Security Analysis of the Khudra Lightweight Cryptosystem in the Vehicular Ad-hoc Networks

Wei Li [1, 2, 3, 4, 5], Chenyu Ge [1], Dawu Gu [2], Linfeng Liao [1], Zhiyong Gao [1], Xiujin Shi [1], Ting Lu [1], Ya Liu [6, 2], Zhiqiang Liu [2]

[1]School of Computer Science and Technology, Donghua University, Shanghai, 201620, China
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[3] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
[4] Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai, 200240, China
[5] Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai, 200240, China
[6] Department of Computer Science and Engineering, University of Shanghai for Science and Technology Shanghai, 200093, China
[e-mail: sxj@dhu.edu.cn]
*Corresponding author: Xiujin Shi

Received December 16, 2016; revised September 5, 2017; accepted January 7, 2018; Published July 31, 2018

Abstract

With the enlargement of wireless technology, vehicular ad-hoc networks (VANETs) are rising as a hopeful way to realize smart cities and address a lot of vital transportation problems such as road security, convenience, and efficiency. To achieve data confidentiality, integrity and authentication applying lightweight cryptosystems is widely recognized as a rather efficient approach for the VANETs. The Khudra cipher is such a lightweight cryptosystem with a typical Generalized Feistel Network, and supports 80-bit secret key. Up to now, little research of fault analysis has been devoted to attacking Khudra. On the basis of the single nibble-oriented fault model, we propose a differential fault analysis on Khudra. The attack can recover its 80-bit secret key by introducing only 2 faults. The results in this study will provides vital references for the security evaluations of other lightweight ciphers in the VANETs.

Keywords: Vehicular Ad-hoc Networks, Khudra, Lightweight Cryptosystem, Differential Fault Analysis

This work is supported by the National Natural Science Foundation of China under Grant No. 61772129, No. 61472250, No. 61402288, No. 61672347, No.61402286 and No. 61572192, Shanghai Natural Science Foundation under Grant No. 15ZR1400300 and No. 16ZR1401100, Innovation Program of Shanghai Municipal Education Commission under Grant No. 14ZZ066, National Key Basic Research Program of China under Grant No. 2013CB338004, Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security under Grant No. AGK201703, Opening Project of Shanghai Key Laboratory of Scalable Computing and Systems, National Cryptography Development Fund, and the Fundamental Research Funds for the Central Universities.

http://doi.org/10.3837/tiis.2018.07.023 **ISSN : 1976-7277**

1. Introduction

Vehicular Ad-hoc Networks (VANETs) comprise vehicle-to-vehicle and vehicle-to--infrastructure communications based on the wireless local area network technologies, aiming to provide a wide spectrum of safety and comfort applications to drivers and passengers. It has been immensely successful and naturally attracted considerable attention from both academia and industry since its introduction [1]. However, the perfect composition of airborne computers and location devices, raises formidable research challenges. VANETs are networks with high dynamic topology and their communication is vulnerable to attacks. For instance, it is necessary to ensure that the vital information can't be interrupted by an attacker; similarly, the system should be helpful to build the drivers' responsibility; but at the same time, it should protect the privacy of the drivers and passengers as much as possible. Nodes in VANETs should be confident that each communication has been started from a trustworthy source node and messages are not varied by malicious vehicles. Although these issues seem similar to those used in traditional communication networks, there are individual characteristic for vehicular networks. The selforgnazied nature of the networks, the mobility of the vehicles, the relevance of their loations, and the irregular connectivity among nodes can lead to a varity of security matters [2-4]. Due to the limitations of running capableness, power provison and memory space of devices in vehicles, classical cryptosystems cannot play direct roles in lots of security actions such as digital signature, message authentication, encryption and decryption, etc. It is very critical to realize efficient cryptosystems in VANETs, i.e., lightweight cryptosystems are mostly desired [5-16]. Thus, application of the lightweight cryptosystems can not only allow lower energy consumption for devices, but also provide more network links with devices.

As an active side channel attack technique, fault analysis can exploit easily accessible information like input-output behavior under malfunctions, magnify and estimate the leaked information by means of mathematical cryptanalysis [16-18]. In 1996, Boneh et al. proposed a fault analysis on RSA by exploiting the faulty calculations [16,17]. Later differential fault analysis (DFA) was presented to break DES and AES [18-21]. The attackers can inject faults to the running procedure of a cryptosystem by laser, electromagnetic and voltage interference in the hardware implementation, or alter the internal state of the code in the software implementation. They are often much more powerful than the classical cryptanalysis. Usually only a few faults suffice to break a cryptosystem [22-27].

2. Related work

The Khudra lightweight cryptosystem, proposed by Kolay et al. in 2014, has a good compact hardware implementation and maintains good software-friendly features [28]. Khudra has a 64-bit block size and supports a 80-bit secret key. It is based on a generalized type-2 Feistel Network structure with 18 rounds and the whitening layers. Since its introduction, Khudra has been the target of classical cryptanalytic efforts [28-32]. The designers of Khudra took many cryptanalytic techniques into account, such as differential cryptanalysis, linear cryptanalysis, impossible differential attack, differential-linear attack, algebraic attack, boomerang type attacks, slide key attack and related key attack, etc [28]. Then Tolba et al. made use of an offline independently distinguisher in an online phase and presented a meet-in-the-middle attack on 13 rounds and 14 rounds, respectively [29]. Later

Özen et al. improved the above 14-round meet-in-the-middle attack and reduced the memory complexity from $2^{64.8}$ to $2^{32.8}$. Furthermore, they applied a guess-and-determine attack on the same 14 rounds [30]. In 2015, Ma et al. attacked 16-round Khudra without whitening key by computing the minimum number of active *F*-functions in differential characteristics of the related-key setting [31]. Later Yang et al. proposed a related-key impossible differential analysis to attack the full-round Khudra without whitening keys [32]. Up to now, little study has been published concerning the Khudra cryptosystem against fault analysis.

In the literature, the previous differential fault analysis targets on the last several rounds of cyrptosystems [23-32]. Their basic principle is to derive the secret key by calculating the differential relationship of S-boxes resulting from a correct operation and a faulty operation. Different from the structure of other lightweight cryptosystems with Generalized Feistel Networks, every round function consists of 6-rounds S-boxes layers to provide nonlinearity. It increases the attacking difficulties in computing the input differences and output differences of S-boxes after 6-round diffusions. Furthermore, since adding protections from fault attack increasing the processing consumptions, some countermeasures are suggested to protect only the last several rounds. In the real applications of VANETs, random faults can be injected into deeper rounds of the cryptosystem. In this point, it is important to investigate the deepest rounds of Khudra with a few faults.

This paper proposes a differential fault analysis on the full 6-round function by injecting only 2 faults into the antepenultimate round of Khudra. The attackers only inject the faults out of the F-function, and depend on the ciphertext difference to derive the accurate locations of faults. Moreover, they can take advantage of 2 faults to the most extent. Hence, it not only decreases the number of faults, but also improves the efficiency of injecting faults. To the best of our knowledge, this is the first work that a differential fault attack on Khudra has been successfully put into practice. Compared with the classical cryptanalysis, differential fault attack on Khudra has a good performance in data complexity, time complexity and memory complexity, as **Table 1** shows.

Table 1. Cryptanalysis of Khudra

Method	Whitening layer	Rounds	Complexity			Ref.
			Data	Time	Memory	
Meet-in-the-middle attack	Yes	13	$2^{51.00}$	$2^{66.00}$	$2^{64.8}$	[15]
		14	$2^{51.00}$	$2^{66.19}$	$2^{32.80}$	[16]
Guess-and-determine attack	Yes	14	$2^{1.00}$	$2^{64.00}$	\	[16]
Related-key rectangle attack	No	16	$2^{57.82}$	$2^{78.68}$	\	[17]
Related-key impossible differential attack	Yes	18	$2^{63.00}$	$2^{64.46}$	$2^{64.00}$	[18]
Differential fault analysis	Yes	18	$2^{1.58}$	$2^{20.17}$	$2^{20.00}$	This paper

The remainder is organized as follows. Section 3 briefly describes the specification of Khudra. Section 4 proposes our differential fault analysis to break Khudra. Section 5 and 6 calculates the attacking complexity and summarizes the experimental results. The last section concludes the paper.

3. Description of Khudra

3.1 Structure

Khudra is an 18-round lightweight block cipher with a 64-bit plaintext size and a 80-bit key size. It employs a Generalized type-2 Feistel Structure as **Fig. 1** shows. The plaintext is divided into four 16-bit branches. To deal with these branches in every round, Khudra has two 16×16 *F*-functions in each round. The *F*-function consists of six substitution layers using two 4×4 S-boxes in each round to provide nonlinearity. This kind of double layer structure makes Khudra more difficult to attack. The whitening keys added at last make sure that the intermediate states are inaccessible for the attackers.

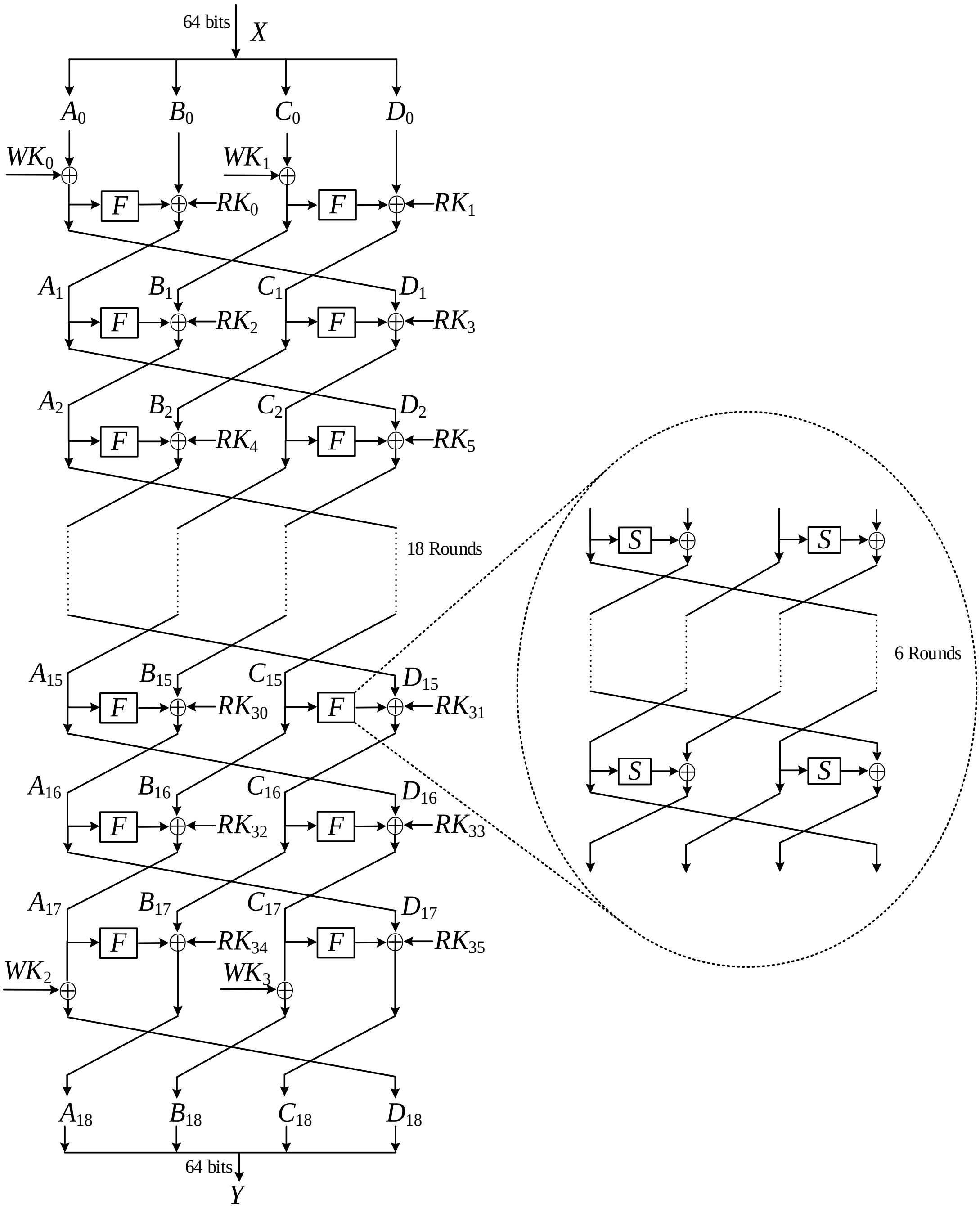


Fig. 1. Structure of Khudra

The detailed encryption of Khudra is presented as **Table 2** shows. The decryption shares the same structure with the encryption, except for the reverse subkeys with the reverse order.

Table 2. Encryption of Khudra

Input: X, K **Output:** Y
$A_0 \| B_0 \| C_0 \| D_0 = X$ $WK_0 \| WK_1 \| WK_2 \| WK_3 \| WK_4 = K$ $A \| B \| C \| D = X$ $C_0 = C_0 \oplus WK_1$ **for** i**=0 to 17 do** $A_{i+1} = F(A_i) \oplus B_i \oplus RK_{2i}$ $B_{i+1} = C_i$ $C_{i+1} = F(C_i) \oplus D_i \oplus RK_{2i+1}$ $D_{i+1} = A_i$ $D_{18} = D_{18} \oplus WK_2$ $B_{18} = B_{18} \oplus WK_3$ $Y = A_{18} \| B_{18} \| C_{18} \| D_{18}$

As the input of a key schedule, the secret key K produces round keys for each round and whitening keys as **Table 3** shows.

Table 3. Key schedule of Khudra

Input: K **Output:** RK, WK
$WK_0 \| WK_1 \| WK_2 \| WK_3 \| WK_4 = K$ **for** w**=0 to 35 do** $RC_w = 0_{(1)} \| w_{(6)} \| 0_{(2)} \| w_{(6)} \| 0_{(1)}$ $RK_w = WK_{w \bmod 5} \oplus RC_w$

3.2 Notations

The notations of Khudra and its analysis are described as **Table 4** shows.

Table 4. Notations of Khudra

Notations	Description
$A_i, B_{i+1}, C_i, D_{i+1}$	The right inputs of the F-function in round i with $0 \le i \le 17$
$\Delta A_i^*, \Delta B_{i+1}^*, \Delta C_i^*, \Delta D_{i+1}^*$	The difference inputs of the F-function in round i with $0 \le i \le 17$ when the fault is injected in A_{15}
$\Delta A_i', \Delta B_{i+1}', \Delta C_i', \Delta D_{i+1}'$	The difference inputs of the F-function in round i with $0 \le i \le 17$ when the fault is injected in C_{15}
$X = A_0 \| B_0 \| C_0 \| D_0$	The plaintext
$Y = A_{18} \| B_{18} \| C_{18} \| D_{18}$	The right ciphertext
Y^*, Y'	The faulty ciphertext when the fault is injected in A_{15} and C_{15}, respectively

K, WK_l, RC_w	The secret key, the whitening keys and the round constant, with $0 \le l \le 4$ and $0 \le w \le 35$
(m)	The bit number of the constant m, with $1 \le m \le 6$
Encryption(X, K)	The encryption procedure

4. Different Fault Analysis on Khudra

4.1 Fault Model

The fault model is the chosen plaintext attack. Furthermore, the attackers can induce a 4-bit error to one layer. However, the location and the value of this nibble in this layer are both unknown. As for the attack, they can analyze a fault occurring near the end of the cryptosystem and assume the general random fault model where the fault modifies the processed data in a random way. Actually, the attackers can inject one-bit error. It does not influence the attacking procedure.

4.2 Attacking Procedure

This section proposes a novel differential fault analysis to break Khudra. **Table 5** shows the algorithm of our attack as follows:

Table 5. Differential fault analysis of Khudra

Input: X, Y, Y^*, Y'
Output: K
$\Delta A_{18}^* \parallel \Delta B_{18}^* \parallel \Delta C_{18}^* \parallel \Delta D_{18}^* = Y \oplus Y^*$
$\Delta A_{18}' \parallel \Delta B_{18}' \parallel \Delta C_{18}' \parallel \Delta D_{18}' = Y \oplus Y'$
for t_1=0 to 2^{16}-1 do
if $F(t_1) \oplus F(t_1 \oplus \Delta D_{18}^*) = \Delta A_{18}^*$
$A_{17} = t_1$
if $F(t_1) \oplus F(t_1 \oplus \Delta B_{18}') = \Delta C_{18}'$
$C_{17} = t_1$
$WK_2 = A_{17} \oplus D_{18}$
$WK_3 = C_{17} \oplus B_{18}$
for t_2=0 to 2^{16}-1 do
if $F(t_2) \oplus F(t_2 \oplus \Delta B_{18}^*) = F(C_{17}) \oplus F(C_{17} \oplus \Delta B_{18}^*) \oplus \Delta C_{18}^*$
$A_{15} = t_2$
if $F(t_2) \oplus F(t_2 \oplus \Delta D_{18}') = F(A_{17}) \oplus F(A_{17} \oplus \Delta D_{18}') \oplus \Delta A_{18}'$
$C_{15} = t_2$
for t_3=0 to 2^{16}-1 do
if $F(t_3) \oplus F(t_3 \oplus F(C_{17}) \oplus F(C_{17} \oplus \Delta B_{18}^*) \oplus \Delta C_{18}^*) = \Delta D_{18}^*$ **and** $F(t_3) \oplus C_{15} = RC_{32} \oplus D_{18}$
$A_{16} = t_3$
if $F(t_3) \oplus F(t_3 \oplus F(A_{17}) \oplus F(A_{17} \oplus \Delta D_{18}') \oplus \Delta A_{18}') = \Delta B_{18}'$ **and** $F(t_3) \oplus A_{15} = RC_{33} \oplus B_{18}$
$C_{16} = t_3$
$WK_0 = RC_{35} \oplus A_{16} \oplus F(C_{17}) \oplus C_{18}$
$WK_4 = RC_{34} \oplus C_{16} \oplus F(A_{17}) \oplus A_{18}$
for t_4=0 to 2^{16}-1 do
$K = WK_0 \parallel t_4 \parallel WK_2 \parallel WK_3 \parallel WK_4$

if $Y = \text{Encryption}(X, K)$
$WK_1 = t_4$
$K = WK_0 \| WK_1 \| WK_2 \| WK_3 \| WK_4$

To recover all whitening keys, the detailed attacking steps are listed as follows:

Step 1. A correct ciphertext Y is derived when an arbitrary plaintext X is encrypted with a secret key K.

Step 2. This step aims at recovering the whitening keys WK_2 and WK_3. The fault injection targets at A_{15}, B_{15}, C_{15}, or D_{15} in the 15^{th} round. Any change of one nibble provokes a series of XOR-differences in the last three rounds as follows:

$$\Delta A_{16}, \Delta D_{16}, \Delta A_{17}, \Delta C_{17}, \Delta D_{17}, \Delta A_{18}, \Delta B_{18}, \Delta C_{18}, \Delta D_{18},$$

$$\Delta A_{16}, \Delta A_{17}, \Delta D_{17}, \Delta A_{18}, \Delta B_{18}, \Delta C_{18}, \Delta D_{18},$$

$$\Delta B_{16}, \Delta C_{16}, \Delta A_{17}, \Delta B_{17}, \Delta C_{17}, \Delta A_{18}, \Delta B_{18}, \Delta C_{18}, \Delta D_{18},$$

or

$$\Delta C_{16}, \Delta B_{17}, \Delta C_{17}, \Delta A_{18}, \Delta B_{18}, \Delta C_{18}, \Delta D_{18}.$$

These alter an original ciphertext into a faulty ciphertext. **Table 6** shows the relations of the fault locations of the F-function in the 15^{th} round and the affected *j-th* nonzero nibble in the ciphertext difference with $0 \le j \le 15$. Thus, the attackers can depend on the number and locations of nonzero nibbles of the ciphertext differences to derive the fault location in each register.

Table 6. The relationship between the fault locations and the affected nibbles of the ciphertext difference

The fault location on each register	The nibble in the 15^{th} round	The *j-th* nonzero nibbles of the ciphertext difference
A_{15}	0	0,1,2,3,4,8,9,10,11,12,13,14,15
	1	0,1,2,3,5,8,9,10,11,12,13,14,15
	2	0,1,2,3,6,8,9,10,11,12,13,14,15
	3	0,1,2,3,7,8,9,10,11,12,13,14,15
B_{15}	4	0,1,2,3,8,12,13,14,15
	5	0,1,2,3,9,12,13,14,15
	6	0,1,2,3,10,12,13,14,15
	7	0,1,2,3,11,12,13,14,15
C_{15}	8	0,1,2,3,4,5,6,7,8,9,10,11,12
	9	0,1,2,3,4,5,6,7,8,9,10,11,13
	10	0,1,2,3,4,5,6,7,8,9,10,11,14
	11	0,1,2,3,4,5,6,7,8,9,10,11,15
D_{15}	12	0,4,5,6,7,8,9,10,11
	13	1,4,5,6,7,8,9,10,11
	14	2,4,5,6,7,8,9,10,11
	15	3,4,5,6,7,8,9,10,11

The relationship between input differences and output differences of the *F*-function are as follows:

$$\Delta A_{i+1} = F(A_i) \oplus F(A_i \oplus \Delta A_i) \oplus \Delta B_i,$$
$$\Delta C_{i+1} = F(C_i) \oplus F(C_i \oplus \Delta C_i) \oplus \Delta D_i,$$

With the help of a pair of right and faulty ciphertexts, the relationships among these differences of the *F*-functions are defined in the last round. When the fault is injected in A_{15}, the attackers can deduce the value of A_{17} depending on

$$\Delta A_{18} = F(A_{17}) \oplus F(A_{17} \oplus \Delta A_{17}) \oplus \Delta B_{17} \oplus \Delta RK_{34}$$
$$= F(A_{17}) \oplus F(A_{17} \oplus \Delta D_{18}),$$

where

$$\Delta A_{18} \| \Delta B_{18} \| \Delta C_{18} \| \Delta D_{18} = \Delta Y,$$
$$\Delta A_{17} = \Delta D_{18},$$
$$\Delta C_{17} = \Delta B_{18},$$
$$\Delta B_{17} = \Delta RK_{34} = 0.$$

When the fault is injected in C_{15}, the attackers can deduce the value of C_{17} depending on

$$\Delta C_{18} = F(C_{17}) \oplus F(C_{17} \oplus \Delta C_{17}) \oplus \Delta D_{17} \oplus \Delta RK_{35}$$
$$= F(C_{17}) \oplus F(C_{17} \oplus \Delta B_{18}),$$

where

$$\Delta A_{18} \| \Delta B_{18} \| \Delta C_{18} \| \Delta D_{18} = \Delta Y,$$
$$\Delta A_{17} = \Delta D_{18},$$
$$\Delta C_{17} = \Delta B_{18},$$
$$\Delta D_{17} = \Delta RK_{35} = 0.$$

The values of WK_2 or WK_3 in the last round can be deduced as follows:

$$WK_2 = A_{17} \oplus A_{18},$$
$$WK_3 = C_{17} \oplus C_{18}.$$

Step 3. No faults are induced in this step. The attackers can depend on the faults in step 2 to get another two whitening keys. In conjunction with the ciphertext difference, A_{16} and C_{16} can be deduced. When the fault is injected in A_{15}, the following equation is helpful to deduce the value of A_{16}:

$$\Delta A_{17} = \Delta D_{18}$$
$$= F(A_{16}) \oplus F(A_{16} \oplus \Delta A_{16}) \oplus \Delta B_{16}$$
$$= F(A_{16}) \oplus F(A_{16} \oplus \Delta A_{16})$$
$$= F(A_{16}) \oplus F(A_{16} \oplus \Delta D_{17})$$
$$= F(A_{16}) \oplus F(A_{16} \oplus F(C_{17}) \oplus F(C_{17} \oplus \Delta C_{17}) \oplus \Delta C_{18})$$
$$= F(A_{16}) \oplus F(A_{16} \oplus F(C_{17}) \oplus F(C_{17} \oplus \Delta B_{18}) \oplus \Delta C_{18}),$$

where

$$\Delta B_{16} = 0,$$
$$\Delta A_{17} = \Delta D_{18},$$
$$\Delta C_{17} = \Delta B_{18}.$$

Similarly, when the fault is injected in C_{15}, the following equation is helpful to deduce the value of C_{16}:

$$\begin{aligned}\Delta C_{17} &= \Delta B_{18} \\ &= F(C_{16}) \oplus F(C_{16} \oplus \Delta C_{16}) \oplus \Delta D_{16} \\ &= F(C_{16}) \oplus F(C_{16} \oplus \Delta C_{16}) \\ &= F(C_{16}) \oplus F(C_{16} \oplus \Delta B_{17}) \\ &= F(C_{16}) \oplus F(C_{16} \oplus F(A_{17}) \oplus F(A_{17} \oplus \Delta A_{17}) \oplus \Delta A_{18}) \\ &= F(C_{16}) \oplus F(C_{16} \oplus F(A_{17}) \oplus F(A_{17} \oplus \Delta D_{18}) \oplus \Delta A_{18}),\end{aligned}$$

where

$$\Delta D_{16} = 0,$$
$$\Delta A_{17} = \Delta D_{18},$$
$$\Delta C_{17} = \Delta B_{18}.$$

The attackers can make advantage of the previous faults to deduce A_{15} and C_{15}, respectively. It is helpful to reduce the number of subkey candidates and improve the attacking efficiency. The previous fault injected in A_{15} can deduce the value of A_{15}:

$$F(A_{15}) \oplus F(A_{15} \oplus \Delta A_{15}) \oplus \Delta B_{15} = \Delta A_{16},$$

where

$$\Delta A_{15} = \Delta B_{18},$$
$$\Delta B_{15} = 0,$$
$$\Delta A_{16} = F(C_{17}) \oplus F(C_{17} \oplus \Delta B_{18}) \oplus \Delta C_{18}.$$

Similarly, the previous fault injected in C_{15} can deduce the value of C_{15}:

$$F(C_{15}) \oplus F(C_{15} \oplus \Delta C_{15}) \oplus \Delta D_{15} = \Delta C_{16},$$

where

$$\Delta C_{15} = \Delta D_{18},$$
$$\Delta D_{15} = 0,$$
$$\Delta C_{16} = F(A_{17}) \oplus F(A_{17} \oplus \Delta D_{18}) \oplus \Delta A_{18}.$$

Thus the attackers can derive the correct A_{16} and C_{16} directly by checking whether the following equations are right or not.

$$\begin{aligned}F(A_{16}) &= RK_{32} \oplus A_{17} \oplus B_{16} \\ &= RC_{32} \oplus WK_2 \oplus WK_2 \oplus D_{18} \oplus C_{15} \\ &= RC_{32} \oplus D_{18}, \\ F(C_{16}) &= RK_{33} \oplus C_{17} \oplus D_{16} \\ &= RC_{33} \oplus WK_3 \oplus WK_3 \oplus B_{18} \oplus A_{15} \\ &= RC_{33} \oplus B_{18}.\end{aligned}$$

Hence,

$$\begin{aligned}RK_{34} &= F(A_{17}) \oplus B_{17} \oplus A_{18} \\ &= F(D_{18} \oplus WK_2) \oplus C_{16} \oplus A_{18}, \\ RK_{35} &= F(C_{17}) \oplus D_{17} \oplus C_{18} \\ &= F(B_{18} \oplus WK_3) \oplus A_{16} \oplus C_{18},\end{aligned}$$

and two whitening keys can be recovered on the basis of the key schedule:

$$WK_0 = RK_{35} \oplus RC_{35},$$
$$WK_4 = RK_{34} \oplus RC_{34}.$$

Step 4. The attackers can do brute-force search for the value of the remaining 16-bit whitening key WK_1. The 80-bit secret key can be recovered as follows:

$$K = WK_0 \,||\, WK_1 \,||\, WK_2 \,||\, WK_3 \,||\, WK_4.$$

5. Attacking Complexity

The attacking procedure is summarized to select whitening key candidates for a secret key. The time complexity of brute-force search for one fault injection is

$$\mu = 2^{\sigma},$$

where σ denotes the size of the F-function layer. Furthermore, the calculation of the number of faults is important. In the above attacking steps, both the fault location and the fault model decide the number of faulty ciphertexts to break a whitening key.

The attackers can inject a random fault at any round of the lightweight cryptosystem. If the fault is injected in the last round, only one single nibble in the input of the SubBytes layer will be modified. It can recover no more than one nibble of the last two whitening keys by the analysis. To recover the last two whitening keys, it is essential to inject more faults into other nibbles. If the fault is induced before the last round, there is only one modified nibble in the input difference and output difference of the F-function in this round. However, on the diffusion of linear transformation, there are multinibbles in the output difference of the MixRows layer. Hence, there are multinibbles in the input difference of the SubBytes layer of the last round.

We take the derivation of WK_2 as an example. On the differential relationship of F-function, if A_{17} is a candidate, $A_{17} \oplus \Delta A_{17}$ may be another whitening key candidate. In other words, when the input candidates set of F-functions is not null, the input A_{17} may have several candidates. It indicates that WK_2 may have some possible elements. Usually, more than two faults can have an intersection of WK_2. The attackers continue deriving intersection of whitening key candidates sets until the intersection has only one element. Thus, over two faults are required to derive multinibbles of a whitening key. In the proposed method, when the faults are injected A_{15} and C_{15}, any whitening key can be deduced by only one fault. The theoretical minimum number of faults to recover one whitening key is defined as

$$v = \begin{cases} 0 & if \quad q = 0 \\ \left\lceil \dfrac{\theta \cdot \sigma}{q} \right\rceil & if \quad 1 \le q \le 16 \end{cases},$$

where

$$\theta = \begin{cases} 1 & if \quad \text{a fault is injected in } A_{15} \text{ or } C_{15} \\ 2 & if \quad \text{a fault is injected in } B_{15} \text{ or } D_{15} \end{cases},$$

σ represents the size of the F-function, and q denotes the maximum number of bits in the F-function derived by two faults. To calculate the whitening key, the value of q is the same as the number of bits in the nonzero output difference of the nonlinear transformation in this round. When q=0, there is no bits of a whitening key derived and thus v=0.

The overall time complexity to recover a secret key is

$$\mu \cdot v \cdot g + 2^{r-\eta} = \begin{cases} 2^{r-\eta} & if \quad q = 0 \\ 2^{\sigma} \cdot \left\lceil \frac{\theta \cdot \sigma \cdot g}{q} \right\rceil + 2^{r-\eta} & if \quad 1 \le q \le 16 \end{cases},$$

the data complexity is

$$v \cdot g + 1 = \begin{cases} 1 & if \quad q = 0 \\ \left\lceil \frac{\theta \cdot \sigma \cdot g}{q} \right\rceil + 1 & if \quad 1 \le q \le 16 \end{cases},$$

and the memory complexity is

$$64 \cdot 2 + \eta + 2^{r-\eta},$$

where θ is the coefficient of fault injection, g represents the number of whitening keys to break the secret key, σ denotes the size of the F-function, q is the maximum number of bits in the F-function derived by DFA, r represents the size of the secret key, and η denotes the minimal number of bits in the secret key derived by the differential fault analysis. When $q = 0$, then there is no bits of a whitening key derived and thus $\eta = 0$.

In general, when the faults are injected in A_{15} or C_{15}, the time complexity to recover the 80-bit keys of Khudra is $2^{20.17}$, the data complexity is $2^{1.58}$ and the memory complexity is $2^{20.00}$, where $\theta=1, \sigma = 16, q = 16, g = 2, r = 80,$ and $\eta=60$. When the faults are injected in B_{15} or D_{15}, the time complexity is $2^{18.32}$, the data complexity is $2^{2.32}$ and the memory complexity is $2^{16.01}$, where $\theta=2, \sigma = 16, q = 16, g = 2, r = 80,$ and $\eta=64$.

6. Experimental Results

The attack is implemented in a personal computer with 32GB memory. The fault injections are simulated by the Java program. The attack algorithm runs with 1000 process units. The parameters of accuracy, reliability and latency are applied to estimate the experimental results.

There are 5 groups in average divided in the experiments. They are denoted as G_1, G_2, G_3, G_4 and G_5. **Fig. 2** shows the number of bits recovered in the 80-bit secret key. The x-axis denotes the number of experiments, and the y-axis represents the recovered bits number of the secret key. The colored lines denote the number of the recovered bits of the secret key by injecting faults into A_{15} or C_{15}. We use accuracy, reliability and latency for evaluating the experimental results in detail.

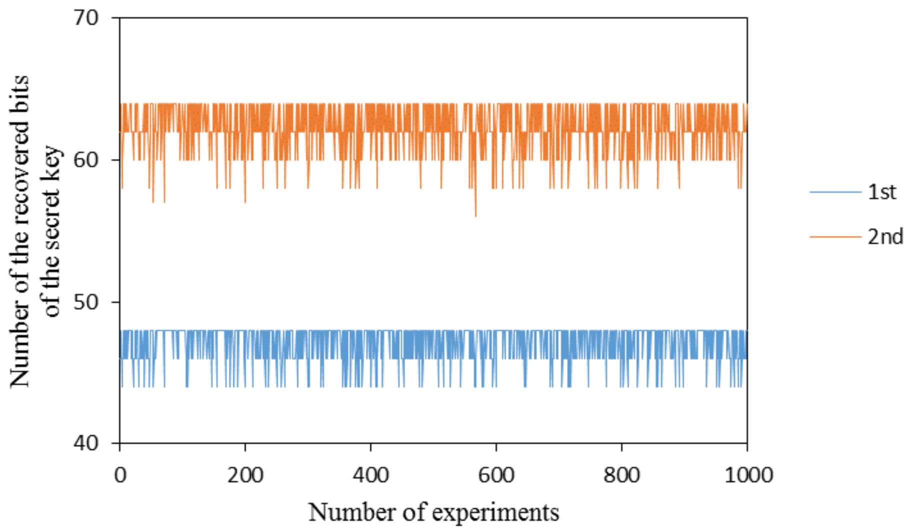


Fig. 2. Number of bits recovered in Khudra

Accuracy defines how close the number of the secret key is to the true number of whitening key candidates. The closer the experimental number of the secret key candidates is to the true number, the more accurate the experiment is. Thus, the Root Mean-Square Error (RMSE) is to measure the accuracy by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(h_0 - h_i)},$$

where N is the number of experiments in a set, i represents the index of the experiment, h_0 denotes the number of bits in the secret key, and h_i represents the number of bits recovered in the secret key candidates. The closer the RMSE value is to 0, the more accurate the experiments are. The RMSE values for every fault injections of whitening key candidates are shown in **Table 7**, where N=200, $i \in \{1,\cdots,1000\}$ and $h_0 = 64$. Eventually, the values of RMSE by two fault injections is nearly zero, so we can derive at least 60 and at most 64 bits of the secret key in the

Table 7. Accuracy measured by RMSE for Khudra

Fault	G_1	G_2	G_3	G_4	G_5
1st	4.09	4.12	4.12	4.10	4.11
2nd	1.24	1.31	1.32	1.34	1.33

corresponding injections. That is, 2 faulty ciphertexts are required to recover secret key. Furthermore, the accuracy in each group for the target interaction is very similar.

Reliability is the ratio of successful experiments out of all experiments made. When only one secret key is derived, the experiment is successful. **Table 8** shows the ratios of successful experiments in each fault injection. The experimental results show that two faults are enough to recover the secret key. That is, the reliability is nearly 100% if the attackers induce only 2 random faults to break a secret key. The reliability in each group for the target interaction is very close.

Table 8. Reliability for Khudra

Fault	G_1	G_2	G_3	G_4	G_5
1st	73.88%	73.45%	73.53%	73.70%	73.64%
2nd	97.60%	97.31%	97.30%	97.20%	97.23%

Latency is the time consumption to the recovery of the whitening key by fault injections. It is measured in seconds. **Fig. 3** shows that the latency of 1000 experiments. The attacking procedure requires 30.88s on average for one experiment.

The attackers only require 2 faults to recover the 80-bit key of Khudra. The overall time complexity is

$$2^{16} \cdot 2 + 2^{20} \approx 2^{20.17},$$

the data complexity is

$$2 + 1 = 3,$$

and the memory complexity is

$$2 \cdot 2^{6} + 60 + 2^{20} \approx 2^{20.00}.$$

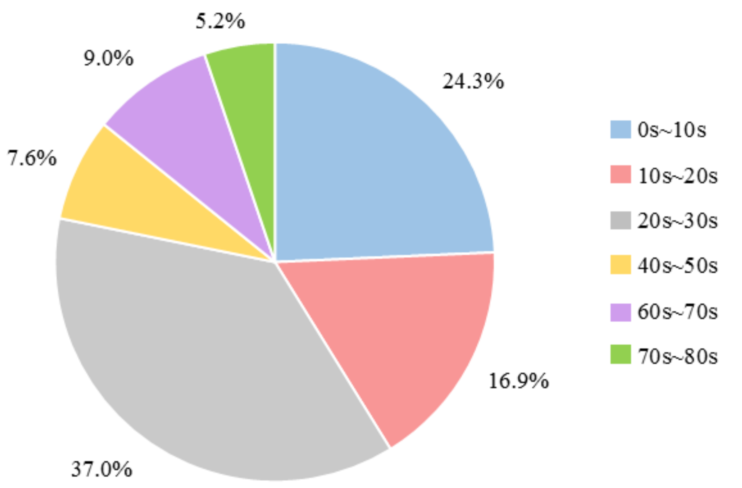


Fig. 3. Latency in DFA attacking

7. Conclusion

This paper presents a differential fault analysis on Khudra in a single nibble-oriented fault model. The analysis can break Khudra by only 2 faults. It shows that Khudra is vulnerable to the differential fault analysis. Hence, more software and hardware protection of the last several rounds should be strengthened in the VANETs.

References

[1] D. H. Kim, S. J. Baek and J. Lim: "Measures for automaker's legal risks from security threats in connected car development lifecycle," *KSII Transactions on Internet & Information Systems*, vol. 11, pp. 865-882, 2017. Article (CrossRef Link).

[2] B. F. Wu, J. H. Juang, and J. Luo: "Real-time vehicle detector with dynamic segmentation and rule-based tracking reasoning for complex traffic conditions," *KSII Transactions on Internet & Information Systems*, vol. 15, pp. 2355-2373, 2011. Article (CrossRef Link).

[3] H. Han, L. Hua, and S. A. Ma: "A Self-authentication and deniable efficient group key agreement protocol for VANET," *KSII Transactions on Internet & Information Systems*, vol. 11, pp. 3678-3698, 2017. Article (CrossRef Link).

[4] J. Nam, K. K. R. Choo, J. Paik and D. Won: "Efficient and anonymous two-factor User authentication in wireless sensor networks: Achieving User Anonymity with Lightweight Sensor Computation," *Plos One*, vol. 10, 2015. Article (CrossRef Link).

[5] D. Engels, M. J. O. Saarinen, P. Schweitzer and E. M. Smith: "The Hummingbird-2 lightweight authenticated encryption algorithm," in *Proc. of International Workshop on Radio Frequency Identification: Security and Privacy Issues*, vol. 7055, pp. 19-31, June, 2011. Article (CrossRef Link).

[6] A. Luykx, B. Preneel, E. Tischhauser, and K. Yasuda: "A MAC mode for lightweight block ciphers," in *Proc. of International Conference on Fast Software Encryption*, vol. 9783, pp. 43-59, July, 2016. Article (CrossRef Link).

[7] Y. Yang, H. Cai, Z. Wei, H. Lu and K. K. R. Choo: "Towards lightweight anonymous entity authentication for IoT applications," in *Proc. of Proceedings of 21st Australasian Conference on Information Security and Privacy*, vol. 9722, pp. 265-280, July, 2016. Article (CrossRef Link).

[8] Y. Yang, J. Lu, K. K. R. Choo and J. Liu: "On lightweight security enforcement in cyber-physical systems," in *Proc. of Proceedings of International Workshop on Lightweight Cryptography for Security & Privacy*, vol. 9542, pp. 97-112, September, 2015. Article (CrossRef Link).

[9] W. Ren, S. Huang, Y. Ren and K. K. R. Choo: "LiPISC: A Lightweight and flexible method for privacy-aware intersection set computation," *Plos One*, vol. 11, 2016. Article (CrossRef Link).

[10] S. K. Ojha, N. Kumar, K. Jain and Sangeeta: "TWIS-A lightweight block cipher," in *Proc. of International Conference on Information Systems Security*, vol. 5905, pp. 280-291, December, 2009. Article (CrossRef Link).

[11] A. bogdanov, L. R. Knudsen, G. Lender, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin and C. Vikkelsoe: "PRESENT: An ultra-lightweight block cipher," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 4727, pp. 450-466, September, 2007. Article (CrossRef Link).

[12] W. Wu and L. Zhang: "LBlock: A Lightweight Block Cipher," in *Proc. of International Conference on Applied Cryptography and Network Security*, vol. 6715, pp. 327-344, June, 2011. Article (CrossRef Link).

[13] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang and I. Verbauwhede: "RECTANGLE: A Bit-slice Lightweight Block Cipher Suitable for Multiple Platforms," *Science China Information Sciences*, vol. 58, pp. 1-15, 2014. Article (CrossRef Link).

[14] L. Li, B. Liu and H. Wang: "QTL: A new ultra-lightweight block cipher. Microprocessors and Microsystems," *Embedded Hardware Design*, vol. 45, pp. 45-55, 2016. Article (CrossRef Link).

[15] X. Dai, Y. Huang, L. Chen, T. Lu and F. Su: "VH: A Lightweight Block Cipher Based on Dual Pseudo-random Transformation," in *Proc. of International Conference on Cloud Computing and Security*, vol. 9483, pp. 3-13, January, 2015. Article (CrossRef Link).

[16] D. Boneh, R. A. DeMillo, R. J. Lipton and M. Yung: "On the importance of checking cryptographic protocols for faults," in *Proc. of International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 37-51, May, 1997. Article (CrossRef Link).

[17] D. Boneh, R. A. DeMillo and R. J. Lipton: "On the importance of eliminating errors in cryptographic computations," *Journal of cryptology*, pp. 101-119, 2001. Article (CrossRef Link).
[18] E. Biham and A. Shamir: "Differential fault analysis of secret key cryptosystems," in *Proc. of International Conference on Advances in Cryptology*, vol. 1294, pp. 513-525, August, 1997. Article (CrossRef Link).
[19] M. Amir, T. M. S. Mohammad and S. Mahmoud: "A generalized method of differential fault attack against AES cryptosystem," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 91-100, October, 2006. Article (CrossRef Link).
[20] P. Dusart, G. Letourneux and O. Vivolo:"Differential fault analysis on AES," in *Proc. of International Conference on Applied Cryptography and Network Security*, pp. 293-306, October, 2003. Article (CrossRef Link).
[21] M. Karpovsky, K. J. Kulikowski and A. Taubin: "Differential fault analysis attack resistant architectures for the Advanced Encryption Standard," in *Proc. of International Conference on Smart Card Research and Advanced Applications VI*, pp. 177-192, August, 2004. Article (CrossRef Link).
[22] G. Piret and J. J. Quisquater: "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 77-88, September 2003. Article (CrossRef Link).
[23] L. Hemme and L. Hoffmann: "Differential fault analysis on the SHA1 compression function," in *Proc. of International Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 54-62, September, 2011. Article (CrossRef Link).
[24] W. Fischer and A. C. Reuter: "Differential fault analysis on Grøstl," in *Proc. of International Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 44-54, September, 2012. Article (CrossRef Link).
[25] R. AlTawy and A. M. Youssef: "Differential fault analysis of Streebog," in *Proc. of International Conference on Information Security Practice and Experience*, pp. 35-49, May, 2015. Article (CrossRef Link).
[26] N. Bagheri, N. Ghaedi and K. S. Sanadhya: "Differential fault analysis of SHA-3," in *Proc. of International Conference in Cryptology*, pp. 253-269, December, 2015. Article (CrossRef Link).
[27] W. Li, W. Zhang, D. Gu, Z. Tao, Z. Zhou, Y. Liu and Z. Liu: "Security analysis of the lightweight cryptosystem TWINE in the Internet of Things," *KSII Transactions on Internet and Information Systems*, vol. 9, pp. 793-810, 2015. Article (CrossRef Link).
[28] S. Kolay, and D. Mukhopadhyay: "Khudra: A new lightweight block cipher for FPGAs," in *Proc. of International Conference on Security, Privacy, and Applied Cryptography Engineering*, vol. 8804, pp. 113-127, October, 2014. Article (CrossRef Link).
[29] M. Tolba, A. Abdekhalek and A. M. Youssef: "Meet-in-the-Middle Attacks on Round-Reduced Khudra," in *Proc. of International Conference on Security, Privacy and Applied Cryptography Engineering*, vol. 9354, pp. 127-138, October, 2015. Article (CrossRef Link).
[30] O. Mehmet, C. Mustafa and K. Ferhat: "A guess-and-determine attack on reduced-round Khudra and weak keys of full cipher," *IACR Cryptology ePrint Archive*, pp.135-146, 2015. Article (CrossRef Link).
[31] X. Ma and K. Qiao: "Related–key rectangle attack on round-reduced Khudra block cipher," in *Proc. of International Conference on Network and System Security*, vol. 9408, pp. 331-344, November, 2015. Article (CrossRef Link).
[32] Q. Yang, L. Hu, S. Sun and L. Song: "Related-key impossible differential analysis of full Khudra," in *Proc. of International Workshop on Security Advances in Information and Computer Security*, vol. 9836, pp. 135-146, September, 2016. Article (CrossRef Link).

Wei Li is currently an associate professor in School of Computer Science and Technology, Donghua University. She was awarded as B.S. degree in engineering from Anhui University in 2002, and her M.S. degree and Ph.D. degree in engineering in 2006 and 2009, both from Shanghai Jiao Tong University. She serves as the member for CACR (China Association of Cryptologic Research), CCF (China Computer Federation) and ACM. Her research interests include the design and analysis of symmetric ciphers.

Chenyu Ge is currently a Master candidate in School of Computer Science and Technology, Donghua University. Her research interests include security analysis of lightweight ciphers.

Dawu Gu is a professor at Shanghai Jiao Tong University in Computer Science and Engineering Department. He was awarded a B.S. degree in applied mathematics in 1992, and a Ph.D. degree in cryptograpgy in 1998, both from Xidian University of China. He serves as technical committee members for CACR (China Association of Cryptologic Research) and CCF (China Computer Federation), also as the members of ACM, IACR, IEICE. He was the winner of New Century Excellent Talent Program made by Ministry of Education of China in 2005. He has been invited as Chairs and TPC members for many international conferences like E-Forensics, ISPEC, ICIS, ACSA, CNCC, etc. His research interests cover cryptology and computer security. He has got over 100 scientific papers in academic journals and conferences.

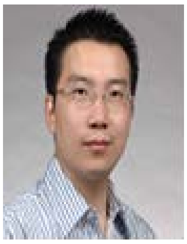
Linfeng Liao is currently a Master candidate in School of Computer Science and Technology, Donghua University. His research interests include security analysis of symmetric ciphers.

Zhiyong Gao is currently a Master candidate in School of Computer Science and Technology, Donghua University. His research interests include fault analysis.

Xiujin Shi is an associate professor in School of Computer Science and Technology, Donghua University. His research interests include Vehicular Ad-hoc Networks.

Ting Lu is an associate professor in School of Computer Science and Technology, Donghua University. Her research interests include security analysis of Vehicular Ad-hoc Networks.

Ya Liu is currently a lecturer in Department of Computer Science and Engineering, University of Shanghai for Science and Technology. She was awarded her Ph.D. degree from Shanghai Jiao Tong University in 2013. Her research interests include the design and analysis of symmetric ciphers and computational number theory.



Zhiqiang Liu is now a Post-doc in the department of Computer Science and Engineering, Shanghai Jiao Tong University. He received his B.S. degree and M.S. degree in Mathematics, and Ph.D. degree in Cryptography from Shanghai Jiao Tong University in 1998, 2001 and 2012 respectively. From 2001 to 2008, he worked in ZTE, Alcatel and VLI in the realm of Next Generation Network (NGN)/IP Multimedia Subsystem (IMS). Currently, his research interests include cryptanalysis and design of block ciphers and hash functions.