Security Analysis of the PHOTON Lightweight Cryptosystem in the Wireless Body Area Network

Wei Li[1, 2, 3], Linfeng Liao[1], Dawu Gu[2], Chenyu Ge[1], Zhiyong Gao[1], Zhihong Zhou[3], Zheng Guo[5], Ya Liu[4, 2], and 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]Shanghai Key Laboratory of Integrate Administration Technologies for Information Security
Shanghai, 200240, China
[4]Department of Computer Science and Engineering, University of Shanghai for Science and Technology
Shanghai, 200093, China
[5] School of Microelectronics, Shanghai Jiao Tong University
Shanghai, 200240, China
[e-mail: zhouzhihong@sjtu.edu.cn]
*Corresponding author: Zhihong Zhou

Received May 22, 2017; revised August 25, 2017; accepted September 21, 2017; published January 31, 2018

Abstract

With the advancement and deployment of wireless communication techniques, wireless body area network (WBAN) has emerged as a promising approach for e-healthcare that collects the data of vital body parameters and movements for sensing and communicating wearable or implantable healthful related information. In order to avoid any possible rancorous attacks and resource abuse, employing lightweight ciphers is most effective to implement encryption, decryption, message authentication and digital signature for security of WBAN. As a typical lightweight cryptosystem with an extended sponge function framework, the PHOTON family is flexible to provide security for the RFID and other highly-constrained devices. In this paper, we propose a differential fault analysis to break three flavors of the PHOTON family successfully. The mathematical analysis and simulating experimental results show that 33, 69 and 86 random faults in average are required to recover each message input for PHOTON-80 /20/16, PHOTON-160/36/36 and PHOTON-224/32/32, respectively. It is the first result of breaking PHOTON with the differential fault analysis. It provides a new reference for the security analysis of the same structure of the lightweight hash functions in the WBAN.

Keywords: Wireless Body Area Network, Lightweight Cryptosystem, Fault Analysis, PHOTON

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, the Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security under Grant No. AGK201703, and the Fundamental Research Funds for the Central Universities.

http://doi.org/10.3837/tiis.2018.01.023 **ISSN : 1976-7277**

1. Introduction

With the rapid development of wearable medical sensors and wireless communication, wireless body area network (WBAN) has emerged as a new application scenario that will revolutionalize the way of seeking healthcare [1]. It has shown great potential in improving healthcare quality, and provides inherently a perfect way to sense ubiquitous health monitoring, computer assisted rehabilitation and emergency medical response systems as **Fig. 1** shows. However, WBAN is the networks with high dynamic topology and their communication is vulnerable to all kind of vicious attacks, and the attackers can exploit WBAN to send deceptive information to beguile others. Furthermore, it takes great challenges coming from stringent resource constraints of in-body and on-body devices, and the high demand for both security/privacy and practicality/usability. Hence, employing cryptosystems, either while stored inside the WBAN or during their transmission outside of the WBAN, is widely recognized as one of the most effective approach for security of WBAN [2-9]. Due to the limitation of processing capability, storage space and power supply of RFID tag and other highly-constrained devices, classical cryptosystems cannot play directly roles in a variety of security applications, such as encryption, decryption, digital signature, and message authentication, etc. It is very critical to implement efficient lightweight cryptosystems in WBAN, i.e., lightweight cryptosystems are mostly desired [10-13]. Appliance of lightweight cryptosystems can reduce energy consumption for devices, and allow more network communications with lower-resource devices.

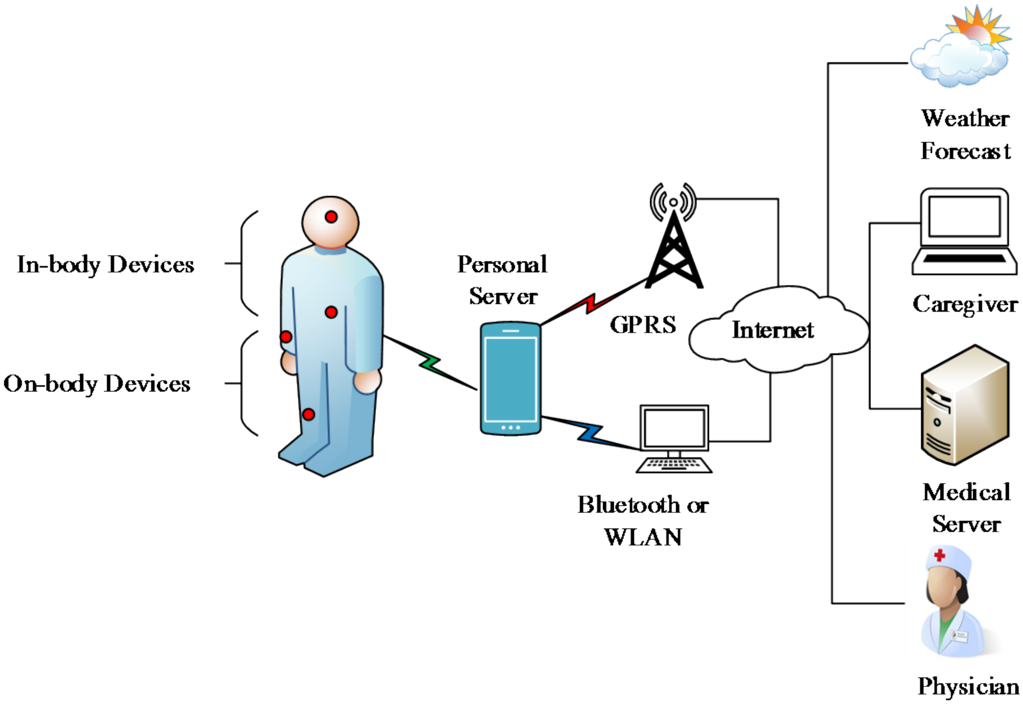

Fig. 1. The WBAN application scenario.

The lightweight hash function is an atomic primitive of authentication and digital signature at the bottom level of a secure WBAN. It can be implemented in the software and hardware modules by the execution file, the static library files, the dynamic link library files, the hard core and the embedded software, etc. As a typical family of lightweight hash functions, PHOTON was proposed by Guo et al. in CRYPTO [14]. It uses an extended sponge function framework to keep the internal memory size as low as possible, and can achieve excellent area and throughput trade-offs. Hence, it is widely applied in the RFID tag deployment and other application characterized by highly-constrained devices in WBAN. Since its introduction, PHOTON is strong against much classical cryptanalysis by the designers, including differential and linear analysis, rebound and super s-box analysis, cube testers and algebraic analysis, slide analysis, rotational analysis, integral analysis and other analysis [14].

Different from classical cryptanalysis, fault analysis puts forward a serious threat for cryptographic implementation against fault analysis in the last two decades [15]. It can exploit easily accessible information like the input-output behavior under malfunctions, which are mounted by anyone using the low-cost equipments. Usually, fault analysis is much more powerful than classical cryptanalysis. In 1996, Boneh et al. presented RSA against fault analysis by provoking the faulty operations [15, 16]. Then differential fault analysis (DFA) was first proposed by Biham et al. on DES in 1997 [17]. It has been applied to cryptanalysis of the public-key ciphers, stream ciphers, block ciphers, and hash functions, such as ECC, RC4, AES and Grøstl etc [18-30] .

To the best of our knowledge, little research has been devoted to the security of PHOTON against the DFA analysis. The previous DFA analysis of hash functions with a sponge function framework depends on the output bitrates to recover the “whole” output of the intermediate state to retrieve the input of the compression function. W. Fischer et al. retrieved the whole 512-bit input of the Grøstl with 296 faults for each block [30]. However, the output bitrates of three flavors of PHOTON, including PHOTON-80/20/16, PHOTON-160/36/36 and PHOTON-224/32/32, are 16 bits, 32 bits and 36 bits, respectively. They are shorter than the output bitrates of the Grøstl. More precisely, these short bitrates are not enough for DFA to recover the “whole” bits of the intermediate state directly. Those DFA techniques on other hash functions are not fully suitable for attacking the PHOTON family. It increases the difficulty of DFA on PHOTON in essence.

In this study, we propose a novel effective differential fault analysis method to break three flavors of the PHOTON family. In the fault model, the attacker can inject faults into the inner layers of PHOTON. Both the fault locations and values are unknown. In the DFA analysis, we add the fault detection to decrease the number of faults and define accuracy, reliability and latency to illustrate the experimental implementation. The method only requires 33, 69 and 86 random faults in average to obtain every block of the message input for PHOTON-80/20/16, PHOTON-160/36/36 and PHOTON-224/32/32, respectively.

The rest of this paper is organized as follows: Section 2 briefly introduces PHOTON. Section 3 presents our method of the DFA analysis on PHOTON. The next two sections summarize the attacking complexity and the experimental results of the DFA analysis. Finally the last section concludes the paper.

2. Specification of PHOTON

2.1 Structure

PHOTON, designed by Guo et al. in Crypto 2011, is a typical family of lightweight hash functions to be suitable for extremely constrained devices [14]. It has the AES-like primitive as an internal unkeyed permutation, and uses a sponge-like construction as the domain extension algorithm. There are five flavors of the PHOTON family covering a wide of spectrum of applications. Each flavor can be denoted as PHOTON-$n/r/r'$, where n denotes its hash output size, r denotes the input size, and r' denotes the output bitrate, respectively. The family includes PHOTON-80/20/16, PHOTON-128/16/16, PHOTON-160/36/36, PHOTON-224/32/32 and PHOTON-256/32/32. The internal state size t depends on the output size and can take five distinct values: 100, 144, 195, 256 and 288 bits. The parameters of the PHOTON family are listed in **Table 1**.

Table 1. The parameters of the PHOTON family.

Flavor	***n***	***r***	***r′***	***l′***	***t***	***d***	***s***
80/20/16	80	20	16	4	100	5	4
128/16/16	128	16	16	7	144	6	4
160/36/36	160	36	36	4	196	7	4
224/32/32	224	32	32	6	256	8	4
256/32/32	256	32	32	7	288	6	8

The PHOTON family takes the message M as an input. The message M is first padded by appending one bit and some zeros so that the total length is l blocks of the bitrate r. Let $M = m_0 \| m_1 \| \cdots \| m_{l-1}$ be an l-block message after padding, where || denotes the concatenation. The t bit initial vector IV is set as

$$IV = \{0\}^{t-24} \| n/4 \| r \| r',$$

where each value is coded on 8 bits.

Each PHOTON flavor is composed of an absorbing stage and a squeezing stage. At iteration e of the absorbing stage, it absorbs the message block m_e on the leftmost part of the internal state S_e and then applies the permutation P as

$$S_0 = IV,$$

$$S_{e+1} = P\ (S_e \oplus (m_e \| \{0\}^c)),$$

where $0 \le e \le l-1$, and $c = t - r$. Once all l message blocks have been absorbed, the hash value can be derived by concatenating the successive r'-bit output blocks $z_{e'}$ until we reach the appropriate output size n in the squeezing stage:

$$Z = z_0 \| z_1 \| \cdots \| z_{l'-1},$$

where l' denotes the number of squeezing iterations, that is $l' = \lceil n/r' \rceil - 1$.And $z_{e'}$ is the r' leftmost bits of the internal state $S_{l+e'}$. There is

$$S_{l+e'+1} = P\ (S_{l+e'}),$$

where $0 \le e' \le l'-1$. If the hash output size is not a multiple of r', one just truncates $z_{l'-1}$ to n mod r' bits.

Furthermore, the internal permutation P is computed as a serial way while maintaining optimal diffusion properties. It is applied on an internal state d^2 elements of s bits each, and can be represented by a $(d \times d)$ matrix. Each P is composed of 12 rounds and each round contains 4 layers in Fig. 2 as follows:

- AddConstants (*AC*) processes the intermediate state with a constant.
- SubCells (*SC*) processes the intermediate state with a nonlinear s-bit substitution table.
- ShiftRows (*SR*) cyclically shifts the last d-1 rows of the intermediate state by different offsets.
- MixColumnsSerial (*MC*) takes all columns of the intermediate state and mixes their data to produce new columns.

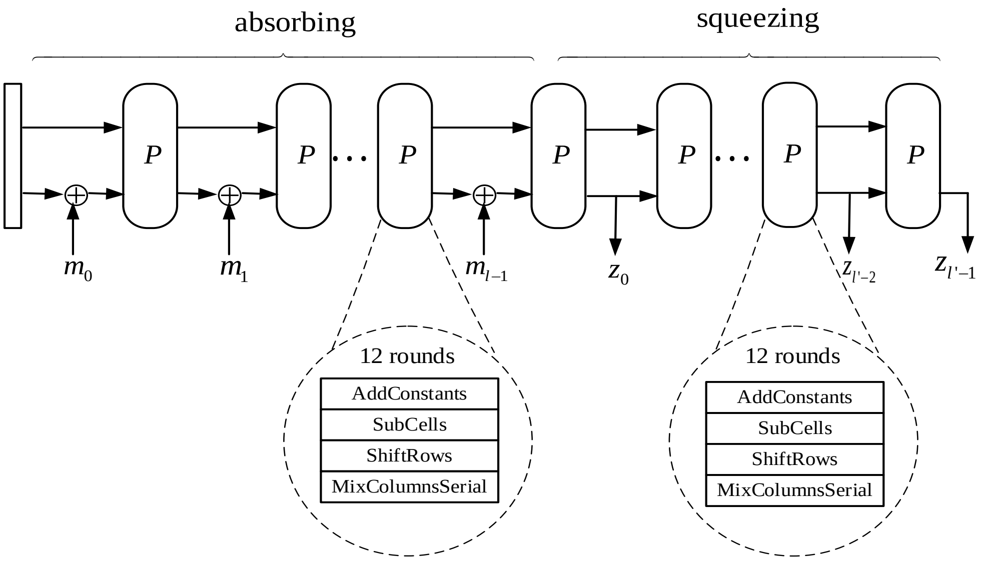

Fig. 2. Message digest generation using PHOTON.

2.2 Notations

The notations of PHOTON and its analysis are described as follows:

Table 2. Notations of PHOTON.

Notations	Description
$M = m_0 \| m_1 \| \cdots \| m_{l-1}$	The message input after padding
$Z = z_0 \| z_1 \| \cdots \| z_{l'-1}$	The hash output
$A_i^x, B_i^x, C_i^x, D_i^x$	The right inputs of AddConstants, SubCells, ShiftRows and MixColumnsSerial in the i-th round of the x-th P with $0 \le x \le l + l' - 1$ and $1 \le i \le 12$
$\overline{A_i^x}, \overline{B_i^x}, \overline{C_i^x}, \overline{D_i^x}$	The faulty inputs of AddConstants, SubCells, ShiftRows and MixColumnsSerial in the i-th round of the x-th P with $0 \le x \le l + l' - 1$ and $1 \le i \le 12$
$\Delta A_i^x, \Delta B_i^x, \Delta C_i^x, \Delta D_i^x$	The input differences of AddConstants, SubCells, ShiftRows and MixColumnsSerial in the i-th round of the x-th P with $0 \le x \le l + l' - 1$ and $1 \le i \le 12$
$AC^{-1}, SC^{-1}, SR^{-1}, MC^{-1}$	The inverse operations of AddConstants, SubCells, ShiftRows and MixColumnsSerial
$a_{i,j}^x, b_{i,j}^x, c_{i,j}^x, d_{i,j}^x$	The j-th cell value of $A_i^x, B_i^x, C_i^x, D_i^x$ with $0 \le x \le l + l' - 1, 1 \le i \le 12$ and $0 \le j \le d^2 - 1$
$\Delta a_{i,j}^x, \Delta b_{i,j}^x, \Delta c_{i,j}^x, \Delta d_{i,j}^x$	The j-th cell value difference of $A_i^x, B_i^x, C_i^x, D_i^x$ with $0 \le x \le l + l' - 1$, $1 \le i \le 12$ and $0 \le j \le d^2 - 1$
F	The matrix used in the MC layer
f_i	The j-th cell value of F

The relationship between the input difference and output difference of the SubCells layer is defined as follows:

$$SS(\Delta a_{i,j}^{x}, \Delta b_{i,j}^{x}) = \{a_{i,j}^{x} \mid a_{i,j}^{x} \in \{0,1\}^{s}, Sbox(a_{i,j}^{x}) \oplus Sbox(a_{i,j}^{x} \oplus \Delta a_{i,j}^{x}) = \Delta b_{i,j}^{x}\},$$

where *Sbox* represents an *S*-box in the SubCells layer with $0 \le x \le l + l' - 1$, $1 \le i \le 12$ and $0 \le j \le d^{2} - 1$.

3. Differential Fault Analysis on PHOTON

3.1 The fault model and basic idea

The DFA analysis exploits the difference between a normal output and a faulty output after processing the same message input. Our proposed fault model includes two assumptions: the attacker has the capability to choose one message to process and obtain the corresponding right and faulty hash outputs. And the attacker can induce random nibble faults to one layer where *s* denotes the size of a cell in an internal state. However, both the location and value of any fault in this layer are unknown.

The main attacking procedure is as follows: a hash output is obtained when a message after padding is processed. When inducing a random error in some round of the hash function, the attacker can obtain a faulty hash output. The analysis exploits the difference between a normal output and a faulty output stemming from operations of the same message. By differential analysis, all intermediate values in the squeezing stage can be recovered. The attacker makes use of the differential analysis to derive the intermediate values in the absorbing stage. Thus, each block of the message input can be obtained by the XOR operation between the intermediate values. The whole message can be derived by the concatenation of all blocks of message input.

3.2 Recovering the intermediate value in the squeezing stage.

This phase aims at recovering $S_{l'+l-1}$ in the squeezing stage. A hash output

$$Z = z_{0} \| z_{1} \| \cdots \| z_{l'-1}$$

is obtained when a message *M* after padding is processed. In the squeezing stage, an *s*-bit random fault is injected before the *MC* layer in the 11th round of the last P. As **Fig. 3** shows, a fault may be induced on either the *AC*, *SC* or *SR* layer whereas the approach is identical in either case. Any modification of one cell provokes the XOR-differences $\Delta A_{11}^{l+l'-1}$ on $A_{11}^{l+l'-1}$, $\Delta B_{11}^{l+l'-1}$ on $B_{11}^{l+l'-1}$, or $\Delta C_{11}^{l+l'-1}$ on $C_{11}^{l+l'-1}$, and $\Delta D_{11}^{l+l'-1}$ on $D_{11}^{l+l'-1}$, $\Delta A_{12}^{l+l'-1}$ on $A_{12}^{l+l'-1}$, $\Delta B_{12}^{l+l'-1}$ on $B_{12}^{l+l'-1}$, $\Delta C_{12}^{l+l'-1}$ on $C_{12}^{l+l'-1}$, and $\Delta D_{12}^{l+l'-1}$ on $D_{12}^{l+l'-1}$. It alters the original hash output bitrate $z_{l'-1}$ into the faulty output bitrate $\overline{z_{l'-1}}$. The attacker has

$$\Delta Z = 0 \| 0 \| 0 \cdots \| 0 \| z_{l'-1} \oplus \overline{z_{l'-1}} = 0 \| 0 \| 0 \cdots \| 0 \| \Delta z_{l'-1},$$

$$\begin{aligned}
\Delta D_{12}^{l+l'-1} &= \xi(\Delta z_{l'-1}),\\
\Delta C_{12}^{l+l'-1} &= MC^{-1}(\Delta D_{12}^{l+l'-1}) = MC^{-1}(\xi(\Delta z_{l'-1})),\\
\Delta B_{12}^{l+l'-1} &= SR^{-1}(\Delta C_{12}^{l+l'-1}) = SR^{-1}(MC^{-1}(\xi(\Delta z_{l'-1}))),
\end{aligned}$$

where $\Delta z_{l'-1}$ denotes the difference of $z_{l'-1}$, and $\xi(\cdot)$ is a function to expand the value of $\Delta z_{l'-1}$ from r' bits to *t* bits. The *MC* layer in the 11th round propagates one *s*-bit input difference to *d* *s*-bit output differences. Then *MC* in the 12th round propagates *d*-multiple *s*-bit input differences to d^{2}-multiple *s*-bit output differences. On the basis of difference characteristics of

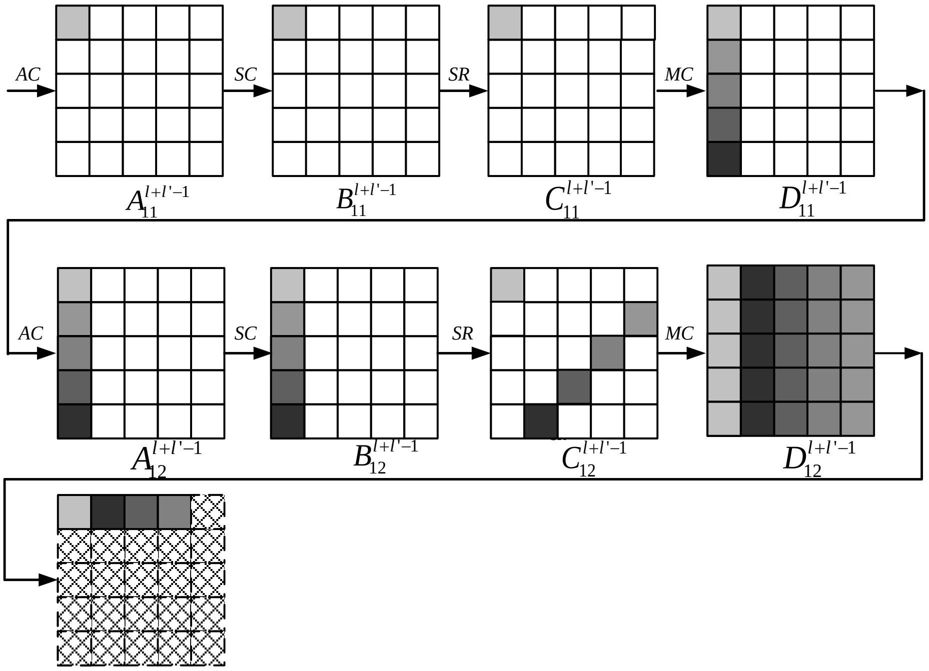

Fig. 3. One 4-bit fault propagation path in squeezing stage of the PHOTON-80/20/16.

every layer, there are the equal or proportional relationships among $\Delta A_{12}^{l+l'-1}$, $\Delta B_{12}^{l+l'-1}$, $\Delta C_{12}^{l+l'-1}$ and $\Delta D_{12}^{l+l'-1}$, as **Fig. 4** shows. Thus, as for $\Delta D_{12}^{l+l'-1}$, all cells' values in different rows of the same column are proportional. On the basis of the relationship between $\Delta z_{l'-1}$ and $\Delta D_{12}^{l+l'-1}$, the attacker can deduce the former $\min(d, r'/s)$ columns in $\Delta D_{12}^{l+l'-1}$ immediately, and $\min(d, r'/s) \cdot d$ elements of $\Delta B_{12}^{l+l'-1}$. It is calculated that there are $d \cdot 2^{(d-\min(d,r'/s)) \cdot d \cdot s}$ possibilities of $\Delta B_{12}^{l+l'-1}$.

The input difference and output difference of the *SC* layer in the 12th round can be represented by $\Delta A_{12}^{l+l'-1}$ and $\Delta B_{12}^{l+l'-1}$. Their relationship is defined as below:

$$SS(\Delta a_{12,j}^{l+l'-1}, \Delta b_{12,j}^{l+l'-1}) = \{a_{12,j}^{l+l'-1} \mid a_{12,j}^{l+l'-1} \in \{0,1\}^s, Sbox(a_{12,j}^{l+l'-1}) \oplus Sbox(a_{12,j}^{l+l'-1} \oplus \Delta a_{12,j}^{l+l'-1}) = \Delta b_{12,j}^{l+l'-1}\},$$

where $0 \le j \le d^2 - 1$.

The above equations, in conjunction with a pair of right and faulty outputs, allow to infer a relation between $\Delta A_{12}^{l+l'-1}$ and $\Delta B_{12}^{l+l'-1}$. It is helpful to restrict a list of possible candidates for $\min(d, r'/s) \cdot d$ elements of $\Delta A_{12}^{l+l'-1}$. The attacker can inject random faults in different columns until he can derive $\min(d, r'/s) \cdot d \cdot s$ bits of $\Delta A_{12}^{l+l'-1}$.

The detail steps are as follows:

Step 1: **Computation**. When a fault is induced before the *MC* layer in the 11th round of the last *P*, the values of the faulty elements have the possibility of $2^s - 1$, and the layer has d^2 elements. Hence $\Delta A_{12}^{l+l'-1}$ has $(2^s - 1) \cdot d^2$ values. The attacker computes all possible differences $\Delta A_{12}^{l+l'-1}$ at the input of the *SC* layer and stores them in a list *Q*.

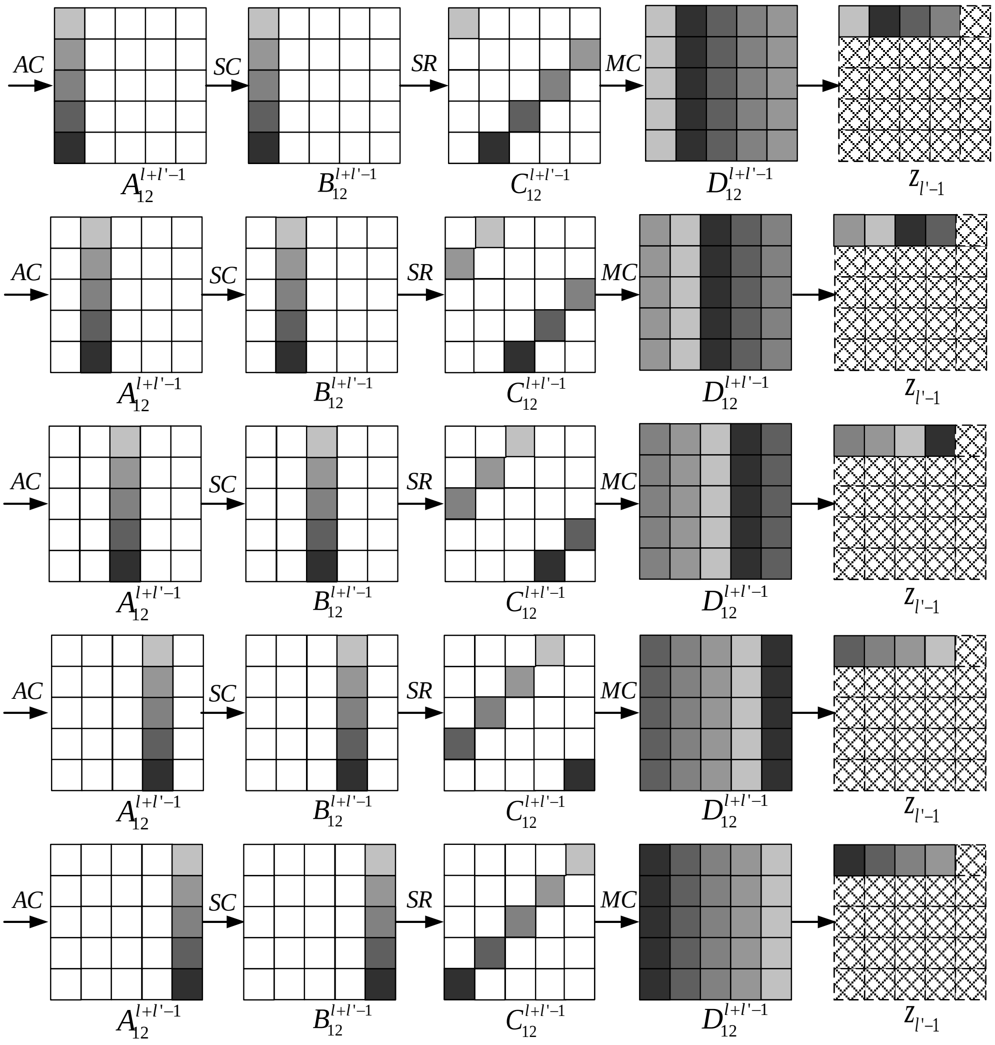

Fig. 4. Relationships of faulty propagation values in PHOTON.

Step 2: Fault injections. The attacker induces a fault before the *MC* layer in the 11th round of the last *P*, and considers one pair of right hash output and corresponding faulty hash output *Z* and $\overline{Z}$. Any fault can lead to *d* *s*-bit differences after the computation of the *MC* layer, and the differences of all elements satisfy the proportional relationships. This important property helps to do brute force search for computing $A_{12}^{l+l'-1}$ in next step.

Step 3: Brute force search. The nonzero elements in one column of $\Delta A_{12}^{l+l'-1}$ are denoted as $(\Delta a_{12,y}^{l+l'-1}, \Delta a_{12,d+y}^{l+l'-1}, \Delta a_{12,2d+y}^{l+l'-1}, \cdots, \Delta a_{12,(d-1)d+y}^{l+l'-1})$, where $0 \le y \le d-1$. For all candidates of $(a_{12,y}^{l+l'-1}, a_{12,d+y}^{l+l'-1}, a_{12,2d+y}^{l+l'-1}, \cdots, a_{12,(d-1)d+y}^{l+l'-1})$, the attackers can compute:

$$(Sbox(a_{12,y}^{l+l'-1}) \oplus Sbox(a_{12,y}^{l+l'-1} \oplus a_{12,y}^{l+l'-1}), (Sbox(a_{12,d+y}^{l+l'-1}) \oplus Sbox(a_{12,d+y}^{l+l'-1} \oplus a_{12,d+y}^{l+l'-1}),$$
$$\cdots, (Sbox(a_{12,d(d-1)+y}^{l+l'-1}) \oplus Sbox(a_{12,d(d-1)+y}^{l+l'-1} \oplus a_{12,d(d-1)+y}^{l+l'-1}))$$
$$= (\Delta b_{12,y}^{l+l'-1}, \Delta b_{12,d+y}^{l+l'-1}, \Delta b_{12,2d+y}^{l+l'-1}, \cdots, \Delta b_{12,d(d-1)+y}^{l+l'-1}),$$

where $(\Delta a_{11,y}^{l+l'-1}, \Delta a_{11,d+y}^{l+l'-1}, \Delta a_{11,2d+y}^{l+l'-1}, \Delta a_{11,3d+y}^{l+l'-1}, \cdots, \Delta a_{11,(d-1)d+y}^{l+l'-1})$ has $(2^s - 1) \cdot d^2$ possible values for the candidates selection. They just equal the nonzero elements in $\Delta A_{12}^{l+l'-1}$ of Q. The attacker does brute force search for $A_{12}^{l+l'-1}$ and derives the candidates set of

$$(a_{12,y}^{l+l'-1}, a_{12,d+y}^{l+l'-1}, a_{12,2d+y}^{l+l'-1}, a_{12,3d+y}^{l+l'-1}, \cdots, a_{12,(d-1)d+y}^{l+l'-1}) .$$

he just repeats the same procedure until $\min(d, r'/s) \cdot d$ elements of $A_{12}^{l+l'-1}$ have only one value.

On the basis of $\min(d, r'/s) \cdot d \cdot s$ bits of $A_{12}^{l+l'-1}$ retrieved, there are $2^{(d-\min(d,r'/s)) \cdot d \cdot s}$ possibilities of $A_1^{l+l'-1}$ as follows:

$$A_1^{l+l'-1} = SC^{-1}(SR^{-1}(MC^{-1}(\cdots(AC^{-1}(SC^{-1}(SR^{-1}(MC^{-1}(AC^{-1}(A_{11}^{l+l'-1}))\cdots)))))).$$

if $d \neq \min(d, r'/s)$, the attacker can depend on the values of $z_{e'}$ to recover the other bits of $A_1^{l+l'-1}$ with $0 \le e' \le l'-2$, by comparing the former r' bits of the candidates of $A_1^{l+l'-1}$ with the value of $z_{l'-2}$. If these two values are equal and the number of candidates is more than 1, the attacker stores them in the candidates set and continues shrinking their scope by having 12 groups of operations of $SC^{-1}(SR^{-1}(MC^{-1}(AC^{-1}(\cdot))))$, and comparing the former r' bits of $A_1^{l+l'-2}$ with the value $z_{l'-3}$. This procedure can be repeated until the candidate set has only one element. So the whole value of $A_1^{l+l'-1}$ is derived. After having the operation of *P*, we can recover the right value of $S_{l+l'-1}$, and other intermediate values of $S_{l+e'}$ in the squeezing stage with $0 \le e' \le l'-2$.

3.3 Recovering the intermediate value in the absorbing stage.

This attacking phase aims at recovering S_{l-1} in the absorbing stage. The whole output of the first *P* in the squeezing stage, denoted as S_l, has been derived in the previous phase. Thus, the whole input of the first *P* in the squeezing stage can be expressed as follows:

$$AC^{-1}(A_1^{l-1}) = AC^{-1}(SC^{-1}(SR^{-1}(MC^{-1}(\cdots(AC^{-1}(SC^{-1}(SR^{-1}(MC^{-1}(S_l))))\cdots)))))).$$

in the process of the absorbing stage, an *s*-bit random fault is induced before the *MC* layer in the last three rounds of the penultimate *P* in the absorbing stage. As **Fig. 5** shows, a fault may be induced on either *AC*, *SC* or *SR* whereas the approach is identical in either case. The attacker induces a random fault in one of the three round of the penultimate *P* in the absorbing stage and obtains a faulty hash output, denoted as $\overline{Z}$. Then, he injects a random fault into this round again and simultaneously a random fault in the 11th round of the last *P* in the squeezing stage, and then obtains another faulty hash output, denoted as $\overline{Z}'$ The relationship between $\overline{Z}$ and $\overline{Z}'$ in this phase is just as same as that between Z and $\overline{Z}$ in the previous phase. On the basis of a pair of $\overline{Z}$ and $\overline{Z}'$, the attacker takes the similar attacking method to derive the faulty input of the first *P* in the squeezing stage, which produces the faulty hash output $\overline{Z}$. Note that the attack may require several different faulty hash outputs. The locations and values of the faults into the last three rounds of the penultimate *P* must be the same in this procedure of continuous inducing faults.

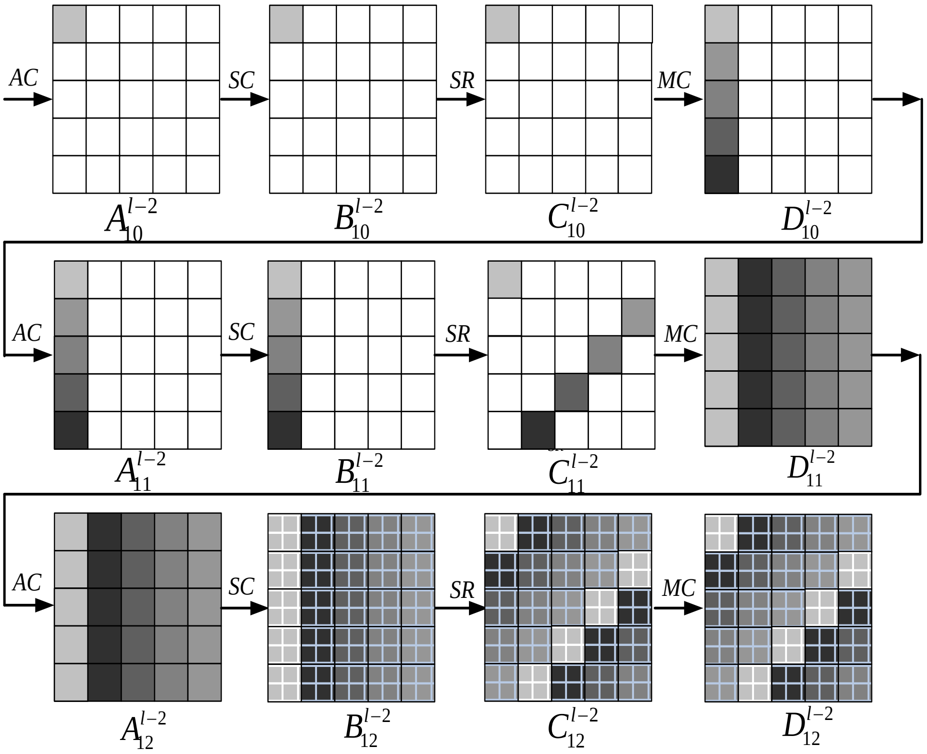

Fig. 5. One 4-bit fault propagation path in absorbing stage of PHOTON.

However, random fault model in the proposed attack can't guarantee the same locations and values of faults. So the attacker takes a fault detection method as follows. As we know, the hash output includes $z_0, z_1, \cdots, z_{l'-1}$ When the faults induced into the last three rounds of the penultimate *P*, he can depend on the values of $z_0, \cdots, z_{l'-2}$ to decide whether the faults are the same or not. In practice, it is enough to detect the fault location and the fault value with the help of $z_0, \cdots, z_{l'-2}$.The pairs of right and faulty inputs of the first *P* in the absorbing stage can help to recover the last message block A_{12}^{l-2} as follows:

$$\Delta D_{12}^{l-2} = \Delta A_1^{l-1} \oplus \Delta m_{l-1} = \Delta A_1^{l-1},$$
$$\Delta C_{12}^{l-2} = MC^{-1}(\Delta D_{12}^{l-2}),$$
$$\Delta B_{12}^{l-2} = SR^{-1}(MC^{-1}(\Delta D_{12}^{l-2})),$$

where B_{12}^{l-2} is the output difference of the SubCells layer in the 12th round of the penultimate P. Each cell satisfies

$$SS(\Delta a_{12,j}^{l-2}, \Delta b_{12,j}^{l-2}) = \{a_{12,j}^{l-2} \mid a_{12,j}^{l-2} \in \{0,1\}^s, Sbox(a_{12,j}^{l-2}) \oplus Sbox(a_{12,j}^{l-2} \oplus \Delta a_{12,j}^{l-2}) = \Delta b_{12,j}^{l-2}\},$$

where $0 \le j \le d^2 - 1$.

Thus，the *j*-th cell of $A_{12,j}^{l-2}$ satisfies $A_{12,j}^{l-2} \in SS(\Delta a_{12,j}^{l-2}, \Delta b_{12,j}^{l-2})$ where $0 \le j \le d^2 - 1$.

Therefore,

$$A_{12}^{l-2} = a_{12,0}^{l-2} \parallel a_{12,1}^{l-2} \parallel \cdots \parallel a_{12,d^2-1}^{l-2}.$$

To decrease the number of faults, an effective approach was presented to select the input difference of the *SC* layer. We take the derivation of A_{12}^{l-2} as an example. One *s*-bit error can lead to *d* *s*-bit differences independently after the computation of the diffusion layer in A_{11}^{l-2}, and the differences can result in the d^2 *s*-bit differences in A_{12}^{l-2} as **Fig. 5** shows. This important property helps to do brute force search on ΔA_{12}^{l-2}. Usually, the 10th round is an appropriate location for the attack. On the basis of the candidates set of ΔA_{12}^{l-2}, the values of A_{12}^{l-2} illustrate proportional relationships among different rows in one column. Hence, the value of ΔA_{12}^{l-2} can be represented by

$$\Delta A_{12}^{l-2} = \Delta D_{11}^{l-2} = F \cdot \Delta C_{11}^{l-2}$$

$$\in \left\{ \begin{aligned} &\begin{pmatrix} f_0 & f_1 & \cdots & f_{d-2} & f_{d-1} \\ f_d & f_{d+1} & \cdots & f_{2d-2} & f_{2d-1} \\ f_{2d} & f_{2d+1} & \cdots & f_{3d-2} & f_{3d-1} \\ & & \cdots & & \\ f_{(d-1)d} & f_{(d-1)d+1} & \cdots & f_{d^2-2} & f_{d^2-1} \end{pmatrix} \cdot \begin{pmatrix} p_0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & p_1 \\ 0 & 0 & \cdots & p_2 & 0 \\ & & \cdots & & \\ 0 & p_{d-1} & \cdots & 0 & 0 \end{pmatrix}, \\ &\begin{pmatrix} f_0 & f_1 & \cdots & f_{d-2} & f_{d-1} \\ f_d & f_{d+1} & \cdots & f_{2d-2} & f_{2d-1} \\ f_{2d} & f_{2d+1} & \cdots & f_{3d-2} & f_{3d-1} \\ & & \cdots & & \\ f_{(d-1)d} & f_{(d-1)d+1} & \cdots & f_{d^2-2} & f_{d^2-1} \end{pmatrix} \cdot \begin{pmatrix} 0 & p_0 & \cdots & 0 & 0 \\ p_1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & p_2 \\ & & \cdots & & \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}, \cdots, \\ &\begin{pmatrix} f_0 & f_1 & \cdots & f_{d-2} & f_{d-1} \\ f_d & f_{d+1} & \cdots & f_{2d-2} & f_{2d-1} \\ f_{2d} & f_{2d+1} & \cdots & f_{3d-2} & f_{3d-1} \\ & & \cdots & & \\ f_{(d-1)d} & f_{(d-1)d+1} & \cdots & f_{d^2-2} & f_{d^2-1} \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & \cdots & 0 & p_0 \\ 0 & 0 & \cdots & p_1 & 0 \\ 0 & 0 & \cdots & p_2 & 0 \\ & & \cdots & & \\ p_{d-1} & 0 & \cdots & 0 & 0 \end{pmatrix} \end{aligned} \right\},$$

where *F* is a matrix in the *MC* layer, and $p_0, p_1, \cdots, p_{d-1}$ are the nonzero *s*-bit values in every column of ΔC_{11}^{l-2}, respectively. The above approach can be repeated by inducing random faults until A_{12}^{l-2} has only one value. Thus,

$$S_{l-1} = MC(SR(SC(A_{12}^{l-2}))).$$

3.4 Recovering the value of the message.

The value of m_{l-1} is the former *r* bits of

$$S_{l-1} \oplus AC^{-1}(A_1^{l-1}).$$

after recovering the last message input m_{l-1}, the attacker makes advantage of the previous phases to derive the right input of the penultimate *P* in the absorbing stage, and induce faults into the 10th round of the antepenultimate *P*. Similar to the previous phases, we can derive the message block m_{l-2}. In this way, we can recover other blocks of message input $m_{l-3}, \cdots, m_1$.The message block m_0 is the former *r* bits of A_0^1.The message *M* can be computed as

$$M = m_0 \| m_1 \| \cdots \| m_{l-1} .$$

4. ATTACKS ON HMAC/NMAC-PHOTON

The keyed-hash message authentication Code, abbreviated as *HMAC*, can work with any cryptographic hash function [31]. The HMAC with *K* and *M* can be calculated as follows:

$$HMAC(K,M) = H_p(K \oplus opad \| H_p(K \oplus ipad \| M)) ,$$

where H_p represent the PHOTON hash function, and || denotes concatenation. The outer padding $opad$ and the inner padding $ipad$ are two one-block long different public constants.

The attacker can compute the input of $H_p(K \oplus opad \| H(K \oplus ipad \| M)$ by DFA. The values of $K \oplus ipad$ and $H(K \oplus ipad \| M)$ can be separated and recovered. Thus, he can recover the value of $K \oplus ipad$, and further derive the value of *K* by xoring the public constant $opad$.

HMAC is always viewed as a single-key variant of NMAC [31]. NMAC is based on a hash function *H* and takes the inner and outer keys, denoted as K_{in} and K_{out} . It is computed by

$$NMAC(K_{out}, K_{in}, M) = H_{K_{out}}(H_{K_{in}}(M)),$$

where the function $H_{K_{in}}(\cdot)$ stands for the hash function with its initial value replaced by K_{in}, and similarly for $H_{K_{out}}(\cdot)$. The internal states $F(IV, K \oplus opad)$ and $F(IV, K \oplus ipad)$ of *HMAC* is equivalent to K_{out} and K_{in} of *HMAC* respectively, where *F* is the compression function and *IV* is the public initial value of *H*. It refers $F(IV, K \oplus opad)$ and $F(IV, K \oplus ipad)$ as the equivalent outer and inner keys, respectively. If these two equivalent keys are recovered, the attacker will be able to forge any message, resulting in a universal forgery attack on *HMAC*.

5. Attacking Complexity

An estimation of the number of faults necessary for the attack to be successful is vital. In the attacking procedure, the number of faults to recover the intermediate value depends on the fault location and the fault model. We take the derivation of the $A_{12}^{l+l'-1}$ in the last *P* of the squeezing stage as an example. On the definition of the SubCells layer, if $A_{12}^{l+l'-1}$ is a candidate, then there may be another candidate $A_{12}^{l+l'-1} \oplus \Delta A_{12}^{l+l'-1}$. In other words, if the input candidates set of the SubCells layer is not null, then the input $A_{12}^{l+l'-1}$ may have several possible elements.

In the fault model, a random error can be induced at any layer of the hash function. If the fault occurs in the last round, only one single s-bit cell in the input of the SubCells layer will change, which can recover at most one s-bit of the intermediate value by the analysis. To recover the intermediate value, it is necessary to induce many faults into different cells.

If the fault is induced at an ideal location before the last round, then the input difference and output difference of the SubCells layer in this round contain only a nonzero *s*-bit cell. However, the output difference of the MixColumnsSerial layer has multicells owing to the diffusion of linear transformation. Thus, the input difference of the SubCells layer in the last round contains multicells after the computation of the last several rounds. The above idea is applied in the attacking procedure to improve the efficiency of fault injection.

Since at least two faults can make one element in the intersection of $A_{12}^{l+l'-1}$, we continue deriving intersection of the candidates sets until the intersection has only one element. Thus, at least two faults are required to derive multicells of one input. The theoretical minimum number of faults to recover one input is defined as

$$\begin{cases} 0 & if \ \ w_s = 0, \\ \lceil 2t / w_s \rceil & if \ \ 1 \le w_s < t, \end{cases}$$

where t represents the size of the internal state, and w_s represents the maximum number of bits in an intermediate state affected by two faults in the squeezing stage. To derive the internal state, the value of w_s equals the number of bits in the nonzero output difference of the nonlinear transformation in this round. If $w_s = 0$, then there is no bits of an input derived and thus the number of faults is zero.

The time complexity of brute force search for one fault injection is

$$2^{s \cdot d} \cdot 2^{s} \cdot \lceil t / s \rceil.$$

Furthermore, when $d \ne \min(d, r'/s)$ the attacker must do brute force search of the other bits in the intermediate value and then compare these bits with the hash values. Thus, the theoretical minimum attacking complexity of brute force search and comparison

$$2^{(d-\min(d,r'/s))\cdot d \cdot s} + l' \cdot r'^{2} \ .$$

Thus, the attacking complexity to recover one intermediate value in the squeezing stage is

$$\begin{cases} 0 & if \ \ w_s = 0 \ , \\ 2^{(d+1)\cdot s+1} \cdot \lceil t^2 / (s \cdot w_s) \rceil + 2^{(d-\min(d,r'/s))\cdot d \cdot s} + l' \cdot r'^{2} & if \ \ 1 \le w_s \le t, d \ne \min(d, r'/s) \ , \\ 2^{(d+1)\cdot s+1} \cdot \lceil t^2 / (s \cdot w_s) \rceil & if \ \ 1 \le w_s \le t, d = \min(d, r'/s) \ , \end{cases}$$

where t denotes the size of the SubCells layer, s represents the input size of one S-box, l' denotes the number of squeezing iterations, r' represents the output bitrate of the hash function, and w_s denotes the maximum number of bits in an intermediate value derived by two faults in the squeezing stage.

When the faults are injected into the absorbing stage, the similar estimation can be applied into their attacking complexity. Thus, the attacking complexity to recover one intermediate value in the absorbing stage is

$$\begin{cases} 0 & if \ \ w = 0 \ , \\ 2^{(d+1)\cdot s+1} \cdot \lceil t^2 / (s \cdot w_a) \rceil & if \ \ 1 \le w_a < t \ , \end{cases}$$

where t denotes the size of the SubCells layer, s denotes the input size of one S-box, and w_a represents the maximum number of bits in an intermediate value derived by two faults in the absorbing stage.

The overall attacking complexity is

$$\begin{cases} 0 & if \ \ w = 0, \\ 2^{(d+1)\cdot s+1} \cdot \lceil t^2 / (s \cdot w_s) + t^2 / (s \cdot w_a) \rceil + 2^{(d-\min(d,r'/s))\cdot d \cdot s} + l' \cdot r'^{2} & if \ 1 \le w_s, w_a \le t, d \ne \min(d, r'/s), \\ 2^{(d+1)\cdot s+1} \cdot \lceil t^2 / (s \cdot w_s) + t^2 / (s \cdot w_a) \rceil & if \ 1 \le w_s, w_a \le t, d = \min(d, r'/s), \end{cases}$$

where t denotes the size of the SubCells layer, s represents the input size of one S-box, l' denotes the number of squeezing iterations, r' represents the output bitrate of the hash function, w_s and w_a denote the maximum number of bits in an intermediate value derived by two faults in the squeezing and absorbing stages, respectively.

6. Experimental Results

We implemented the attack on a PC using the Java language with 32GB memory. The fault induction was simulated by computer software. The definitions of the accuracy, reliability and latency for evaluating the experimental results are described in detail. We take the PHOTON-80/20/26 as an example and consider its absorbing stage and squeezing stage independently. In this situation, we ran the attack algorithm to 1000 process units.

The accuracy is a measure that defines how close the number of the candidates is to the true number of the intermediate value candidates. Basically, the closer the experimental number of the candidates is to the true number, the more accurate the experiment is. Thus, we consider the Root Mean-Square Error (RMSE) to measure the accuracy, where RMSE is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{\varepsilon=1}^{N}[h(\varepsilon)-1]},$$

where N is the number of experiments in a set and ε is the index of the experiment, $h(\varepsilon)$ is the number of candidates, As we know, there is only one true intermediate value. The closer the RMSE value is to 0, the more accurate the experiments are. The experiments include two stages as G_a and G_s, and the experiments in every stage are divided as five groups in average, denoted as $G_{a1}, G_{a2}, \cdots, G_{a5}$ and $G_{s1}, G_{s2}, \cdots, G_{s5}$.The RMSE values for every intersection of the candidates of the PHOTON family, are shown in **Table 3** and **Fig. 6**, where $N = 200$ and $\varepsilon \in \{1, \cdots, 1000\}$. Thus, three interactions are required to recover one intermediate value in the absorbing stage, while one interaction is required to recover any column of the intermediate value in the squeezing stage.

Reliability is the ratio of successful experiments out of all experiments made. If the attacker can derive only one value, the experiment is successful. As for the PHOTON-80/20/16, it is observed in **Table 4** that the ratio of successful experiments for every intersections of candidates in the absorbing stage are 5.9%, 64.6% and 100% respectively, while the ratio in the squeezing stage is 100%. That is, the reliability is 100% if the attacker induces 3.3 random faults in the absorbing stage, and $2 \cdot d$ random faults in the squeezing stage, where d denotes the number of columns in a matrix.

Latency is the time from the first fault injection to the recovery of the intermediate state in our software simulation. It shows that the time of 94.2% experiments is between 0.2 seconds and 0.6 seconds in the absorbing stage, while the time of 88.7% experiments is between 2 hours and 5 hours in the squeezing stage as for the PHOTON-80/20/16 in **Fig. 7**. Note that the latency in the squeezing stage decreases to a few seconds in other two flavors of the PHOTON family. The attacker has to do the brute force search for 2^{20} bits of the PHOTON-80/20/16 with 3 hours in average, since $d \neq \min(d, r'/s)$.

Table 3. The accuracy by RMSE in the absorbing and squeezing stages of the PHOTON family

PHOTON	G	1st	G	1st	2nd	3rd	4th	5th	6th
80/20/16	G_{a1}	0	G_{s1}	18.2	1.5	0	\	\	\
	G_{a2}	0	G_{s2}	17.3	1.3	0	\	\	\
	G_{a3}	0	G_{s3}	205.3	1.3	0	\	\	\
	G_{a4}	0	G_{s4}	32.7	0.1	0	\	\	\
	G_{a5}	0	G_{s5}	25.0	1.6	0	\	\	\
160/36/36	G_{a1}	4.0	G_{s1}	2202.52	8.74	8.74	0.49	0	\
	G_{a2}	4.0	G_{s2}	5355.44	13.11	13.11	0.37	0	\
	G_{a3}	2.6	G_{s3}	2453.75	6.87	6.87	0.70	0	\

	G_{a4}	4.2	G_{s4}	1419.18	289.9	289.4	0.49	0	\
	G_{a5}	4.0	G_{s5}	9606.22	21.13	21.13	0.46	0	\
224/32/32	G_{a1}	4.5	G_{s1}	76803.5	13.93	2.25	0.56	0.26	0
	G_{a2}	4.1	G_{s2}	318041.1	30.77	1.25	0.49	0.19	0
	G_{a3}	5.0	G_{s3}	42984.9	37.55	1.07	0.38	0.26	0
	G_{a4}	3.8	G_{s4}	17517.12	23.22	2.04	0.62	0	0
	G_{a5}	4.9	G_{s5}	845492.8	410.2	1.69	0.56	0.26	0

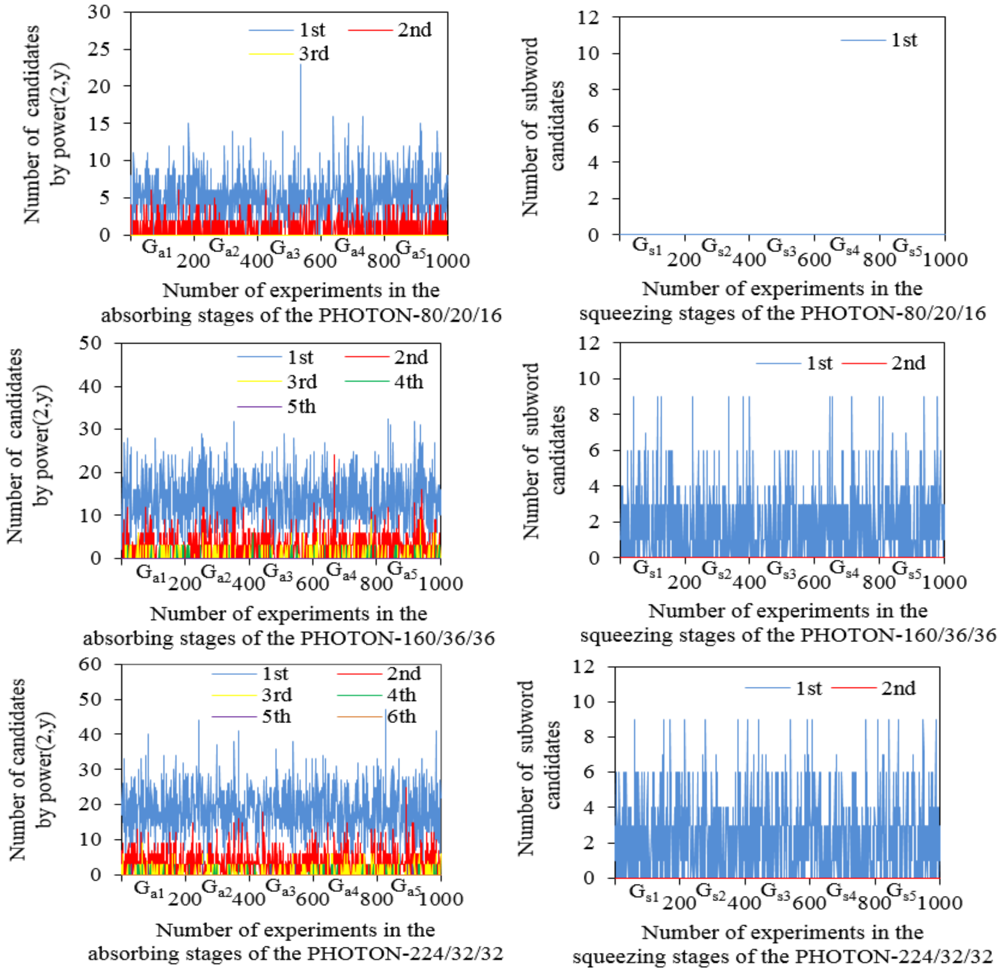

Fig. 6.The intersections of candidates in the absorbing and squeezing stages of the PHOTON family

Table 4.The accuracy by RMSE in the absorbing and squeezing stages of the PHOTON family

PHOTON	G_a	1st	G_s	1st	2nd	3rd	4th	5th	6th
80/20/16	G_{a1}	100%	G_{s1}	4.5%	63.5%	100%	\	\	\
	G_{a2}	100%	G_{s2}	7.0%	63.5%	100%	\	\	\
	G_{a3}	100%	G_{s3}	9.5%	70.0%	100%	\	\	\
	G_{a4}	100%	G_{s4}	4.0%	65.5%	100%	\	\	\
	G_{a5}	100%	G_{s5}	4.5%	60.5%	100%	\	\	\
160/36/36	G_{a1}	38.0%	G_{s1}	1.0%	40.5%	82.0%	96.5%	100%	\
	G_{a2}	41.0%	G_{s2}	0	43.0%	88.5%	98.0%	100%	\

	G_{a3}	41.5%	G_{s3}	0	46.0%	86.5%	97.0%	100%	\
	G_{a4}	35.5%	G_{s4}	1.0%	38.5%	86.0%	96.5%	100%	\
	G_{a5}	29.5%	G_{s5}	0	43.0%	87.0%	97.0%	100%	\
224/32/32	G_{a1}	36.5%	G_{s1}	0.5%	28%	76%	95.5%	99.0%	100%
	G_{a2}	30.0%	G_{s2}	0.5%	27.5%	84%	96.5%	99.5%	100%
	G_{a3}	31.0%	G_{s3}	1.0%	30%	87%	97.5%	99%	100%
	G_{a4}	34.5%	G_{s4}	0	34%	76.5%	94.5%	100%	100%
	G_{a5}	31.0%	G_{s5}	1.0%	27%	79%	95.5%	99.0%	100%

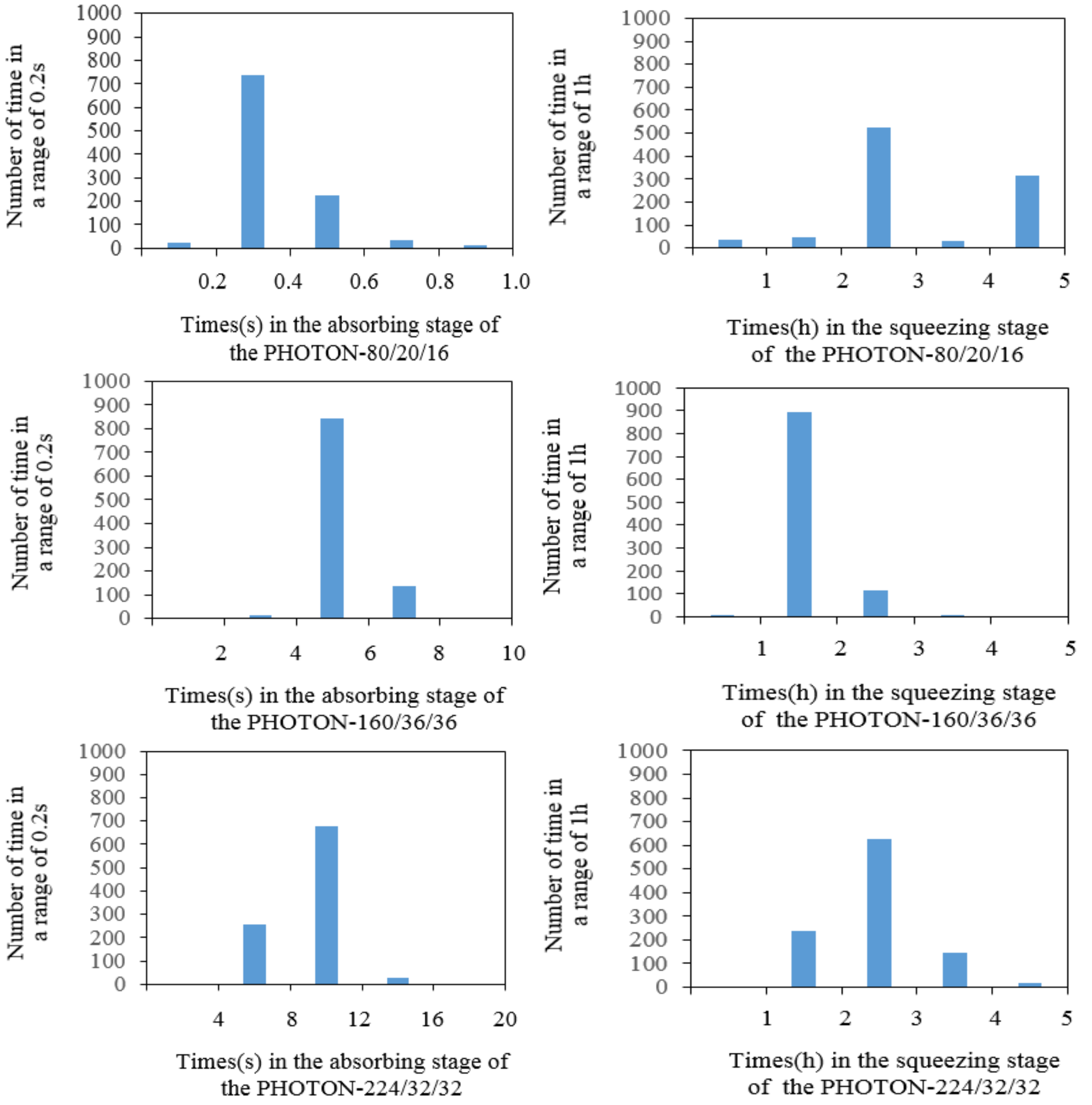

Fig. 7. Latency in the absorbing and squeezing stages of the PHOTON family

As for the PHOTON-80/20/16, the experimental shows that 3.3 and $2\cdot d$ faults are required to recover the intermediate values in the squeezing stage and the absorbing stage respectively. So we need about 33 faulty injections to retrieve the current block of message input, and $33\cdot l$ random faults to retrieve the whole original l blocks of message input after padding.

The attacking complexity is 2^{32} and 2^{33} to recover one block of message input in theory and in experiments, respectively. Thus, the attacking complexity is about $2^{33} \cdot l$ to break the message by the differential fault analysis, where *l* represents the blocks of the message input with $l \geq 1$. **Table 5** shows all experimental results of the PHOTON family, where *FC* and *AC* represent the number of faulty ciphertexts and the attacking complexity, *a* and *s* denote the absorbing stage and the squeezing stage, *T* and *E* are in theory and in experiments, respectively.

Table 5. Fault analysis on three flavors of the PHOTON family.

Flavor	FC_a^T	FC_a^E	FC_s^T	FC_s^E	AC^T	AC^E
80/20/16	2	3.3	10	10	2^{32}	2^{33}
160/36/36	2	3.8	14	18.2	2^{42}	2^{43}
224/32/32	2	4.0	16	21.6	2^{46}	2^{47}

7. Conclusion

This paper presents a differential fault analysis three flavors of the PHOTON family. The analysis can completely break the PHOTON-80/20/16, PHOTON-160/36/36 and PHOTON-224/32/32. We expect that our work will provide deeper understanding of the security of AES-like hash functions. Future research will extend the attack on the other two flavors of the PHOTON family, including PHOTON-128/16/16 and PHOTON-256/32/32.

References

[1] S. S. Javadi and A. M. Razzaque, “Security and Privacy in Wireless Body Area Networks for Health Care Applications.” *Wireless Networks and Security*, vol. 163, pp. 165-187, September, 2013. Article (CrossRef Link).

[2] R. V. Sampangi, S. Dey, R. S. Urs and S. Sampalli, “IAMKeys: Independent and Adaptive Management of Keys for Security in Wireless Body Area Networks.” in *Proc. of 2nd Int. Conf. Computer Science and Information Technology*, vol. 86, pp. 482-494, January, 2012. Article (CrossRef Link).

[3] J. Kang and S. Adibi, “A Review of Security Protocols in mHealth Wireless Body Area Networks (WBAN).” in *Proc. of 1st Int. Conf. Future Network Systems and Security*, vol. 523, pp. 61-83, May, 2015. Article (CrossRef Link).

[4] N. D. Han, L. Han, D. M. Tuan, “A Scheme for Data Confidentiality in Cloud-assisted Wireless Body Area Networks.” *Information Sciences*, vol. 284, pp. 157-166, November, 2014. Article (CrossRef Link).

[5] C. Wang, J. Wu, S. Jiang, “An Asymmetric Signcryption Scheme for Cloud-Assisted Wireless Body Area Network.” in *Proc. of International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, vol. 10067, pp.288-296, November, 2016. Article (CrossRef Link).

[6] K. Zhang, X. Liang, M. Baura, R. Lu and X. Shen, “PHDA: A Priority Based Health Data Aggregation with Privacy Preservation for Cloud Assisted WBANs.” *Information Sciences*, vol. 284, pp. 130-141, November, 2014. Article (CrossRef Link).

[7] X. Liang, X. Li, Q. Shen, R. Lu, X. Lin and X. Shen, “Exploiting Prediction to Enable Secure and Reliable Routing in Wireless Body Area Networks.” *IEEE INFOCOM*, vol. 131, pp. 388-396, March, 2012. Article (CrossRef Link)

[8] E. Biham and A. Shamir, “Differential Fault Analysis of Secret Key Cryptosystems,” in *Proc. of Annual International Cryptology Conference*, vol. 1294, pp. 513-525, August, 1997. Article (CrossRef Link).

[9] K. S. Raja and U. Kiruthika, “An Energy Efficient Method for Secure and Reliable Data Transmission in Wireless Body Area Networks Using RelAODV.” *Wireless Personal Communications*, vol. 83, pp. 2975-2997, August, 2015. Article (CrossRef Link).

[10] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, “HIGHT: A New Block Cipher Suitable for Low-resource Device,” in *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 4249, pp. 46-59, October, 2006. Article (CrossRef Link).

[11] C. H. Lim and T. Korkishko, “mCrypton-A Lightweight block cipher for security of low-cost RFID tags and sensors,” in *Proc. of 6th Int. International Workshop on Information Security Applications*, vol. 3786, pp. 243-258, August, 2005. Article (CrossRef Link).

[12] S. K. Ojha, N. Kumar, K. Jain and Sangeeta, “TWIS-A Lightweight Block Cipher,” in *Proc. of 5th International Conference on Information Systems Security*, vol. 5905, pp. 280-291, December, 2009. Article (CrossRef Link).

[13] 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 9th Int. International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 4727, pp. 450-466, September, 2007. Article (CrossRef Link).

[14] J. Guo, T. Peyrin, and A. Poschmann, “The PHOTON Family of Lightweight Hash Functions,” in *Proc. of 31st Annual Int. Annual Cryptology Conference-CRYPTO*, vol. 6841, pp. 222-239, August, 2011.Article (CrossRef Link).

[15] D. Boneh, R. A. DeMillo and R. J. Lipton, “On the Importance of Checking Cryptographic Protocols for Faults.” in *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 1233, pp. 37–51, 1997. Article (CrossRef Link).

[16] D. Boneh, R. A. DeMillo and R. J. Lipton, “On the Importance of Eliminating Errors in Cryptgraphic Computations,” *Journal of Cryptography*, vol. 14, pp. 101-119, 2001. Article (CrossRef Link).

[17] E. Biham and A. Shamir, “Differential Fault Analysis of Secret Key Cryptosystems,” in *Proc. of 17th Annual Int. Annual International Cryptology Conference*, vol. 1294, pp. 513-525, August, 1997. Article (CrossRef Link).

[18] I. Biehl, B. Meyer and V. M¨uller, “Differential Fault Attacks on Elliptic Curve Cryptosystems.” in *Proc. of 20th Annual Int. Annual International Cryptology Conference*, vol.1880, pp. 131-146, August, 2000. Article (CrossRef Link).

[19] J. J. Hoch and A. Shamir, “Fault Analysis of Stream Ciphers.” in *Proc. of 6th Int. International Workshop of Cryptographic Hardware and Embedded Systems*, vol. 3156, pp. 240-253, August, 2004. Article (CrossRef Link).

[20] S. Banik, S. Maitra and S. Sarkar, “A Differential Fault Attack on the Grain Family of Stream Ciphers.” in *Proc. of 14th Int. International Workshop of Cryptographic Hardware and Embedded Systems*, vol. 7428, pp. 122-139, September, 2012. Article (CrossRef Link).

[21] Y. Yang, J. Lu, K. K. R. Choo and J. Liu, “On Lightweight Security Enforcement in Cyber-physical Systems,” in *Proc. of 4th Int. Conf. International Workshop on Lightweight Cryptography for Security and Privacy*, vol. 9542, pp. 97-112, September, 2015. Article (CrossRef Link).

[22] Y. Yang, H. Cai, Z. Wei, H. Lu and K. K. R. Choo, “Towards Lightweight Anonymous Entity Authentication for IoT Applications.” in *Proc. of 21st Int. Conf. Australasian Conference on Information Security and Privacy*, vol. 9722, pp. 265-280, July, 2016. Article (CrossRef Link).

[23] W. Wu and L. Zhang, “LBlock: A Lightweight Block Cipher,” in *Proc. of 21st Int. Conf. International Conference on Applied Cryptography and Network Security*, vol. 6715, pp. 327-344, June, 2011. Article (CrossRef Link).

[24] 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).

[25] L. Li, B. Liu and H. Wang, "QTL: A New Ultra-lightweight Block Cipher," *Microprocessors and Microsystems*, vol. 45, pp. 45-55, 2016. Article (CrossRef Link).

[26] 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).

[27] P. Dusart, G. Letourneux and O. Vivolo, "Differential fault analysis on AES," in *Proc. of Int. Conf. International Conference on Applied Cryptography and Network Security*, pp. 293-306, October, 2003. Article (CrossRef Link).

[28] G. Piret and J. J. Quisquater, "A Differential Fault Attack Technique Against SPN Structures, with Application to the AES and KHAZAD," in *Proc. of 5th Int. International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 77-88, September 2003. Article (CrossRef Link).

[29] 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).

[30] W. Fischer and C. A. Reuter, "Differential Fault Analysis on Grøstl," in *Proc. of International Workshop Fault Diagnosis and Tolerance in Cryptography*, vol. 29, pp. 44-54, September, 2012. Article (CrossRef Link).

[31] M. Bellare, R. Canetti and H. Krawczyk, "Keying Hash Functions for Message Authentication," in *Proc. of Annual International Cryptology Conference*, vol. 1109, pp. 1-15, August, 1996. 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.

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.

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.

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.

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

Zhihong Zhou is a lecturer in Shanghai Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai Jiao Tong Univeristy. He was awarded her Ph.D. degree from Zhejiang Univeristy in 2008. His research interests include information security.

Zheng Guo is currently a lecturer in School of Microelectronics, Shanghai Jiao Tong University. He was awarded his Ph.D. degree from Shanghai Jiao Tong University in 2015. His research interest includes side channel analysis on symmetric ciphers.

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 currently an associate professor 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.