Differential Fault Attack on Lightweight Block Cipher PIPO*

SeongHyuck Lim[1][0000−0002−9166−3402], JaeSeung Han[1][0000−0001−7111−2315], Tae-Ho Lee[1][0000−0003−0892−5414], and Dong-Guk Han[1,2][0000−0003−1695−5103]

[1] Department of Financial Information Security, Kookmin University, Seoul, Republic of Korea

[2] Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, Republic of Korea

{seonghyeck16,jae1115,20141932,christa}@kookmin.ac.kr

Abstract. With the recent development of Internet of Things (IoT) devices, related security issues are also increasing. In particular, the possibility of accessing and hijacking cryptographic devices is also increasing due to the rapid increase in usage of these devices. Therefore, research on cryptographic technologies that can provide a safe environment even in resource-constrained environments has been actively conducted. Among them, there are increasing security issues of side-channel analysis for devices due to their physical accessibility. The lightweight block cipher PIPO was recently proposed in ICISC 2020 to address these issues. The PIPO has the characteristic of providing robust security strength while having less overhead when using the side-channel analysis countermeasures. A differential fault attack is a type of side-channel analysis that induces fault in cryptographic operations and utilizes difference information that occurs. Differential fault attacks on the PIPO have not yet been studied. This paper proposed a single-bit flip-based differential fault attack on the lightweight block cipher PIPO for the first time. We show that simulations enable the recovery of the correct secret key with about 98% probability through 64 fault ciphertexts. Therefore, the PIPO does not provide security against differential fault attacks. When using the PIPO cipher on IoT devices, designers must apply appropriate countermeasures against fault-injection attacks.

Keywords: Side-Channel Analysis · Differential Fault Attack · Bit-Sliced lightweight Cipher · PIPO

1 Introduction

Side-channel analysis (SCA) is cryptanalysis that uses physical information, such as power consumption, electromagnetic emission, and sound that occurs while

* This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00520, Development of SCR-Friendly Symmetric Key Cryptosystem and Its Application Modes).

a cryptographic algorithm operates on real devices [9]. Numerous devices have been widely used worldwide with the recent development of Internet of Things (IoT) devices. Under these circumstances, malicious attackers are becoming more accessible to these devices and are naturally becoming able to attack them physically. Therefore, there is an increasing interest in side-channel security in this environment, and a variety of lightweight ciphers and SCA countermeasures are being studied. However, existing block ciphers are inefficient due to large time and memory overhead when countermeasure is applied. When operating lightweight ciphers in an IoT environment, it is not easy to provide side-channel security at the algorithm level. In ICISC 2020, the bit-sliced lightweight block cipher PIPO has been introduced to mitigate overhead [7]. PIPO is designed to respond to SCA effectively. When SCA countermeasures such as masking are applied, it has very little overhead and provides security strength similar to existing ciphers. Various cryptanalysis and optimization papers on the PIPO have been continuously introduced recently, leading to an increasing interest in the PIPO [8, 10].

Differential fault attack (DFA) is a type of semi-invasive SCA that uses difference information that occurs when cryptographic algorithms operate on a device by inducing an artificial fault. DFA was first proposed by Biham et al. for DES in 1997 [1]. Subsequently, DFAs on various cryptographic algorithms such as AES were studied [2, 3, 5], and fault-injection techniques for practical attacks were developed in many ways [11, 12, 14]. DFAs on bit-sliced block ciphers have been studied, but the actual target is mostly lookup table-based implementations and few studies on bit-sliced implementations. In 2018, Sinha et al. proposed a DFA on RECTANGLE-80 implemented with the bit-slice technique [16]. They are based on forcing certain bits to zero or flipping certain consecutive bits or all bits in a word. And they observe the change in the scope of a brute force attack according to each attacker's assumption. The attacker's assumption with coercion and continuity, as discussed in the paper above, is a challenging problem, and the comparatively weak attacker's assumption has a limit in that the range of brute force attacks is enormous.

Our Contributions. Our contributions are twofold. First, we proposed a DFA logic on the recently introduced the lightweight block cipher PIPO for the first time. A random bit-flip at a specific byte position is used in the proposed attack. Experiments have shown that 64 ciphertexts can recover the correct secret key with overwhelming probability. The PIPO does not provide security against DFA. So it suggests the need to apply fault-injection countermeasure techniques in operating the PIPO in the real world. The second contribution is that the proposed attack process can be applied to perform DFAs on various bit-sliced block ciphers. Through this, we expect to contribute to the evaluation of DFA security against unverified bit-sliced block ciphers.

Organization. The remainder of this paper is structured as follows. In Section 2, we introduce the lightweight block cipher PIPO and DFA. Section 3 provides

Table 1. Notation for the PIPO cipher

Parameter	Description
RK_r, c_r	r-round key and constant
S, R, R^{-1}	S-Layer, R-Layer, and Inverse R-Layer
Sb	S-Box operation
$C, C^{\lightning}$	Normal and fault ciphertexts
$m, m^{\lightning}$	Normal and fault S-Layer output
j, k	Matrix index of PIPO input/output ($j, k \in [0, 7]$)
a_{col}^{j}	j-column vector values of PIPO input/output
a_{row}^{k}	k-row vector values of PIPO input/output
x	S-Layer input
i	S-Box input difference
b_p	1-bit of intermediate value ($p \in [0, 63]$)
$\|\|$	Bit concatenation operation

the methodology of the proposed DFA logic on the PIPO. Then, in Section 4, we evaluate the validity of the proposed attack with a simulation-based experiment and discuss the applicability of other bit-sliced block ciphers. Finally, we conclude the paper in Section 5.

2 Backgrounds

2.1 PIPO : Plug-In Plug-Out

The PIPO cipher is described in this section using the notation in Table 1. Han et al. introduced the PIPO a lightweight block cipher in 2020 [7]. It is an SPN-structured cryptographic algorithm that encrypts 64-bit fixed-size plaintexts and is designed for bit-sliced implementation. It is divided into PIPO 64/128 and PIPO 64/256 according to key size and consists of 13 rounds and 17 rounds, respectively. The PIPO has a simple structure in which round operations consisting of S-Layer, R-Layer, and AddRoundKey are repeated after key whitening, as shown in Fig. 1. The operational input and output forms of the PIPO can be expressed in the 8 × 8 matrix, as shown in Fig. 2. When 64-bit plaintexts are defined as $(b_{63}||b_{62}||...||b_1||b_0)$, It is stored in rows by 8-bits as follows:

$$a_{row}^{k} = \left(b_{8\times k+7}||b_{8\times k+6}||...||b_{8\times k+1}||b_{8\times k}\right), k \in [0, 7] \tag{1}$$

The S-Layer operation can be considered that the 8-bit S-Box operation is performed on a column basis in the matrix in Fig. 2 since the PIPO is a bit-sliced structure. When the bit slicing operation is defined as $BitS$, the S-Layer operation can be expressed as follows:

$$S(a) = BitS\left(Sb\left(a_{col}^{0}\right), Sb\left(a_{col}^{1}\right), ..., Sb\left(a_{col}^{6}\right), Sb\left(a_{col}^{7}\right)\right) \tag{2}$$

The R-Layer of the PIPO is a bit permutation that only uses bit-rotations in bytes. The R-Layer operation is performed on row units in Figure 2, At this

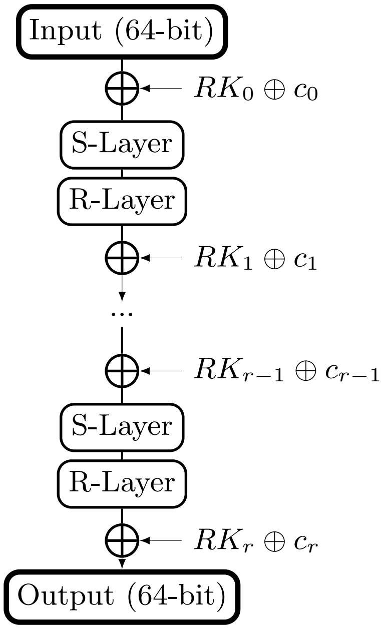


Fig. 1. Overall structure of the PIPO.

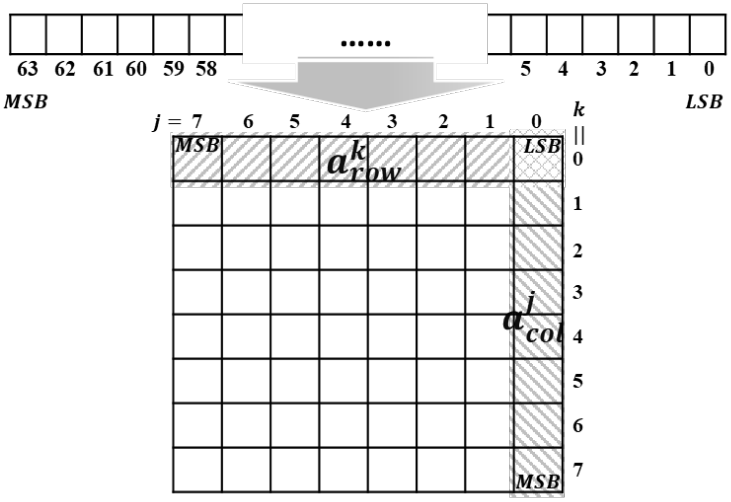


Fig. 2. Operational input and output form of the PIPO 64/128.

point, the rotation operation is performed as follows:

$$\begin{aligned} R\left(a\right) = &(a_{row}^{0} \lll 0, a_{row}^{1} \lll 7, a_{row}^{2} \lll 4, a_{row}^{3} \lll 3 \\ &a_{row}^{4} \lll 6, a_{row}^{5} \lll 5, a_{row}^{6} \lll 1, a_{row}^{7} \lll 2) \end{aligned} \tag{3}$$

Finally, the key schedule of the PIPO has a simple structure in which the secret key is divided by 64 bits and used repeatedly, and round constants are added for each round. The key schedule of PIPO 64/128 is expressed as follows:

$$\begin{gathered} K = K_1 || K_0, \quad RK_i = K_{i\ mod\ 2} \oplus c_i \\ where\ i = 1, 2, ..., 13\ and\ c_i = i \end{gathered} \tag{4}$$

2.2 Differential Fault Attack

A DFA, which is a type of SCA, combines existing differential analysis with fault-injection attack. When the cryptographic device is operating, the attack exploits differece information that results from injecting a fault in the middle. The type of fault that arises determines the difficulty of the practical attack while performing a DFA. According to the fault-injection scale and its positioning capability, the fault model can be classified as follows:

- **Fault Model (fault-injection scale)**
  - `Bit Flip`: The attacker can flip a single-bit of a specific word
  - `Byte Error`: The attacker can change a single byte of a specific word into a random value.
  - `Word Error`: The attacker can change a single word into a random value.
- **Fault Model (positioning capability)**
  - `Chosen Position`: The attacker can specify exactly where the fault is injected.
  - `Random Position`: The attacker cannot specify exactly where the fault is injected. The faults occur in a random position.

With a smaller fault-injection scale, the attacker's assumption is stronger. Furthermore, when a specific position can be defined, the attacker's assumption is strong. Fault-injection attacks can be performed through techniques such as row-hammer attack, laser fault-injection, and electromagnetic fault-injection, etc. Bit flipping is difficult to induce with electromagnetic fault-injection technology, but it is possible to perform on powerful fault models with row-hammer attack and laser fault-injection. Through a low-hammer attack and laser fault-injection, this paper propose a bit-flip fault-based DFA logic on the PIPO at a reproducible level.

3 Differential Fault Attack on PIPO

In this section, we propose a DFA logic on the lightweight block cipher PIPO. The attack process is described based on the PIPO 64/128. The attack can be performed for the PIPO 64/256 by applying the same method for additional rounds.

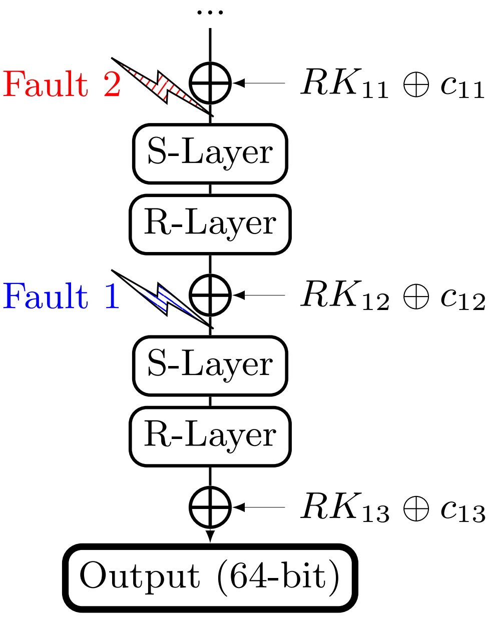


Fig. 3. Fault position of proposed DFA on the PIPO 64/128

3.1 Attacker's Assumption

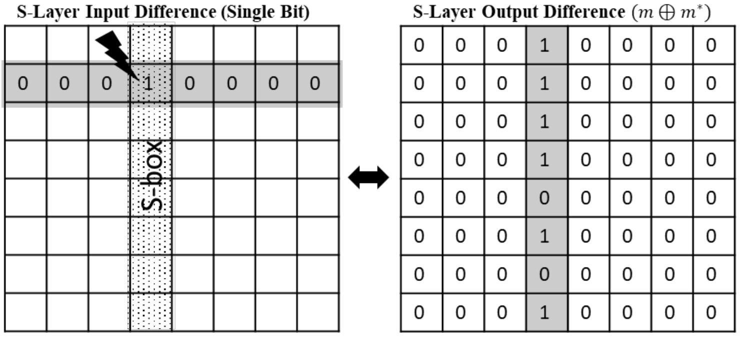
The attacker can conduct encryption in the proposed DFA by obtaining a device with the PIPO operating with a fixed secret key. He or she can induce a fault in which a random single-bit is flipped at a specific byte position during the cryptographic operation and can monitor pairings of normal and fault ciphertexts.

3.2 Proposed DFA Scheme on PIPO

Since the PIPO 64/128 divides the secret key into two parts and uses it repeatedly, recovering two round keys can obtain a secret key completely. Thus, the proposed DFA recovers RK_{13} and RK_{12}, with a total of two attacks on the last round S-Layer input and the penultimate round S-Layer input, as shown Fault 1 and Fault 2 in fig. 3. Each attack consists of four steps as follows:

STEP 1. Calculate the S-Layer I/O difference.

First, during the operation of the PIPO, the attacker is induced to flip a single-bit of a specific byte. This allows to acquire a pair of normal and fault ciphertexts. The normal and fault S-Layer output can be calculated using the acquired normal and fault ciphertext pairs. As a result, the attacker can calculate the S-Layer


S-Layer Input Difference (Single Bit)
S-box
S-Layer Output Difference ($m \oplus m^*$)

0	0	0	1	0	0	0	0

0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0

Fig. 4. Relation between S-Layer input difference and output difference

output difference $\Delta m \left(= m \oplus m^{\lightning}\right)$ as shown in the following equation:

$$\begin{aligned}\Delta m &= R^{-1}\left(C \oplus \left(RK_{13} \oplus c_{13}\right)\right) \oplus R^{-1}\left(C^{\lightning} \oplus \left(RK_{13} \oplus c_{13}\right)\right) \\ &= R^{-1}(C) \oplus R^{-1}\left(RK_{13} \oplus c_{13}\right) \oplus R^{-1}\left(C^{\lightning}\right) \oplus R^{-1}\left(RK_{13} \oplus c_{13}\right) \\ &= R^{-1}(C) \oplus R^{-1}\left(C^{\lightning}\right)\end{aligned} \tag{5}$$

When a certain single-bit of the S-Layer input of the PIPO is flipped, due to bit-sliced structural features, the S-Layer output difference occurs in a column containing that bit, as shown fig. 4. Thus, the attacker can estimate the S-Layer input difference($= i$) depending on whether a specific column of the S-Layer output difference obtained by Equation 5 is zero or not. When the attacker induced a random single-bit flip of t^{th} $(0 \le t \le 7)$ byte and observed the difference in c^{th} $(0 \le c \le 7)$ column of the last round S-Layer output difference, the S-Layer input difference of that byte is 2^c and the S-Box input difference is 2^t.

STEP 2. Determining S-Layer input candidates.

The S-Box input values can be estimated according to pairs of S-Box input differences and output differences determined in **STEP 1**. The S-Box difference equation can be constructed as follows in terms of acquired values:

$$Sb\left(x_{col}^{j}\right) \oplus Sb\left(x_{col}^{j} \oplus i\right) = \left(m \oplus m^{\lightning}\right)_{col}^{j} \tag{6}$$

Through the previous step, the attacker knows the values of i and $\left(m \oplus m^{\lightning}\right)$ exactly. Thus, the S-Box difference table can be used to reduce the number of candidates for S-Box input$\left(= x_{col}^{j}\right)$ satisfying the Equation 6. To determine the only one candidate, multiple difference information is required for the same S-Box. In other words, faults are required for multiple bits in the same column, i.e., the attack on multiple bytes. The attacker can specify the position of the fault-injected byte, and **STEP 1** allows us to find where the bit flip occurs. Thus, he or she can easily be filtering and collecting fault data that fits the conditions. Theoretically, using three difference information to reduce a candidate, it is confirmed as only one candidate with a probability of about 89.1%. And using four difference information, a high probability of about 98.8% determines only one x^j_{col} candidate.

STEP 3. Calculate Round Key.

Repeat **STEP 1** to **2** to confirm S-Layer input in column units. Perform analysis on all columns. Determine all input bytes and calculate the last round key as follows:

$$RK_{13} = R\left(S\left(x\right)\right) \oplus C \oplus c_{13} \tag{7}$$

In the attack process, bit flip faults are required for all column positions in the S-Layer. That is, each of the eight columns must have a fault. And in the previous step, we confirmed that the faults of three or more byte positions uniquely determine one column with high probability. As a result, when the attacker performs an attack after filtering the ciphertext, it is possible to confirm RK_{13} with a probability of about 89.1% through 24 fault ciphertexts and 98.8% through 32 fault ciphertexts.

STEP 4. Confirm secret key.

For RK_{12} recovery, the attacker induces faults in penultimate round S-Layer input, as shown in Fault 2 in fig. 3. The attack process is the same as the recovery of RK_{13}, resulting in only overhead for additional intermediate value calculations and no additional attack logic. Finally, the 128-bit secret key of PIPO 64/128 is completely restored by adding round constants to RK_{12} and RK_{13} obtained through the previous process and then concatenating them as follows:

$$K = (RK_{13} \oplus \texttt{0xd})||(RK_{12} \oplus \texttt{0xc}) \tag{8}$$

4 Experiments and Discussion

4.1 Experimental Result

In this section, evaluation results are shown for the proposed DFA logic on the PIPO 64/128. This experiment is a simulation result and the experimental environment is shown in Table 2. We induced single-bit flip fault randomly for each byte of the last- and penultimate round S-Layer input and filtered the desired form of fault ciphertexts one by one based on **STEP 1** of Section 3.

Table 2. Experimental environment

Item	Configuration
CPU	Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
Memory	32.00G
OS	Window 10 x64
Development platform	Microsoft Visual Studio 2017
Development language	C

As a result of the attack, we were able to analyze the correct round key quickly within a second with 32 fault ciphertexts according to the proposed DFA process. Fig. 5 shows the number of times to determine only one key out of 1000 attacks when the number of attack bytes is different. The number of faults ciphertexts used for each attack is $(8 \times \# \, of \; attack \; bytes)$. The experiments showed similar results to the theoretical predictions. It showed that when performing attacks on three to four bytes, the number of times determined by a single key was higher than 900 times, and a 100% success rate for more bytes.

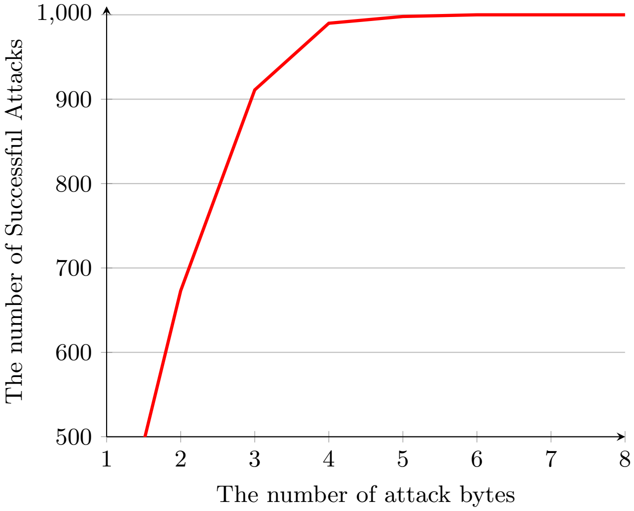


Fig. 5. The number of successful attacks according to the number of attack bytes (1000 times, considered successful when only one key is analyzed)

4.2 DFA on Bit-Sliced Block Ciphers

This section discusses the applicability of the proposed DFA logic to other bit-sliced block ciphers. Bit-sliced block ciphers include ROBIN, FANTOMAS [4], and RECTANGLE [17] etc. Although the S-Box input size and operational word

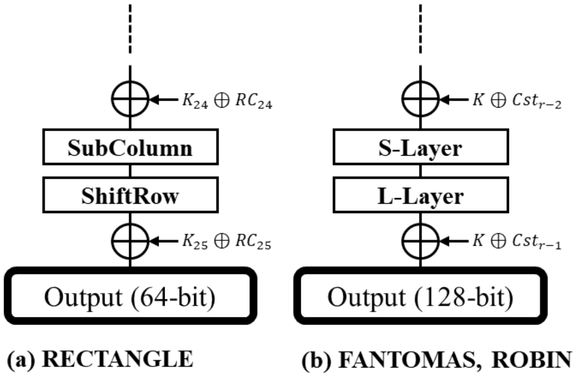


Fig. 6. Last round structure of RECTANGLE, FANTOMAS, and ROBIN

units are different by ciphers. However, when representing intermediate values as matrices as shown in Fig. 2, bit-wise operations are performed per row, while S-Box operations are performed in units of columns. So if a single-bit of a specific byte in the S-Layer input is flipped, a 1-bit difference table for the S-Box can be built. The last round of each cipher is constructed as shown in Figure 6. At this case, since ShiftRow and L-Layer are linear operations, secret key information is not required when calculating the output difference of the substitution layer, and it can be easily calculated through the inverse diffusion layer of the ciphertext difference. Therefore, it is possible to find the exact fault bit position by applying **STEP 1** of Section 3 equally to each cipher. And key candidates can be identified using multiple byte difference information and a difference table. As such, the proposed attack logic entails employing a 1-bit difference table and navigating for fault positions depending on the characteristics of the bit-sliced structure. As a result, it is easy to apply the proposed attack to various bit-sliced block ciphers.

5 Conclusion

In this paper, we proposed a DFA logic on the lightweight block cipher PIPO for the first time and demonstrated the validity of the attack through simulation. With an overwhelming probability, the proposed DFA resulted in the accurate acquisition of the secret key for the PIPO 64/128 with 32 fault ciphertexts for each round, a total of 64 fault ciphertexts. When targeting PIPO 64/256, the proposed attack uses the same logic and performs additional attacks on two rounds. It just requires additional calculations for the intermediate values at this time, and no other logic is demanded, making it easier to apply. As a result, when the PIPO cipher is used in the real world, this paper recommends that the fault-injection countermeasure is used [6, 13, 15]. Because the suggested attack has a structure that is suited for attacking bit-sliced structures, we expect it to be applied in DFAs on various bit-sliced block ciphers. In the future, we plan to employ the proposed attack methodology for a variety of bit-sliced block ciphers. Because the proposed DFA logic is based on a single-bit flip model and must be able to specify a specific byte position, the attacker's assumption is rather strong. Thus, we plan to design an attack logic that alleviates the assumption of attackers, and verify the attack's validity to the real device through an actual electromagnetic fault-injection attack.

References

1. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Jr., B.S.K. (ed.) Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1294, pp. 513–525. Springer (1997). https://doi.org/10.1007/BFb0052259, https://doi.org/10.1007/BFb0052259
2. Dusart, P., Letourneux, G., Vivolo, O.: Differential fault analysis on aes. In: International Conference on Applied Cryptography and Network Security. pp. 293–306. Springer (2003)
3. Floissac, N., L'Hyver, Y.: From AES-128 to AES-192 and aes-256, how to adapt differential fault analysis attacks on key expansion. In: Breveglieri, L., Guilley, S., Koren, I., Naccache, D., Takahashi, J. (eds.) 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2011, Tokyo, Japan, September 29, 2011. pp. 43–53. IEEE Computer Society (2011). https://doi.org/10.1109/FDTC.2011.15, https://doi.org/10.1109/FDTC.2011.15
4. Grosso, V., Leurent, G., Standaert, F., Varici, K.: Ls-designs: Bitslice encryption for efficient masked software implementations. In: Cid, C., Rechberger, C. (eds.) Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8540, pp. 18–37. Springer (2014). https://doi.org/10.1007/978-3-662-46706-0_2, https://doi.org/10.1007/978-3-662-46706-0_2
5. Han, L., Wu, N., Ge, F., Zhou, F., Wen, J., Qing, P.: Differential fault attack for the iterative operation of AES-192 key expansion. In: 20th IEEE International Conference on Communication Technology, ICCT 2020, Nanning, China, October 28-31, 2020. pp. 1156–1160. IEEE (2020). https://doi.org/10.1109/ICCT50939.2020.9295779, https://doi.org/10.1109/ICCT50939.2020.9295779
6. He, W., Breier, J., Bhasin, S., Miura, N., Nagata, M.: Ring oscillator under laser: Potential of pll-based countermeasure against laser fault injection. In: 2016 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2016, Santa Barbara, CA, USA, August 16, 2016. pp. 102–113. IEEE Computer Society (2016). https://doi.org/10.1109/FDTC.2016.13, https://doi.org/10.1109/FDTC.2016.13
7. Kim, H., Jeon, Y., Kim, G., Kim, J., Sim, B., Han, D., Seo, H., Kim, S., Hong, S., Sung, J., Hong, D.: PIPO: A lightweight block cipher with efficient higher-order masking software implementations. In: Hong, D. (ed.) Information Security and Cryptology - ICISC 2020 - 23rd International Conference, Seoul, South Korea, December 2-4, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12593, pp. 99–122. Springer (2020). https://doi.org/10.1007/978-3-030-68890-5_6, https://doi.org/10.1007/978-3-030-68890-5_6

8. Kim, H., Sim, M., Eum, S., Jang, K., Song, G., Kim, H., Kwon, H., Lee, W., Seo, H.: Masked implementation of PIPO block cipher on 8-bit AVR microcontrollers. In: Kim, H. (ed.) Information Security Applications - 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11-13, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13009, pp. 171–182. Springer (2021). https://doi.org/10.1007/978-3-030-89432-0_14, https://doi.org/10.1007/978-3-030-89432-0_14
9. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996). https://doi.org/10.1007/3-540-68697-5_9, https://doi.org/10.1007/3-540-68697-5_9
10. Kwak, Y., Kim, Y., Seo, S.C.: Parallel implementation of PIPO block cipher on 32-bit RISC-V processor. In: Kim, H. (ed.) Information Security Applications - 22nd International Conference, WISA 2021, Jeju Island, South Korea, August 11-13, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13009, pp. 183–193. Springer (2021). https://doi.org/10.1007/978-3-030-89432-0_15, https://doi.org/10.1007/978-3-030-89432-0_15
11. Lim, H., Lee, J., Han, D.G.: Novel fault injection attack without artificial trigger. Applied Sciences **10**(11), 3849 (2020)
12. Roscian, C., Dutertre, J., Tria, A.: Frontside laser fault injection on cryptosystems - application to the aes' last round -. In: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2013, Austin, TX, USA, June 2-3, 2013. pp. 119–124. IEEE Computer Society (2013). https://doi.org/10.1109/HST.2013.6581576, https://doi.org/10.1109/HST.2013.6581576
13. Schneider, T., Moradi, A., Güneysu, T.: Parti - towards combined hardware countermeasures against side-channel and fault-injection attacks. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 302–332. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_11, https://doi.org/10.1007/978-3-662-53008-5_11
14. Seaborn, M., Dullien, T.: Exploiting the dram rowhammer bug to gain kernel privileges. Black Hat **15**, 71 (2015)
15. Seo, H., Park, T., Ji, J., Kim, H.: Lightweight fault attack resistance in software using intra-instruction redundancy, revisited. In: Kang, B.B., Kim, T. (eds.) Information Security Applications - 18th International Conference, WISA 2017, Jeju Island, Korea, August 24-26, 2017, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10763, pp. 3–15. Springer (2017). https://doi.org/10.1007/978-3-319-93563-8_1, https://doi.org/10.1007/978-3-319-93563-8_1
16. Sinha, S., Karmakar, S.: Differential fault analysis of rectangle-80. IACR Cryptol. ePrint Arch. p. 428 (2018), https://eprint.iacr.org/2018/428
17. Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., Verbauwhede, I.: RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Sci. China Inf. Sci. **58**(12), 1–15 (2015). https://doi.org/10.1007/s11432-015-5459-7, https://doi.org/10.1007/s11432-015-5459-7