IEICE Electronics Express, Vol.16, No.11, 1–5

LETTER

A new method for resisting collision attack based on parallel random delay S-box

Fang Zhou[1a)], **Ning Wu**[1b)], **Xiaoqiang Zhang**[2], **and Jinbao Zhang**[1]

Abstract Collision Attack (CA) has posed a huge threat to the security of AES circuit. To protect sensitive information, it's necessary to do research on defense strategy of CA. This letter proposes a new method to defense CA through the implementation of random delay based parallel S-box. It can destroy the consistency of the power consumption curves, confuse the judgment of the collision and the setting of the collision threshold to achieve the goal of resisting the CA. Compared to the well-known random mask method and other CA countermeasures, our strategy can defense CA without changing the AES round transformation architecture and bring extra resource overhead.

Keywords: CA, defense strategy, S-box, random delay insertion
Classification: Integrated circuits

1. Introduction

As an international block cipher algorithm, AES is widely used in the field of information security for its high key sensitivity, short build time and low memory requirements. However, the Collision Attack (CA) technology developed in recent years can quickly recover the key of AES, and has brought great challenges to the security of AES circuits [1, 2, 3, 4, 5, 6, 7, 8, 9].

In order to avoid the leakage of sensitive information, it is essential to carry out research on the CA defense strategy of AES to design a safe and reliable AES circuit. The defense strategy of CA can be divided into two categories [10]: one is to reduce the fluctuation of the power curve to reduce the amount of leaked information [11, 12, 13, 14, 15, 16]; another method is to destroy the data relevance between the power curve and key by increasing the redundant power consumption or random noise [17, 18, 19, 20, 21, 22, 23]. The technology of Random Delay Insertion (RDI) is proposed to reduce the correlation between the power consumption and intermediate processed data. At first Random Delay Insertion is implemented via software methods, which are mostly implemented within the code executed on the processors [24, 25]. On the other hand, in hardware-based methods, the designer of the circuits has more freedom and options to increase the immunity of algorithms. So some researchers designed random delay circuits based on hardware methods [26, 27, 28, 29, 30]. In [28], the randomization of delays is controlled through control signals. But the control signals are part of input data for encryption. So this countermeasure is considered weak and close observation of collected traces, an intelligent adversary can get sensitive information. In [29], four kinds of latch with a randomized propagation delay were used in the data path of SBox. The difference of them lies in the inserting delay chain, which actually is a series of buffers. But there is no detailed description of how to randomize delay chain, and they don't evaluate the CA-resistant measure under some power analysis attacks. An S-box design based on time-delay chaotic systems is proposed in [30]. But the time-delay chaotic system has a complex structure, which will lead to the increase of area and power consumption. Similarly, they don't evaluate the CA-resistant measure.

All the hardware methods suffer from overheads in terms of power consumption and area. Therefore, this paper focuses on the design of CA defense with the lower overhead. In order to design a more reliable and simplified CA defense method, this paper proposes a random delay based parallel S-box AES circuit, our strategy can destroy the conditions of CA by changing the power consumption characteristic of S-box to achieve the goal of resisting the CA.

2. The principle of CA

Based on the Hamming weight model, the power consumption in the implementation of cryptographic algorithm is closely related to the data being processed. The principle of CA is based on the correlation between power consumption and processed data, through the statistical analysis of power consumption curve, we can detect whether a collision occurred or not. When a collision occurs, the relationship between the keys is established according to the known plaintext information. The processes of CA are shown as follows.

2.1 Attack points

S-box is the most common attack target. Take the second and the seventh S-box output of the first round as an example to explain CA, their inputs are denoted as $p_2 \oplus k_2$, $p_7 \oplus k_7$ respectively. CA is shown in Fig. 1.

[1]College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2]College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China
a) zfnuaa@nuaa.edu.cn
b) wunee@nuaa.edu.cn

DOI: 10.1587/elex.16.20190192
Received March 25, 2019
Accepted April 24, 2019
Publicized May 17, 2019
Copyedited June 10, 2019

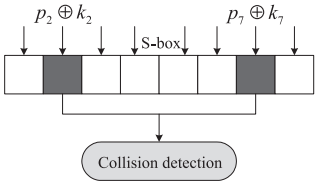

Fig. 1. The attack points of CA

Attacker randomly selects a set of plaintexts as the input of the AES, and obtains a set of power consumption curves, which is the basis for carrying on the statistical analysis and computing the correlation of the two S-boxes power consumption curves. If the correlation coefficient is large, a collision occurs between the S-boxes, so the Eq. (1) can be obtained.

$$\begin{aligned} &S(p_2 \oplus k_2) = S(p_7 \oplus k_7) \Rightarrow \\ &p_2 \oplus k_2 = p_7 \oplus k_7 \Rightarrow p_2 \oplus p_7 = k_2 \oplus k_7 \end{aligned} \tag{1}$$

$S(\,)$ represents the byte substitution operation, p_{i} and k_{i} (i = 2 or 7) denote one byte of plaintext and key respectively. Take other S-box as the attack position, and obtain collision chain by collision attack as Eq. (2), $\Delta_{\mathrm{i,j}}$ $(\mathrm{i,j} = 1, 2, \ldots, 16)$ denote the result of plaintext for p_{i} and p_{j} bitwise xor operation.

$$\begin{cases} k_1 \oplus k_2 = \Delta_{1,2} \\ k_1 \oplus k_3 = \Delta_{1,3} \\ \quad \ldots\ldots \\ k_1 \oplus k_{16} = \Delta_{1,16} \end{cases} \tag{2}$$

2.2 Collision detection

In general, the collision detection method is based on the distance detection method. The general process of distance-based collision detection is as follows: Firstly, the two energy curves are averaged to reduce the influence of noise; the distance between two power consumption curves is obtained. If the distance is less than a certain threshold, it is considered that Collision, otherwise there is no collision. The schematic diagram of the distance-based collision detection method is shown in Fig. 2.

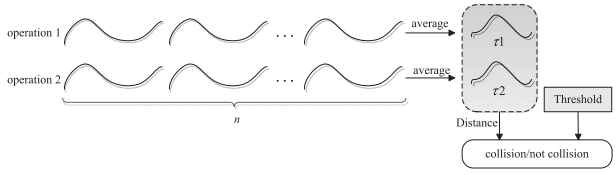

Fig. 2. Collision detection

Operation 1 and 2 are performed n times to obtain two sets of power consumption curves. Two average power consumption curves $\tau 1$, $\tau 2$ are obtained by averaging the power consumption curves of the two sets, r key points are selected on the two curves. The distance between the key points, if the distance is less than the threshold set in advance, the collision occurred otherwise there is no collision. The distance is usually calculated by the Euclidean distance method as expressed in Eq. (3).

$$Dis(\tau 1, \tau 2) = \sum_{i=1}^{r} (\tau 1_i - \tau 2_i)^2 \tag{3}$$

Euclidean distance can better overcome the noise interference, so most of the actual attack selection Euclidean distance method to calculate the distance between two power curves.

3. The proposed parallel random delay S-box AES circuit

The premise of the implementation of collision attacks is that attackers can use the power curve to successfully detect the collision and set a reasonable threshold, the principle is selecting r key points at the same time to calculate the distance between the power consumption curves. So, once the consistency of the power curve is destroyed, the collision determination and threshold settings will be affected, the success rate of collision attacks will be greatly reduced. Based on the analysis above, this paper designs parallel S-box based on random delay; the architecture is shown in Fig. 3.

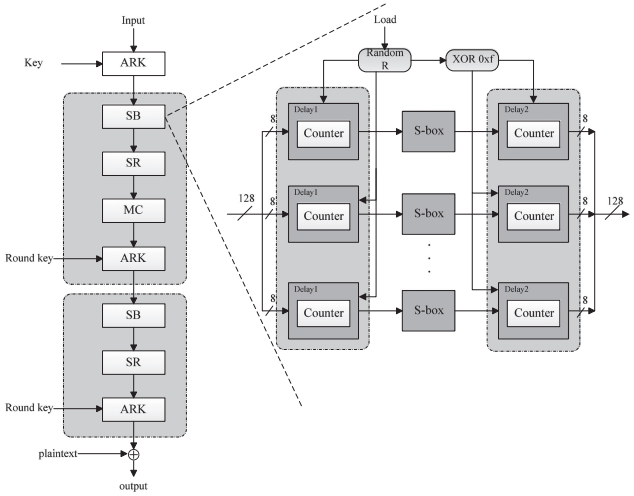

Fig. 3. The parallel random delay S-box AES circuit

The Delay implementation mechanism is a random number counter. When encryption begins, the circuit randomly generates 16 random number called *Delay1*, the range is from 0 to 15 and do not repeat each other and then according to *15-Delay1* generate 16 random numbers as *Delay2*. The random numbers are loaded into each counter in turn. After the ARK (Add Round Key, ARK) is completed, the input part of the counter starts counting. When the counter reaches the set random number, the control signal is pulled high and the result of the ARK is sent to the S-box. Similarly, as S-box operation is completed, the output part of the counter starts counting; when the counter counts to *Delay2*, the S-box operation results will be output.

The generation of traditional random numbers is accomplished through the conversion of various states of the linear feedback shift register (LFSR). In order to simplify the circuit design, the method of generating random num-

bers is to pre-write 16 random numbers in the circuit with a bit width of 4 bits. Each random number plus 1 in per encryption, and each random number is always limited to 0∼15 range, so that it not only ensures the randomness of the data but also simplify the circuit design.

4. Experiments

We realize AES circuit with Verilog HDL, synthesize and simulation AES by Synopsys EDA tool. To test the effectiveness of parallel random delay S-box, we conducted two set of experiments. The first is collision detection to determine the threshold value. The processes are shown as follows.

Setp1: We assume there are two cryptographic chips: Chip_A and Chip_B. Chip_A contains the attacked key, and Chip_B contains the known key. The two are the same except for the key contained.

Step2: We take Chip_B as the experimental object, select the specific plaintext $p_1 = 0$ and $p_1 \oplus p_2 = k_1 \oplus k_2$ (k_1 and k_2 are known). So a collision will occur. This set of plaintexts is encrypted n times to obtain two sets of power consumption curves, which are output of the first and the second S-box. Then we calculate the average of the two power consumption curves $\tau 1$ and $\tau 2$. According to Eq. (3), the value of distance at the time of collision is calculated and expressed as D_0.

Step3: In the same way, we take Chip_B as the experimental object. But we select different plaintext $p_1 = 1$ and p_2 remains unchanged. So a collision will not occur. And then we used the same method to get the distance D_1 when the collision does not happen.

Step4: Repeat Step3, select different plaintext $p_1 = i$ ($i = 2, 3, \ldots, 255$), and get the distance D_i ($i = 2, 3, \ldots, 255$).

Step5: Calculate threshold T_h. Select a value D' which is closest to D_0 from D_i ($i = 2, 3, \ldots, 255$), and set the threshold T_h to $T_h = \frac{D_0 + D'}{2}$.

Step6: Take Chip_A as the experimental object. Input different plaintexts n times, and then select r key points to calculate D. If D is less than the threshold T_h, collision is considered; otherwise, no collision is considered. If no collision is detected, choose other plaintext inputs and repeat Step6 until collision is detected.

In the first experiments, we set $r = 10$, $n = 100$ and Key_B = 128′h2b7e1516_28aed2a6_abf71588_09cf4f3c for Chip_B, perform Step1 to Step5, and calculate the collision threshold T_h is 0.089. And then CA is carried out with Chip_A. We set the random plaintext P = 128′h e5a9ac05_1b60b72a_b415b348_c674b23e and select k_1 as a free variable in the collision chain, so we need to change the input of the plain text $p_2 \sim p_{16}$ in turn so that the second to 16th S-boxes collide with the first S-box respectively. Firstly, we keep plaintext p_1 equal to 0x3e unchanged, p_2 traverses all 256 values from 0x00 to 0xff, perform Step6. As a result, we can get 256 distances D_i ($i = 0, 1, \ldots, 255$) after p_2 is traversal completed, the results are illustrated as Fig. 4.

In Fig. 4, the red dotted line represents the threshold value T_h. And we can see from Fig. 4, D_{51} is below the red

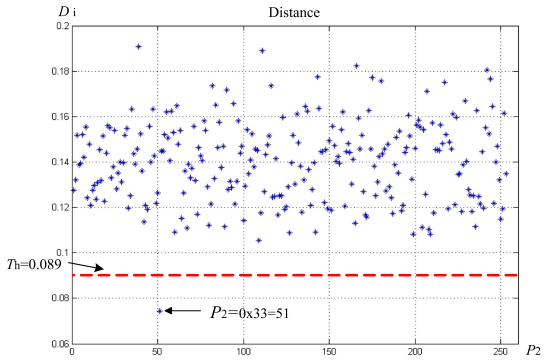

Fig. 4. The distance of the first and the second S-box

dotted line when $p_2 = 0x33$, which means a collision point is detected for CA. And collision doesn't occur with the other values of p_2. According to Eq. (1), we obtain the relationship between k_1 and k_2, which is $k_1 \oplus k_2 = 0x33 \oplus 0x3e = 0x0d$. Similarly, each plaintext byte (from p_3 to p_{16}) traverses all 256 values from 0x00 to 0xff, and the relation between k_i ($\mathrm{i} = 3, 4, \ldots, 16$) and k_1 can be calculated in the same way. We can obtain collision chain by collision attack as Eq. (2).

The second set of experiments is to verify the effectiveness of our parallel random delay S-box. We take our S-box as target to carry out collision attack, do a set of experiments according to the above Step1 to Step5 and calculate the collision threshold T_h as 0.139. We select k_1 as a free variable in the collision chain and want to find the relationship between k_1 and k_2. A plaintext P = 128′h19f48d08_a0c648be_9af8e32b_e93de22a is randomly selected and the above Step6 is performed. As a result, we can get 256 distances D_i ($i = 0, 1, \ldots, 255$) after p_2 is traversal completed, the results are illustrated as Fig. 5.

As we can see from Fig. 5, a plurality of collisions is detected for collision attacks on the proposed S-boxes based on random delays. But as we know, when the values of p_1, k_1 and k_2 are unique deterministic value, there should be only one collision between k_1 and k_2. These indicate that something is wrong with the CA, the collision cannot be correctly detected and the S-box based on random delay has been proved to have a certain ability to withstand collision attacks.

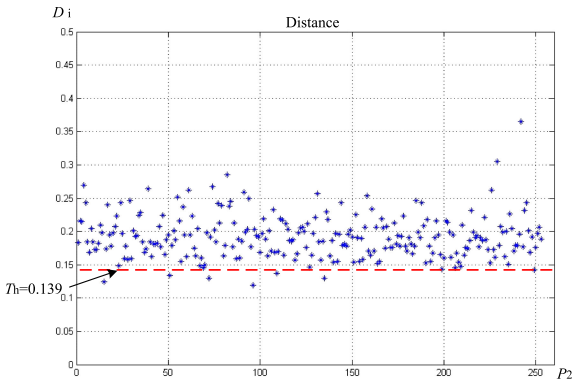

Fig. 5. The distance of the first and the second for the parallel random delay S-box

Besides, to evaluate the overheads in terms of area and power consumption, AES circuit based on the well-known method random mask strategy in [18], inserting delay buffers proposed in [29], and our parallel random delay S-box are synthesized on the Virtex-5 FPGA platform. The performance comparison of three circuits is shown in Table I.

Table I. The performances of AES circuits in different strategies.

Defense strategies	Random mask [18]	Delay buffers [29]	This work
Slices	10637	7259	5844
Power (mW)	11.408	7.985	6.832

From Table I, it is obvious that the defense strategy proposed in this paper reduces the circuit area and power by 45.05% and 40.11% compared to the random mask method, by 19.49% and 14.44% compared to the method in [9]. So our method is a much easier and more efficient way to defense collision attack.

5. Conclusion

This letter presents a random delay method based parallel S-box to resist CA. The S-box circuit designed in this paper destroys the condition of the CA by changing the power consumption characteristics of the S-box. Experiment shows it has the ability to resist the CA. At the same time, compared to the well-known random mask method and other CA countermeasures, our method can defense CA without changing the AES round transformation architecture and bring extra resource overhead.

Acknowledgments

This work is supported by the Natural Science Foundation of Jiangsu Province (No. BK20160806), the Fundamental Research Funds for the Central Universities (NP2019102, NS2017023, NS2016041) and the National Science Foundation of China (No. 61774086).

References

[1] J. Chen, *et al.*: "A collision attack on a double-block-length compression function instantiated with 8-/9-round AES-256," IEICE Trans. Fundamentals **E99.A** (2016) 14 (DOI: 10.1587/transfun.E99.A.14).
[2] Y. Ren, *et al.*: "Double sieve collision attack based on bitwise detection," KSII Trans. Internet Inf. Syst. **9** (2015) 296 (DOI: 10.3837/tiis.2015.01.016).
[3] A. Bogdanov and I. Kizhvatov: "Beyond the limits of DPA: Combined side-channel collision attacks," IEEE Trans. Comput. **61** (2012) 1153 (DOI: 10.1109/TC.2011.140).
[4] K. Schramm, *et al.*: "A new class of collision attacks and its application to DES," Fast Software Encryption (2003) 206 (DOI: 10.1007/978-3-540-39887-5_16).
[5] A. Moradi, *et al.*: "Correlation-enhanced power analysis collision attack," Cryptographic Hardware and Embedded Systems, CHES 2010 (2010) 125 (DOI: 10.1007/978-3-642-15031-9_9).
[6] C. Clavier, *et al.*: "Improved collision-correlation power analysis on first order protected AES," International Conference on Cryptographic Hardware and Embedded Systems (2011) 49 (DOI: 10.1007/978-3-642-23951-9_4).
[7] H. S. Kim and S. Hong: "New type of collision attack on first-order masked AESs," ETRI J. **38** (2016) 387 (DOI: 10.4218/etrij.16.0114.0854).
[8] B. Gérard and F.-X. Standaert: "Unified and optimized linear collision attacks and their application in a non-profiled setting," Cryptographic Hardware and Embedded Systems (CHES) (2012) 175 (DOI: 10.1007/978-3-642-33027-8_11).
[9] D. Wang, *et al.*: "Fault-tolerant linear collision attack: A combination with correlation power analysis," 10th Information Security Practice and Experience Conference (ISPEC 2014) (2014) 232 (DOI: 10.1007/978-3-319-06320-1_18).
[10] H. Marzouqi, *et al.*: "Review of gate-level differential power analysis and fault analysis countermeasures," IET Information Security **8** (2014) 51 (DOI: 10.1049/iet-ifs.2012.0319).
[11] D. D. Hwang, *et al.*: "AES-based security coprocessor IC in 0.18 µm CMOS with resistant to differential power analysis side-channel attacks," IEEE J. Solid-State Circuits **41** (2006) 781 (DOI: 10.1109/JSSC.2006.870913).
[12] T. Popp and S. Mangard: "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," CHES (2005) 172 (DOI: 10.1007/11545262_13).
[13] N. E. C. Akkaya, *et al.*: "A DPA-resistant self-timed three-phase dual-rail pre-charge logic family," 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (2015) (DOI: 10.1109/HST.2015.7140248).
[14] X. Pang, *et al.*: "A DPA resistant dual rail Préchargé logic cell," IEEE International Conference on ASIC (2016) (DOI: 10.1109/ASICON.2015.7517071).
[15] Y. Zhang, *et al.*: "High performance AES-GCM implementation based on efficient AES and FR-KOA multiplier," IEICE Electron. Express **15** (2018) 20180559 (DOI: 10.1587/elex.15.20180559).
[16] W. Tang, *et al.*: "Dual-voltage single-rail dynamic DPA-resistant logic based on charge sharing mechanism," Chin. J. Electron. **26** (2017) 899 (DOI: 10.1049/cje.2017.03.003).
[17] M.-L. Akkar and C. Giraud: "An implementation of DES and AES, secure against some attacks," CHES (2001) 309 (DOI: 10.1007/3-540-44709-1_26).
[18] Y. Ye, *et al.*: "An optimized design for compact masked AES S-box based on composite field and common subexpression elimination algorithm," J. Circuits Syst. Comput. **27** (2018) 1850171 (DOI: 10.1142/S0218126618501712).
[19] Y. Liu, *et al.*: "A new compact hardware architecture of S-box for block ciphers AES and SM4," IEICE Electron. Express **14** (2017) 20170358 (DOI: 10.1587/elex.14.20170358).
[20] A. Roy and S. Vivek: "Analysis and implement of the generic higher-order masking scheme of FSE 2012," CHES (2013) 417 (DOI: 10.1007/978-3-642-40349-1_24).
[21] A. J. Leiserson, *et al.*: "Gate-level masking under a path-based leakage metric," CHES **8731** (2014) (DOI: 10.1007/978-3-662-44709-3_32).
[22] C. Herbst, *et al.*: "An AES smart card implementation resistant to power analysis attacks," ACNS (2006) 239 (DOI: 10.1007/11767480_16).
[23] J. Zhang, *et al.*: "Against transient-steady effect attack using time check blocks," 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA) (2017) 436 (DOI: 10.1109/ICIEA.2017.8282884).
[24] V. Rashtchi and S. H. Mosavi: "Strengthened of AES encryption algorithms within new logic topology," Majlesi J. Electr. Eng. **12** (2018) 87.
[25] F. Durvaux, *et al.*: "Efficient removal of random delays from embedded software implementations using hidden Markov models," CARDIS 2012: Smart Card Research and Advanced Applications (2012) 123 (DOI: 10.1007/978-3-642-37288-9_9).
[26] J.-S. Corona and I. Kizhvatov: "Analysis and improvement of the random delay countermeasure of CHES 2009," International Conference on Cryptographic Hardware & Embedded Systems (2010) (DOI: 10.1007/978-3-642-15031-9_7).
[27] S. Kumar K, *et al.*: "Analysis of side-channel attack AES hardware Trojan benchmarks against countermeasures," 2017 IEEE Com-

puter Society Annual Symposium on VLSI (ISVLSI) (2017) 574 (DOI: 10.1109/ISVLSI.2017.106).
[28] Y. Lu, *et al.*: "Evaluation of random delay insertion against DPA on FPGAs," ACM Trans. Reconfig. Technol. Syst. **4** (2010) 11 (DOI: 10.1145/1857927.1857938).
[29] Z. Liu, *et al.*: "A high-security and low-power AES S-box full-custom design for wireless sensor network," 2007 International Conference on Wireless Communications Networking and Mobile Computing (2007) 2499 (DOI: 10.1109/WICOM.2007.622).
[30] F. Özkaynak and S. Yavuz: "Designing chaotic S-boxes based on time-delay chaotic system," Nonlinear Dyn. **74** (2013) 551 (DOI: 10.1007/s11071-013-0987-4).