.

As shown in Figure 1(b), the FO function is a three-round Feistel structure which consists of three FI functions and key adding stages. A 96-bit round key enters FO function in each round (48 subkey bits KI used in FI and 48 subkey bits KO in the key adding stage). The FI function is another four-round Feistel structure that uses two nonlinear S-boxes S7 and S9 (where S7 is a 7-bit to 7-bit permutation and S9 is a 9-bit to 9-bit permutation). We define half of FI function as $\overline{\text{FI}}$, which is a 16-bit to 16-bit permutation. The structure of FI and $\overline{\text{FI}}$ is illustrated in Figure 1(c).

The key schedule of KASUMI is very simple. More precisely, a 128-bit key is divided into 16-bit words: $(k_1, k_2, \ldots, k_8)$. Round keys are linearly derived from these eight key words (see Table 1). Since the key length needs to be reduced in some cases, the key words should be cyclically repeated to fill 128 bits. The eight key words of KASUMI-64, in particular, are listed as follows: $(k_1, k_2, k_3, k_4, k_1, k_2, k_3, k_4)$.

3. Some Observations of KASUMI

In this section, several observations of KASUMI are given, which are bases of our DFA.

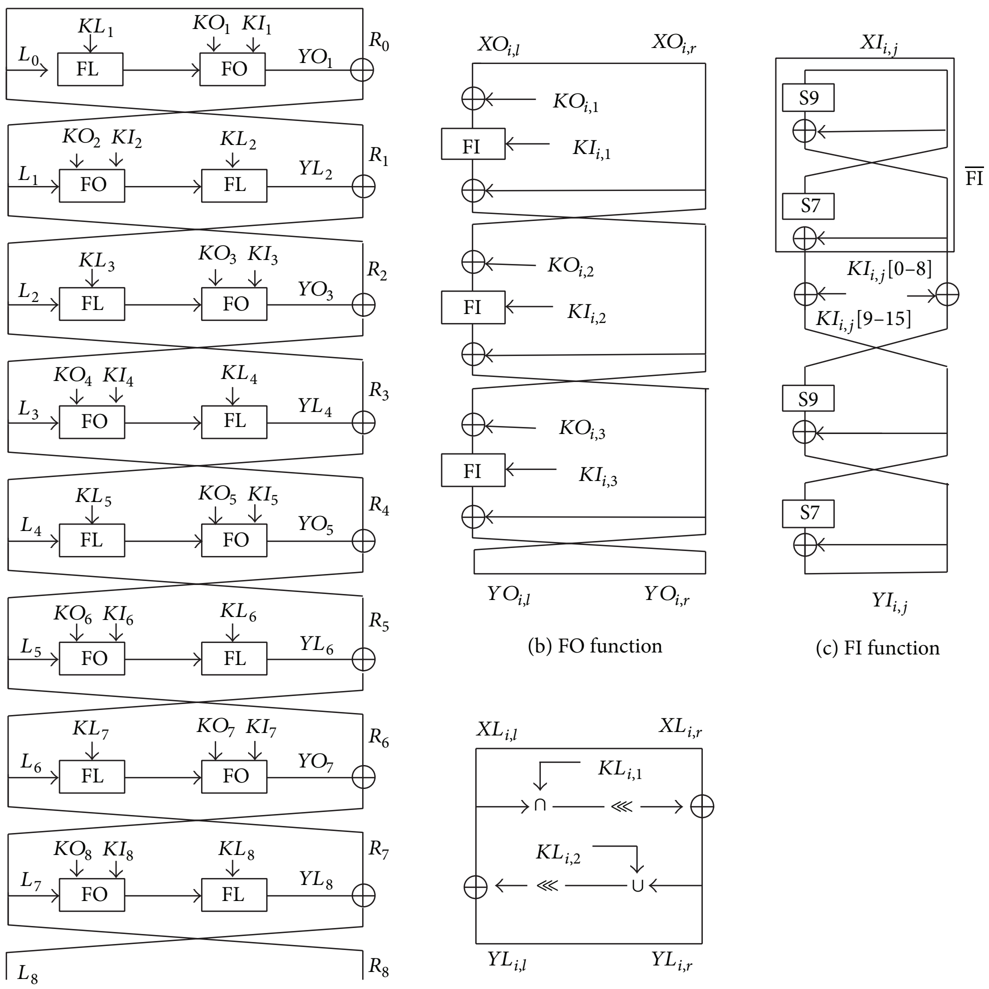
Observation 1 (see [30]). Let X, X' be l-bit values, and $\Delta X = X \oplus X'$. Then there are two difference properties of ADD and OR operations, such that

$$\begin{aligned} (X \wedge K) \oplus \left(X' \wedge K\right) &= \Delta X \wedge K, \\ (X \vee K) \oplus \left(X' \vee K\right) &= \Delta X \oplus (\Delta X \wedge K). \end{aligned} \tag{2}$$

Observation 2. Given the output difference $\Delta Y = \Delta Y_l \parallel \Delta Y_r$ and the key value $KL = (KL_1, KL_2)$ of FL function, the corresponding input difference can be calculated by

$$\begin{aligned} \Delta X_l &= \left(\left(\Delta X_r \oplus \left(\Delta X_r \wedge KL_2\right) \lll 1\right)\right) \oplus \Delta Y_l, \\ \Delta X_r &= \left(\left(\Delta X_l \wedge KL_1\right) \lll 1\right) \oplus \Delta Y_r. \end{aligned} \tag{3}$$

This observation is deduced from Observation 1 and the definition of the FL function easily.



(b) FO function

(c) FI function

(a) KASUMI general structure

(d) FL function

Figure 1: The structure and building blocks of block cipher KASUMI.

Table 1: The key schedule of KASUMI.

Round	$KL_{i,1}$	$KL_{i,2}$	$KO_{i,1}$	$KO_{i,2}$	$KO_{i,3}$	$KI_{i,1}$	$KI_{i,2}$	$KI_{i,3}$
1	$k_1 \lll 1$	k'_3	$k_2 \lll 5$	$k_6 \lll 8$	$k_7 \lll 13$	k'_5	k'_4	k'_8
2	$k_2 \lll 1$	k'_4	$k_3 \lll 5$	$k_7 \lll 8$	$k_8 \lll 13$	k'_6	k'_5	k'_1
3	$k_3 \lll 1$	k'_5	$k_4 \lll 5$	$k_8 \lll 8$	$k_1 \lll 13$	k'_7	k'_6	k'_2
4	$k_4 \lll 1$	k'_6	$k_5 \lll 5$	$k_1 \lll 8$	$k_2 \lll 13$	k'_8	k'_7	k'_3
5	$k_5 \lll 1$	k'_7	$k_6 \lll 5$	$k_2 \lll 8$	$k_3 \lll 13$	k'_1	k'_8	k'_4
6	$k_6 \lll 1$	k'_8	$k_7 \lll 5$	$k_3 \lll 8$	$k_4 \lll 13$	k'_2	k'_1	k'_5
7	$k_7 \lll 1$	k'_1	$k_8 \lll 5$	$k_4 \lll 8$	$k_5 \lll 13$	k'_3	k'_2	k'_6
8	$k_8 \lll 1$	k'_2	$k_1 \lll 5$	$k_5 \lll 8$	$k_6 \lll 13$	k'_4	k'_3	k'_7

$x \lll i$: x rotates left by i bits.
$k'_i = k_i \oplus c_i$, where c_is are fixed constants.

Observation 3. For both S7 and S9, let $s(\cdot)$ be the S-box and consider the following equation:

$$s(x) \oplus s(x \oplus \alpha) = \beta, \tag{4}$$

where α and β are randomly given input and output difference correspondingly. On average, there is a solution x.

Actually, for both S7 and S9, the number of solution of (4) could only be 0 and 2. The probabilities of each case are both 1/2. This property could be verified by traversing every value x under any possible α and β. So on average, for a randomly given pair of α and β, only one solution is found. In practice, we build a look-up table indexed by α and β to help us solve this kind of equation.

Observation 4. Given an input difference Δx and an output difference Δy of $\overline{\text{FI}}$, one could deduce the possible input and output values. On average, there is one input value matching the difference.

$\overline{\text{FI}}$ is made up of an S7 and an S9. From Δx and Δy, we calculate the input and output difference of both S7 and S9. Thus, this observation is derived from Observation 3 normally.

Observation 5. Given an input difference Δx and an output difference Δy under random key KI of the FI function, there are possible input and output values. On average, only one input value can be found under KI.

Given the input difference Δx and the output difference Δy, traverse all input values v, and leave those that satisfy the following equation:

$$\text{FI}(v) \oplus \text{FI}(v \oplus \Delta x) = \Delta y. \quad (5)$$

Then the possible input values are deduced. As there are 2^{16} output differences in total, for any v, the equation holds with probability $1/2^{16}$. Noting that there are also 2^{16} different vs, one could find $2^{16} \times 1/2^{16} = 1$ possible input value on average.

4. DFA on KASUMI

In this section, we describe the DFA on KASUMI in detail, including fault model, attack procedure, and complexity analysis.

4.1. Fault Model and Basic Assumption. As the computing unit of FO and FL function is 16-bit word, the basic storage cell of KASUMI is usually double bytes. So we assume that an attacker can induce a fault to a selected state making a 16-bit word corrupted. The location of the corrupted word may be known. For example, Fukunaga and Takahashi showed that they could control the location of a corrupted byte in [31]. Even if the attacker does not know which word is corrupted, he can repeat injecting until the target 16-bit word corrupted. The assumption is generic and reasonable for devices in which the intermediate values of the encryption are stored.

4.2. General Idea. Only four 16-bit key words are used in KASUMI-64. The general idea is to reduce the number of key candidates by fault injection. More precisely, injecting a 16-bit word fault to the output of the last but one round and making the most of the correct and faulty ciphertexts, the 64-bit key is determined by 32 bits. The possible key space is reduced from 2^{64} to 2^{32}. Then the correct key can be obtained through exhaustive search.

4.3. Attacking Procedure and Complexity Analysis. For better understanding of our method, some notifications are introduced. As illustrated in Figure 2, L and R are the inputs of the last round and C_L and C_R are the ciphertexts. We denote the inputs of FI_i and FL by X_{FI_i} and X_{FL}, respectively. The corresponding outputs are denoted by Y_{FI_i} and Y_{FL}. M, N, and O stand for the intermediate states as shown in Figure 2. Δ is used to define the difference between the correct and faulty values of a state.

Now the attack procedures are described as follows.

Step 1 (obtain the correct and faulty ciphertexts). For a randomly chosen plaintext, obtain the corresponding ciphertext under the unknown key. For the same plaintext, inject a 16-bit word fault to the position as shown in Figure 2, so that the left 16-bit word of L is corrupted. Store the faulty ciphertext. Noting that $L = C_R$, the corrupted value is known.

Step 2 (guess K_2 and K_4 and deduce ΔY_{FI_1}). The injected fault does not affect the value of R. So we have $\Delta R = 0$ and $\Delta Y_{\text{FL}} = \Delta C_L$. For any guesses of K_2 and K_4, as presented in Observation 2, ΔX_{FL} as well as ΔO are deduced. As only the left 16-bit word of L is corrupted, we have $\Delta M = 0$. Thus the input and output differences of FI_2 are both 0. Because

$$\Delta Y_{\text{FI}_1} = \Delta M \oplus \Delta N = \Delta M \oplus \Delta Y_{\text{FI}_2} \oplus \Delta O = \Delta O, \quad (6)$$

ΔY_{FI_1} is determined by the guessing of K_2 and K_4.

Step 3 (match the input and output difference of FI_1 and calculate K_1). From the key schedule of KASUMI, we can see that, in the last round, K_4' is used as KI_1. However, K_4 has been guessed in Step 2. Hence for FI_1, the input and output differences as well as KI_1 are all determined. As shown in Observation 5, there is a value X_{FI_1} matching the input and output difference on average. Since $L = C_R$ and $(K_1 \lll 5) = L_L \oplus X_{\text{FI}_1}$, the possible K_1s are calculated.

Step 4 (deduce the correct and corrupted inputs of FI_3, and determine K_3). K_1, K_2, and K_4 have been guessed or deduced in the above steps. So the correct and corrupted inputs of FI_3 are known. As shown in Figure 3, Q is only affected by the input value of FI_3. So Q is calculated and the input difference of the $\overline{\text{FI}}$ is deduced. Note that the output difference of $\overline{\text{FI}}$ is known by ΔX_{FL} which has been calculated in Step 2. Through Observation 4, the possible input value $X_{\overline{\text{FI}}}$ is known. So KI_3 is obtained by $KI_3 = X_{\overline{\text{FI}}} \oplus Q$. Indeed, KI_3 is K_3' in the last round. Thus K_3 is also determined. Until now, all the information of key is determined by the guessing of K_2 and K_4.

Step 5 (verify the correctness of the guessed key). Encrypt the plaintext with the key obtained in the above steps and check the correctness. If the key is not right, go back to Step 2 with another guess of K_2 and K_4.

We will order the above description in Algorithm 1 where a look-up table indexed by the output difference of FI_1 is established before the guessing of K_2 to reduce the computing complexity.

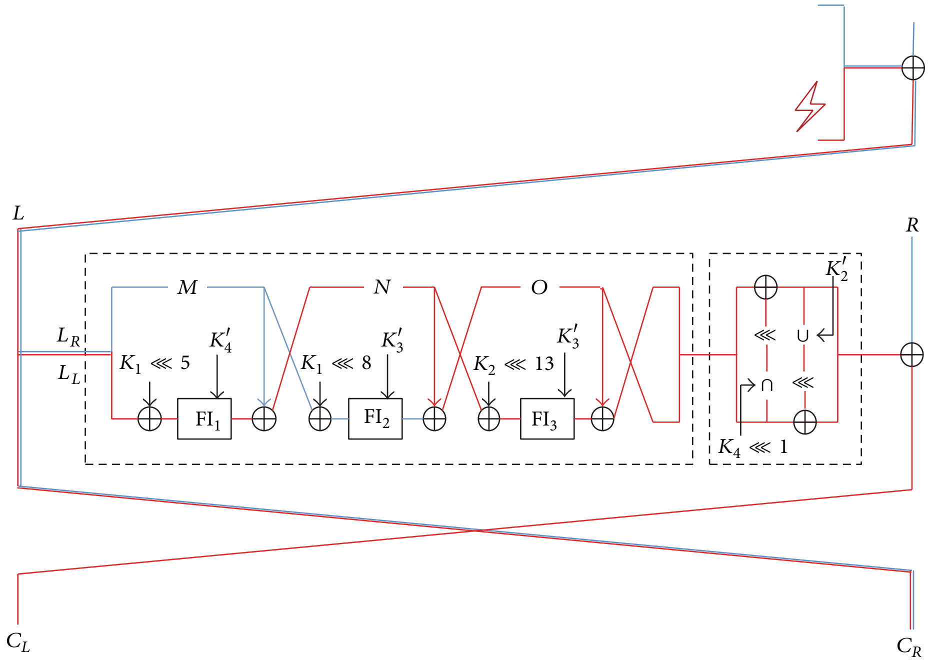


FIGURE 2: DFA on KASUMI. (Red lines are affected by the fault while the blues are not.)

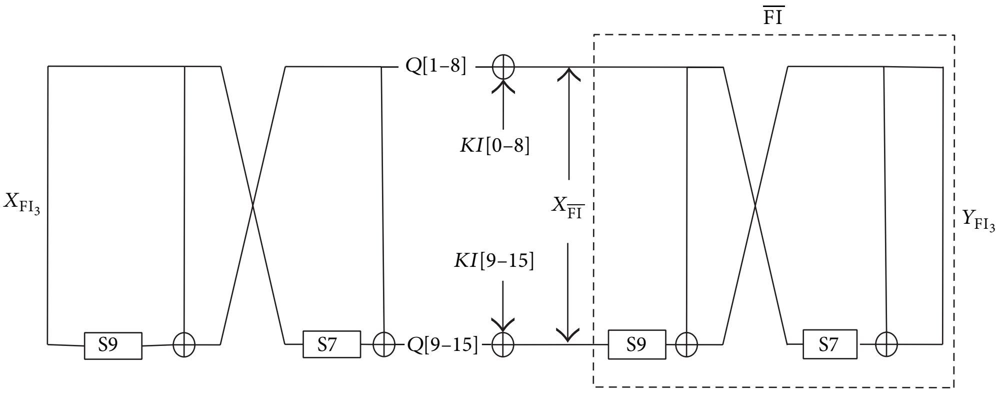


FIGURE 3: FI_3 in the last round.

To evaluate the complexity of Algorithm 1, we count the number of KASUMI encryptions. As we do checking operation for $2^{16+16} = 2^{32}$ times on average, the computing complexity is 2^{32} encryptions. The memory requirement is 2^{17} bytes since a table containing 2^{16} 16-bit words should be stored.

5. Simulation Results

We simulate our DFA on KASUMI-64 with a random key for 1000 times. Each time is denoted as a sample. For each sample, the correct key can be recovered within a few minutes. The number of checking operations for every sample is illustrated in Figure 4. All the numbers are around 2^{32} and the correctness and complexity of our method are verified in practice.

6. Conclusion

This paper describes a DFA attack on KASUMI-64 which is the base of A5/3 cryptosystem used in GSM telephony. We show that only one 16-bit word fault is enough to perform a successful key recovery attack. More impressively, both the computing and memory complexity are practical and the secret key can be recovered in a few minutes. The correctness and complexity are further verified by the simulation results. We emphasize that when applying KASUMI-64, the last two rounds should be specially designed to protect against fault injection. This paper also demonstrates the efficiency

```
Input: $C_L, C_R, \Delta C_L\ \Delta C_R$
Output: $(K_1, K_2, K_3, K_4)$
$\Delta L_L \parallel \Delta L_R = \Delta C_R$;
for all possible values of $K_4$ do                              # $2^{16}$ times
    for $i = 0; i < 2^{16}; i{+}{+}$ do
        $T_i = \Phi$;                                            # Initialize $T_i$ as an empty set
    end for
    for $i = 0; i < 2^{16}; i{+}{+}$ do                          # Construction of the Table
        index $\leftarrow$ $\mathrm{FI}(i, K_4') \oplus \mathrm{FI}(i \oplus \Delta L_L, K_4')$;
        $T_{\text{index}} \leftarrow T_{\text{index}} \cup \{i\}$;
    end for
    for all possible values of $K_2$ do                          # $2^{16}$ times
        Compute the differences $\Delta X_{\mathrm{FL}}$.        # Observation 2
        for all $X_{\mathrm{FI}_1}$ in $T_{\Delta Y_{\mathrm{FI}_1}}$ do    # Once on average
            $K_1 \leftarrow (L_L \oplus X_{\mathrm{FI}_1}) \ggg 5$;
            Deduce $Q$ and $\Delta Q$ as shown in Figure 3.
            Get all possible inputs of $\overline{\mathrm{FI}}$.   # Observation 4
            for all possible $X_{\overline{\mathrm{FI}}}$ do      # Once on average
                $K_3' \leftarrow Q \oplus X_{\overline{\mathrm{FI}}}$;
                if $(K_1, K_2, K_3, K_4)$ is the right key then   # Checking operation
                    return $(K_1, K_2, K_3, K_4)$;
                end if
            end for
        end for
    end for
end for
```

Algorithm 1: DFA on KASUMI cipher.

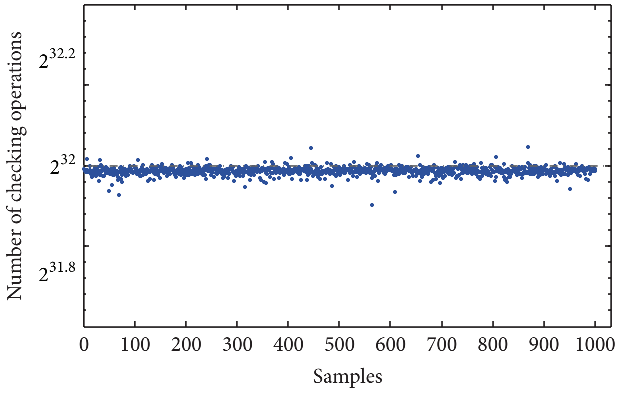


Figure 4: Simulation result.

of differential fault attack. When designing and realizing cryptosystems, this new kind of attack should also be taken into account.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by China's 973 Program (Grant no. 2013CB834205) and National Natural Science Foundation of China (Grant no. 61133013).

References

[1] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.

[2] X. Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615–621, 2010.

[3] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

[4] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, 2010.

[5] M. Briceno, I. Goldberg, and D. Wagner, A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms, 1999.

[6] K. Nohl, *Attacking Phone Privacy*, Black Hat USA, 2010.

[7] K. Nohl and C. Paget, "GSM: SRSLY?" in *Proceedings of the 26th Chaos Communication Congress*, 2009.

[8] K. Nohl and S. Munaut, "Wideband GSM sniffing," in *Proceedings of the 27th Chaos Communication Congress*, 2010.

[9] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of GSM encrypted communication," in *Advances in Cryptology—CRYPTO 2003*, vol. 2729 of *Lecture Notes in Computer Science*, pp. 600–616, Springer, Berlin, Germany, 2003.

[10] 3rd generation partnership project, technical specication group services and system aspects, 3G security, specification of the 3GPP condentiality and integrity algorithms, document 2: KASUMI specificationV3. 1. 1, 2001.

[11] E. Biham, O. Dunkelman, and N. Keller, "A related-key rectangle attack on the full kasumi," in *Advances in Cryptology—ASIACRYPT 2005*, vol. 3788 of *Lecture Notes in Computer Science*, pp. 443–461, Springer, Berlin, Germany, 2005.

[12] O. Dunkelman, N. Keller, and A. Shamir, "A practical-time related-key attack on the kasumi cryptosystem used in GSM and 3G telephony," in *Advances in Cryptology—CRYPTO 2010*, vol. 6223 of *Lecture Notes in Computer Science*, pp. 393–410, Springer, Berlin, Germany, 2010.

[13] J. S. Kang, S. U. Shin, D. Hong, and O. Yi, "Provable security of KASUMI and 3GPP encryption mode $f8$," in *Advances in Cryptology—ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 255–271, Springer, Berlin, Germany, 2001.

[14] K. Jia, C. Rechberger, and X. Wang, "Green cryptanalysis: meet-in-the-middle keyrecovery for the full kasumi cipher," Tech. Rep. 2011/466, 2011.

[15] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[16] M. Wang, "Differential cryptanalysis of reduced-round present," in *Progress in cryptology—AFRICACRYPT 2008*, vol. 5023 of *Lecture Notes in Computer Science*, pp. 40–49, Springer, Berlin, Germany, 2008.

[17] Y. Q. Zhang and X. Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, vol. 273, pp. 329–351, 2014.

[18] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Optics Communications*, vol. 284, no. 16-17, pp. 3895–3903, 2011.

[19] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," in *Advances in Cryptology—CRYPTO 1997*, Lecture Notes in Computer Science, pp. 513–525, Springer, Berlin, Germany, 1997.

[20] G. Piret and J. J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," in *Cryptographic Hardware and Embedded Systems-CHES 2003*, vol. 2779 of *Lecture Notes in Computer Science*, pp. 77–88, Springer, Berlin, Germany, 2003.

[21] C. H. Kim and J. J. Quisquater, "New differential fault analysis on AES key schedule: two faults are enough," in *Smart Card Research and Advanced Applications*, vol. 5189 of *Lecture Notes in Computer Science*, pp. 48–60, Springer, Berlin, Germany, 2008.

[22] C. N. Chen and S. M. Yen, "Differential fault analysis on AES key schedule and some countermeasures," in *Information Security and Privacy*, vol. 2727 of *Lecture Notes in Computer Science*, pp. 118–129, Springer, Berlin, Germany, 2003.

[23] C. H. Kim, "Differential fault analysis of AES: toward reducing number of faults," *Information Sciences*, vol. 199, pp. 43–57, 2012.

[24] H. Chen, W. Wu, and D. Feng, "Differential fault analysis on CLEFIA," in *Information and Communications Security*, vol. 4861 of *Lecture Notes in Computer Science*, pp. 284–295, Springer, Berlin, Germany, 2007.

[25] J. Takahashi and T. Fukunaga, "Improved differential fault analysis on CLEFIA," in *Proceedings of the 5th International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC '08)*, pp. 25–34, IEEE, August 2008.

[26] R. Li, B. Sun, C. Li, and J. You, "Differential fault analysis on SMS4 using a single fault," *Information Processing Letters*, vol. 111, no. 4, pp. 156–163, 2011.

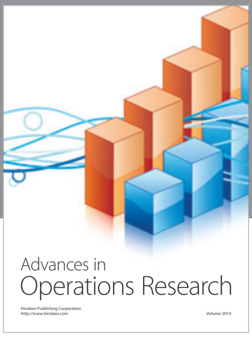
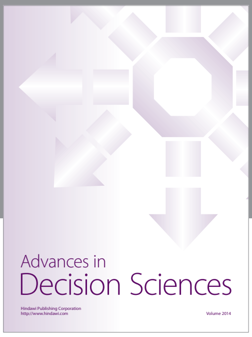
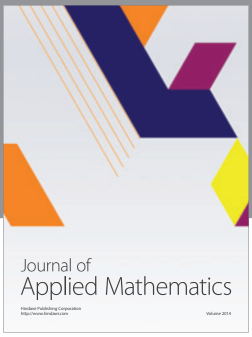
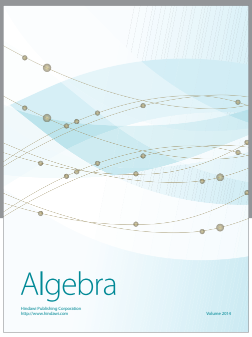
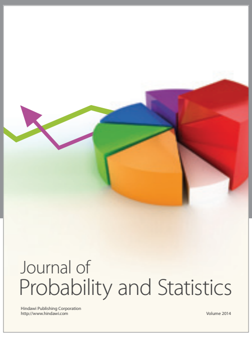
[27] W. Li, D. Gu, and J. Li, "Differential fault analysis on the ARIA algorithm," *Information Sciences*, vol. 178, no. 19, pp. 3727–3737, 2008.

[28] K. Jeong, Y. Lee, J. Sung, and S. Hong, "Fault injection attack on A5/3," in *Proceedings of the 9th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA '11)*, pp. 300–303, May 2011.

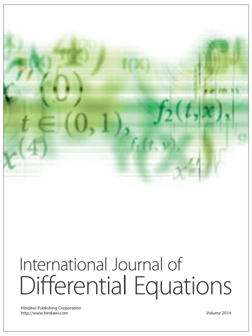
[29] H. Choukri and M. Tunstall, "Round reduction using faults," in *Fault Diagnosis and Tolerance in Cryptography (FDTC)*, vol. 5, pp. 13–24, 2005.

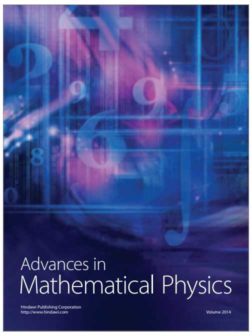
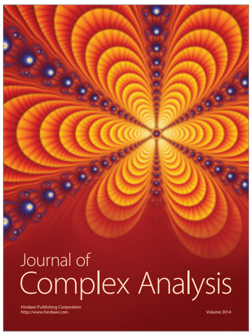
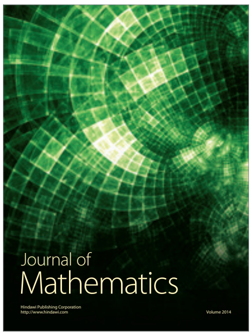
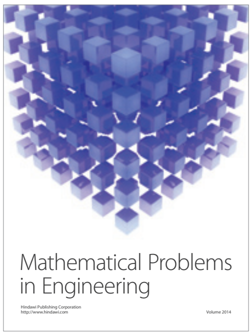
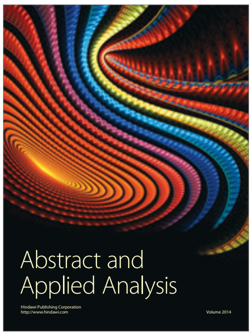
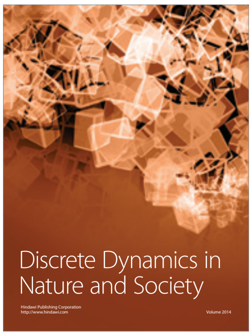
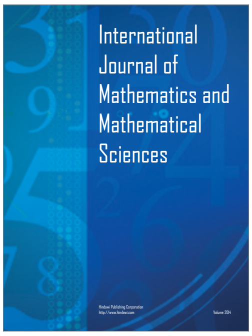
[30] U. Kühn, "Improved cryptanalysis of MISTY1," in *Fast Software Encryption*, Lecture Notes in Computer Science, pp. 61–75, Springer, Berlin, Germany, 2002.

[31] T. Fukunaga and J. Takahashi, "Practical fault attack on a cryptographic LSI with ISO/IEC 18033-3 block ciphers," in *Proceedings of the 6th International Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC '09)*, pp. 84–92, IEEE, September 2009.

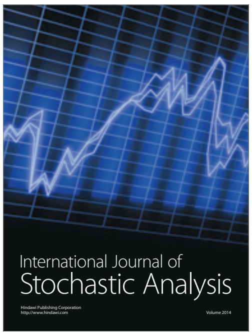

Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

Algebra

Journal of
Probability and Statistics

The Scientific
World Journal


International Journal of
Differential Equations

International Journal of
Combinatorics

Advances in
Mathematical Physics

Journal of
Complex Analysis

Journal of
Mathematics

Mathematical Problems
in Engineering

Abstract and
Applied Analysis

Discrete Dynamics in
Nature and Society

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces

International Journal of
Stochastic Analysis

Journal of
Optimization