Power Attack and Protected Implementation on Lightweight Block Cipher SKINNY

Jing Ge
School of Computer Science, Beijing Institute of Technology
State Key Laboratory of Cryptology,
P.O. Box 5159
Beijing, China
gejinghhh@163.com

Yifan Xu
School of Computer Science, Beijing Institute of Technology
Beijing, China
2138750@sina.com

Ruiqian Liu
Network Security Squadron of Public Security Department of Henan Province
Zhengzhou, China
312260991@qq.com

Enze Si
School of Computer Science, Beijing Institute of Technology
Beijing, China
sez95@sina.com

Ning Shang
School of Computer Science, Beijing Institute of Technology
Beijing, China
shangning128@163.com

An Wang
School of Computer Science, Beijing Institute of Technology
Beijing, China
Key laboratory of network assessment technology & Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
wanganl@bit.edu.cn

***Abstract*—SKINNY is a new lightweight tweakable block cipher family, which can compete to other lightweight cipher in terms of hardware or software implementations. While its theoretical security has been widely studied, little effort has been made to analyze its implementation security such as side-channel attacks protection. In this paper, we first give correlation power attack and its experiment on SKINNY. Then, a masking scheme for software SKINNY is proposed against side-channel attacks. Its implementation shows that the protected SKINNY only costs 8.42%, 183.27%, and 90.85% of extra code, time, and RAM, respectively.**

Keywords—SKINNY, lightweight, side-channel attack, correlation power analysis, countermeasure, mask

I. INTRODUCTION

Due to the development of the internet of things, RFID and wireless sensors have become widely used. In order to protect the data transmitted or processed by such resource-constrained and tight-cost devices, lightweight cryptography has emerged and becomes a research hot in cryptography [1]. Compared to traditional cryptography, lightweight cryptography has three main characteristics [2]. First, the data size typically processed by resource-constrained application environments is small, thus the requirements for lightweight cryptography are less strict than ordinary cryptography in terms of throughput [3]. Second, applications such as RFID and wireless sensors often do not require very high security. Third, lightweight cryptography is mostly implemented in hardware. In addition to security, the primary goal of lightweight cryptography algorithms is to realize the occupied space and achieve efficiency for the limitation of environmental conditions [4].

In 2016, Beierle et al. presented a new lightweight family of block ciphers: SKINNY [5], whose goal was to design a cipher that could be implemented highly efficiently on both software and hardware platforms, with performance comparable or better than the SIMON [6] and SPECK families of block ciphers [7]. SKINNY uses a much smaller total number of AND/NOR/XOR gates compared to all known lightweight block ciphers and provides a very good throughput for a reasonably low area cost. It supports a wide range of block sizes and tweak/key sizes. Besides, SKINNY stands out for using a tweakey input [8]to enhance its security and offering a large security margin within the number of rounds for each member of the SKINNY family. It has a very light key schedule and perfectly suits for a scenario where a server communicates with lots of lightweight devices. What's more, it minimizes the decryption overhead by having almost the same description as the encryption counterpart.

Thanks to the above advantages, SKINNY seems better than SIMON in various aspects [9]. However, regarding security, no one has presented a power attack on SKINNY or provided a countermeasure against power attacks on SKINNY so far. In this paper, we show that naive software implementation is vulnerable to the correlation power attack and performed experiments for the correlation power attack. Furthermore, we propose a countermeasure against this attack on SKINNY.

The organization of this paper is as follows. Section II gives specification of SKINNY algorithm and basic information about power analysis attacks. Next in Section III, we introduce the method of correlation power attacks on SKINNY. Section IV shows our experimental results about correlation power attacks. Then, a countermeasure against SKINNY is presented in Section V. Section VI concludes the whole paper.

II. PRELIMINARIES

A. Specification of SKINNY

SKINNY has 64-bit and 128-bit block versions, and we denote n the block size. For a block size n, SKINNY takes a tweakey input with three main tweakey size: $t=n$, $t=2n$ and $t=3n$ (versions with other tweakey sizes between n and $3n$ are naturally obtained from these main versions). In this paper, we take block size 128-bit and tweakey size 128-bit as an example to introduce SKINNY algorithm.

The internal state is viewed as a 4×4 square array of cells, where each cell is a byte. We use *IS* to describe the internal state. When receiving a 128-bit plaintext $m = m_0 \| m_1 \| ... \| m_{14} \| m_{15}$, we set $IS_i = m_i$ for $0 \le i \le 15$:

978-1-5386-7380-5/18/$31.00 ©2018 IEEE
DOI 10.1109/AsiaJCIS.2018.00020

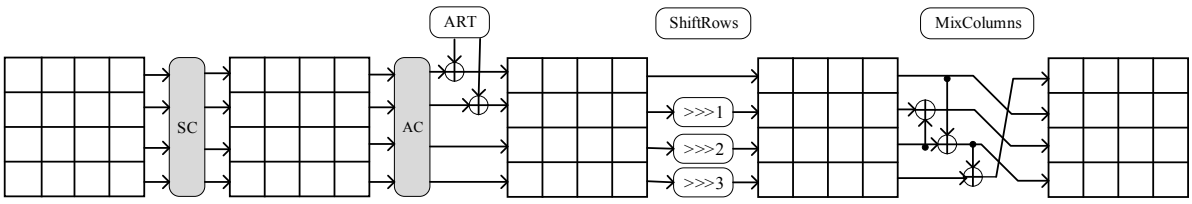

Fig.1. The round function of SKINNY.

$$IS = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}$$

The round function of SKINNY consists of SubCells, AddConstants, AddRoundTweakey, ShiftRows and MixColumns in order (see illustration in Fig. 1), and the number of encryption rounds is 40 with this version.

A S-box is applied to each cell in the internal state. The action of this S-box can be implemented with eight NOR and eight XOR operations. Eight inputs bits of S-box $x_0, x_1, ..., x_7$ (x_0 represents the least significant bit) transform as follows:

$$(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) \rightarrow$$
$$(x_7, x_6, x_5, x_4 \oplus \overline{(x_7 \vee x_6)}, x_3, x_2, x_1, x_0 \oplus \overline{(x_3 \vee x_2)}).$$

Then

$$(x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0) \rightarrow (x_2, x_1, x_7, x_6, x_4, x_0, x_3, x_5) .$$

Regarding AddConstants, the round constants RC_i are XORed to the cipher internal state.

$$RC = \begin{bmatrix} c_0 & 0 & 0 & 0 \\ c_1 & 0 & 0 & 0 \\ c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

with $c_2 = 2$ and

$$(c_0, c_1) = (0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, rc_3 \,||\, rc_2 \,||\, rc_1 \,||\, rc_0,$$
$$0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, 0 \,||\, rc_5 \,||\, rc_4).$$

The six bits are initialized to zero, and updated before used in a given round:

$$(\mathrm{rc}_5 \,||\, rc_4 \,||\, rc_3 \,||\, rc_2 \,||\, rc_1 \,||\, rc_0) \rightarrow$$
$$(rc_4 \,||\, rc_3 \,||\, rc_2 \,||\, rc_1 \,||\, rc_0 \,||\, rc_5 \oplus rc_4 \oplus 1).$$

Then, the tweakey inputs TK_i (TK is also a 4×4 square array of cells, where each cell is a byte) are combined with the state, using bitwise exclusive-or ($0 \le i \le 7$). During each round of encryption, we set $TK_i \leftarrow TK_{P_T[i]}$ with

$$P_T = [9,15,8,13,10,14,12,11,0,1,2,3,4,5,6,7] .$$

For ShiftRows, we set $IS_i \leftarrow IS_{P[i]}$ with

$$P = [0,1,2,3,7,4,5,6,10,11,8,9,13,14,15,12] .$$

MixColumns means pre-multiplying the internal state by matrix M:

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

B. Power Attacks

Side-channel attacks analyze physical leakage that is unintentionally emitted during cryptographic operations in a device [10]. This side-channel leakage is statistically dependent on intermediate processed values involving the secret key, which makes it possible to retrieve the secret from the measured data. While there are various kinds of side channels including power consumption of the target of the device [11], electromagnetic emanation [12], execution time of the cryptographic procedure [13], and so on, the most widely studied one in the last decade is power attack.

Power attacks exploit two types of power consumption dependency: data dependency and operation dependency. It can be roughly summarized into the simple analysis attacks (SPA) [11], the correlation analysis attacks (CPA) [14] and the differential analysis attacks (DPA) [11] .

SPA is a technique that directly explains the power consumption and measures fixed value. Since the amount of power consumed by devices differs depending on the instructions executed by the microprocessor, the attacker can distinguish single instruction by observing the power trace. Instead of intuitive observation, CPA and DPA compare the power consumption of devices according to median values generated by the cryptographic algorithm. In a CPA attack, the attacker obtains two kinds of power traces, the one is obtained by encrypting with the fixed value in internal register, and the other is encrypted with the key guessed. Then the attacker collects key points from two kinds of power traces for correlation analysis to recover the key in register. In a DPA attack, the attacker makes a guess at a bit in the unknown key, partitions the power traces according to some internal register value which depends on this guess, and checks if the partitions show meaningful difference [15].

Since these attack methods were proposed, lots of countermeasures were submitted. Considering that a side channel attack is an attack that takes advantage of information leaked during execution of a cryptographic procedure, thus countermeasures can be categorized into two kinds [16]: independence of computation of procedures from secret information (resistance to SPA) and randomization of the computed objects (resistance to SPA, CPA, and DPA).We can present a countermeasure against SKINNY according to these two principles [17].

III. CORRELATION POWER ATTACKS ON SKINNY

In this section, we outline our strategy for the correlation analysis attacks on SKINNY. First and foremost, a power model

should be chosen to obtain the key used in the encryption process.

In power attacks, it is usually necessary to map the operands to power consumption values, which is a simulation of the power of devices. The Hamming distance model [14] and the Hamming weight model [11] are usually used to complete this mapping in the logic level simulation. In the Hamming weight model, the attacker assumes that the power consumption is proportional to the number of bits set in a data word, while ignoring the values processed before or after the data. In the Hamming distance model, the attacker assumes that the power consumption is related to register contents before and after the target operation done. Based on the fact that the power consumption caused by $0 \to 1$ conversion and $1 \to 0$ conversion have a slight difference, the Hamming weight model can be related at least with the real power consumption to a certain degree. More precisely, the loading and storing of data in memory is usually causing HW leakage, so we use Hamming weight model in our experiment. From now on, $HW(x)$ will represent the Hamming weight of x.

Regarding the place to implement our attack, the output of S-box (denoted as x) in the second round of encryption process is supposed to be chosen. Based on the substitution of the S-box, the internal state would be deeply confused, thus the power consumption after the operation is weakly related to the before. The reason why we do not select the output of S-box in the first round is that the key plays a role after the S-box in each round of encryption. For convenience and simplicity, our attack is implemented in bytes. Now we describe our attack procedures as follows:

(1) We should randomly select n groups of plaintexts and the key fixed in the register to perform SKINNY encryption operations. Each group of plaintexts encrypted can obtain a power trace, denoted as $P_1, P_2 \ldots P_n$.

(2) Then, guess the first byte of the subkey during the first encryption round: k=0;

① k and n groups of plaintexts are used to calculate separately and then obtain n groups of values of x. We can also obtain their corresponding Hamming weight, presented as $HW(x)$.

② For each horizontal axis of the waveform i, we use n groups of $HW(x)$ and n groups of waveform points T_i to compute correlation coefficient, denote as r_j:

$$r_j = \frac{\sum_{i=1}^{n}(HW_i - \overline{HW})(T_i - \overline{T})}{\sqrt{\sum_{i=1}^{n}(HW_i - \overline{HW})^2}\sqrt{\sum_{i=1}^{n}(T_i - \overline{T})^2}}$$

③ Select various values of r_j and calculate their average $\overline{r_k}$.

④ k=k+1, repeat step (2) and not finish until k > 255.

(3) For k=0…k=255, we are supposed to draw Function diagram.

In the end, the key with the biggest correlation coefficient is the real key that we want to retrieve.

IV. Experimental Results

In our experiment, we implement SKINNY algorithm as MCS-51 C codes on STC89C52 processor of MathMagic side-channel analyzer. With sampling rate 1GSa/s, the power consumption can be acquired accurately during the encryption. The acquired power traces are analyzed by Matlab program.

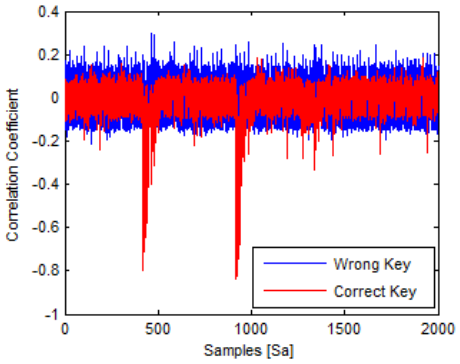

Fig.2. Direct CPA Result

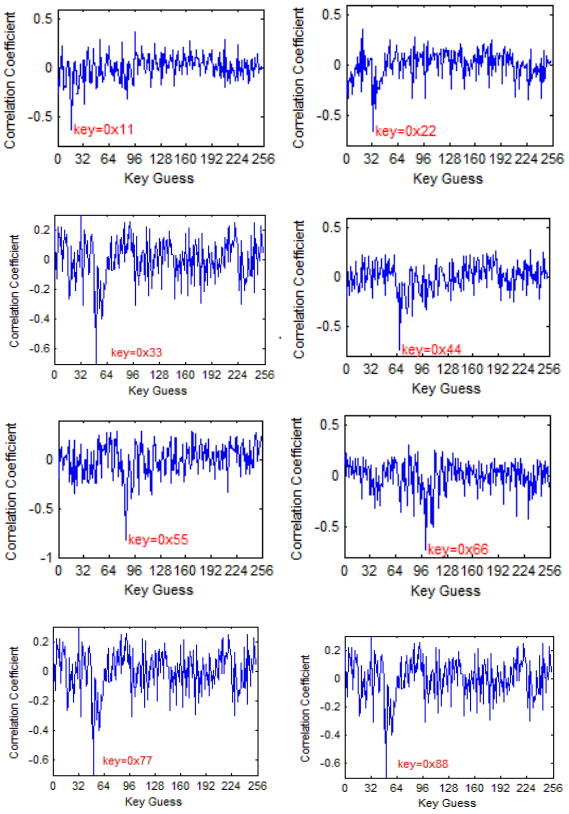

Fig.3. The recovered subkey of first round.

In our experiment, we chose the output of S-box in the second round of encryption to implement our attack. And we selected 100 groups of plaintexts and a fixed key:

key=0x11223344556677889AABBCCDDEEFF00

in the register to conduct our experiment. We assumed the real key during the attack is not known. The real key was just to verify whether our experimental result was true or not. 100 traces and the fixed key was obtained according to these plaintexts. Then we guessed first byte of the subkey during the first encryption round: k=0, …k=255. For each the value of k, k and 100 groups of plaintexts were used to calculate separately and then 100 groups of outputs and their corresponding Hamming weight were also obtained. Afterwards, we selected 2000 key points to calculate correlation coefficient. At last, the key related to the biggest correlation coefficient is the real key while encryption.

Our attack result is presented in Fig. 2. Besides, the first eight bytes of the key were successfully recovered (see Fig. 3). Taking the first byte of the key for example, we compared the efficiency of CPA and DPA (see Fig. 4 and Fig. 5). According to Fig. 4 and Fig. 5, we find that with no more than 20 traces, the key can be retrieved using CPA, while over 80 traces are needed to retrieve the key using DPA. Hence one can see that CPA is more efficient than DPA.

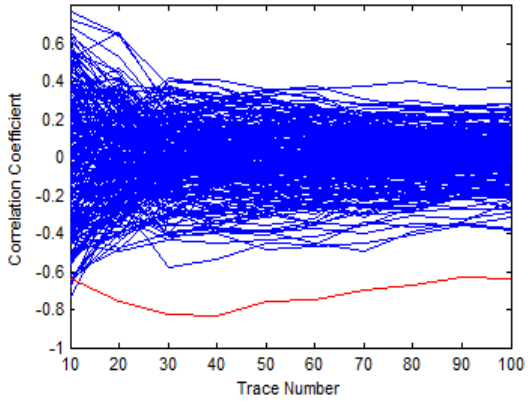

Fig.4. The efficiency of CPA on SKINNY.

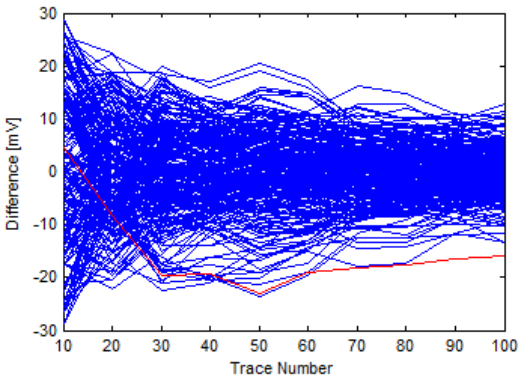

Fig.5. The efficiency of DPA on SKINNY.

V. SKINNY Countermeasure Against Power Attacks

We have implemented the correlation analysis attack on SKINNY. In this section, we should take the countermeasures against SKINNY into account.

Any countermeasure against side-channel attacks is to make the energy consumed by the cryptographic devices don't depend on the median value of the cryptographic algorithm performed by the device. The masking technology, as a well-known countermeasure [18], achieves this goal by randomizing the intermediate values processed by the devices. The advantage of the masking is that it can be implemented at the algorithm level without changing the energy consumption characteristics of the cryptographic devices.

In a masking scheme, every basic intermediate value of a cryptographic algorithm is transformed by a random number called a "mask" [19]. Generally, the mask is generated internally by the cryptographic devices, and the operation is usually defined according to the operation used by the cryptographic algorithm. Moreover, the operations are mostly XOR, modulo, or modular multiplication. The modulus used by the modular plus or modular multiplication is generated based on the cryptographic algorithm. Usually, the mask is directly applied to the plaintext or the key. In order to handle the masked intermediate value and track the mask, the algorithm implementation needs to be modified. For that the result of the encryption is also masked, it is necessary to eliminate the mask at the end of the calculation and restore the real ciphertext.

Referring the masking AES in software for a smart card implementation [20], we brainstorm a mask method to protect SKINNY from correlation analysis attacks. In our masked software implementation of SKINNY, the inputs and outputs of each operation are masked additively. However, we choose a scheme with a fixed mask value. On the one hand, each internal state uses the same mask, on the other hand, each round of encryption uses the same mask [21] (see Fig. 6).

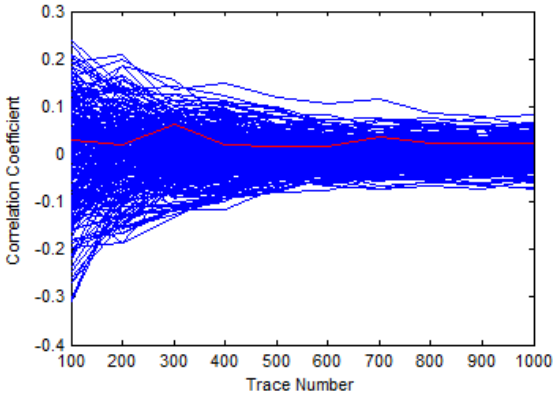

Fig.6. The efficiency of CPA on masked SKINNY.

At first, regarding the plaintext (presented as x), the cryptographic device generates a random number M_p, the input of S-box is $x \oplus M_p$, in order to make the output of S-box is also masked, we have to derive a new masked S-box S'

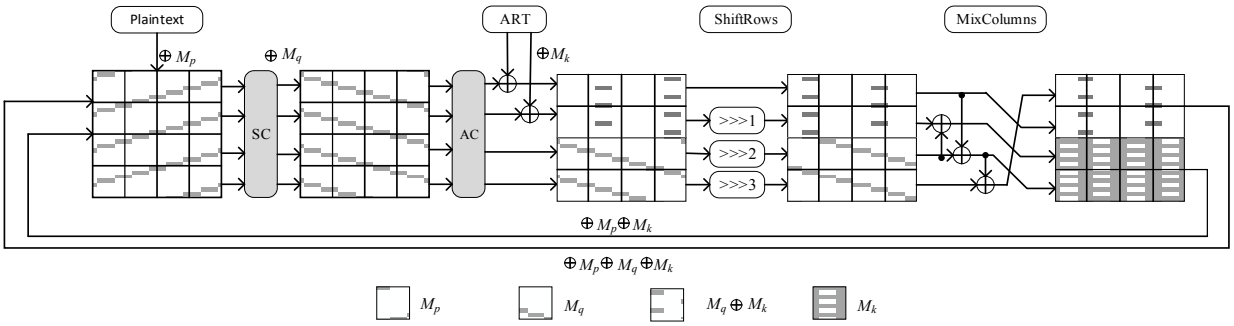

Fig.7. Graphical description of the masked SKINNY.

(the previous S-box is denoted as *S*) with the property $S'(x \oplus M_p) = S(x) \oplus M_q$.

Then, the round constants are combined with the state, respecting array positioning, using bitwise XOR. Therefore, no separate masking effort is required for AddConstants.

After that, the key (denoted as k) should also be masked, so we select a new random number M_k and view $k \oplus M_k$ as the key to complete the following encryption operations. When finishing AddRoundTweakey, the mask of each state is transformed. With regard to the first eight bytes of internal state, the mask is changed into $M_q \oplus M_k$, while the mask of the last eight bytes is still M_q.

The ShiftRows operation is done in combination with the AddRoundTweakey operation by reading and writing the state bytes in a specific order, so extra masking effort is unnecessary.

Finally, after MixColumns, the mask of the first eight bytes of internal state remains its previous values, but the mask of the last eight bytes becomes M_k.

However, before executing next round of encryption, in order to guarantee that each round of encryption uses the same mask, the first eight bytes of the state can add a mask $M_p \oplus M_q \oplus M_k$ to change it into M_p, and the last add a mask $M_p \oplus M_k$. We compare the difference between the naive software implementation of SKINNY and software implementation of masked SKINNY (see TABLE I), and we present the efficiency of CPA on masked SKINNY in Fig. 7.

TABLE I. SOFTWARE IMPLEMENTATION OF SKINNY AND MASKED SKINNY.

	Unmasked SKINNY	Masked SKINNY	Extra cost
Code	2138B	2318B	8.42%
Time	2.3705ms	6.7150ms	183.27%
RAM	284B	542B	90.85%

VI. CONCLUSION

In this paper, we show that a naive software implementation of SKINNY is vulnerable to power attacks, and our experimental results on the correlation power attack on this implementation of SKINNY is presented. In our experiment, it is obviously showed that the efficiency of correlation power attack is higher than differential power attack. Moreover, we proposal a countermeasure against power attacks, both the plaintexts and the key are masked. And using this masking method, SKINNY can effectively be protected from power attacks.

ACKNOWLEDGEMENTS

This work is supported by National Cryptography Development Fund (No. MMJJ20170201), Beijing Natural Science Foundation (No. 4162053), Foundation of Science and Technology on Information Assurance Laboratory, and Beijing Institute of Technology Research Fund Program for Young Scholars. The corresponding author of this work is An Wang.

REFERENCES

[1] Thomas Eisenbarth, Sandeep S. Kumar, Christof Paar, Axel Poschmann, Leif Uhsadel,"A Survey of Lightweight-Cryptography Implementations." IEEE Design & Test of Computers 24(6): 522-533 (2007).

[2] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, Yosuke Todo,"GIFT: A Small Present - Towards Reaching the Limit of Lightweight Encryption. "CHES 2017: 321-345.

[3] Alexey Zhukov,"Lightweight cryptography: modern development paradigms. "SIN 2015: 7.

[4] Gangqiang Yang,"Optimized Hardware Implementations of Lightweight Cryptography. "University of Waterloo, Ontario, Canada 2017.

[5] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, Siang Meng Sim,"The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS." CRYPTO (2) 2016: 123-153.

[6] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, Louis Wingers,"SIMON and SPECK: Block Ciphers for the Internet of Things. "IACR Cryptology ePrint Archive 2015: 585 (2015).

[7] G Yang ,B Zhu,V Suder, MD Aagaard,G Gong,"The Simeck Family of Lightweight Block Ciphers."Springer Berlin Heidelberg , 2015 , 9293 :307-329.

[8] Tetsu Iwata, Kazuhiko Minematsu, Thomas Peyrin, Yannick Seurin,"ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. "CRYPTO (3) 2017: 34-65.

[9] Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, Gaoli Wang,"Related-Key Impossible-Differential Attack on Reduced-Round Skinny. "ACNS 2017: 208-228.

[10] Annelie Heuser, Stjepan Picek, Sylvain Guilley, Nele Mentens,"Side-channel Analysis of Lightweight Ciphers: Does Lightweight Equal Easy? "IACR Cryptology ePrint Archive 2017: 261 (2017).

[11] Paul C. Kocher, Joshua Jaffe, Benjamin Jun,"Differential Power Analysis." CRYPTO 1999: 388-397.

[12] Karine Gandolfi, Christophe Mourtel, Francis Olivier,"Electromagnetic Analysis: Concrete Results. "CHES 2001: 251-261.

[13] Paul C. Kocher,"Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. "CRYPTO 1996: 104-113.

[14] Eric Brier, Christophe Clavier, Francis Olivier,"Correlation Power Analysis with a Leakage Model. "CHES 2004: 16-29.
[15] Mun-Kyu Lee, Jeong Eun Song, Dooho Choi, Dong-Guk Han,"Countermeasures against Power Analysis Attacks for the NTRU Public Key Cryptosystem. "IEICE Transactions 93-A(1): 153-163 (2010).
[16] Katsuyuki Okeya, Kouichi Sakurai,"On Insecurity of the Side Channel Attack Countermeasure Using Addition-Subtraction Chains under Distinguishability between Addition and Doubling. "ACISP 2002: 420-435.
[17] Mohamed Tolba, Ahmed Abdelkhalek, Amr M. Youssef,"Impossible Differential Cryptanalysis of Reduced-Round SKINNY." IACR Cryptology ePrint Archive 2016: 1115 (2016).
[18] François-Xavier Standaert, Gaël Rouvroy, Jean-Jacques Quisquater,"FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks. "FPL 2006: 1-4.
[19] Josep Balasch,Sebastian Faust,Benedikt Gierlichs,and Ingrid Verbaushede,"Theory and Practice of a Leakage Resilient Msking Scheme."Asiacrypto 2012,Beijing,China,6 December 2012.
[20] Elisabeth Oswald, Stefan Mangard, Christoph Herbst, Stefan Tillich,"Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. "CT-RSA 2006: 192-207.
[21] Chang H,Kim K,"Securing AES against second-order DPA by simple fixed-value masking."CSS 2003, 2003(15): 145-150.