SCIENCE CHINA
Information Sciences

• LETTER •

March 2020, Vol. 63 139102:1–139102:3
https://doi.org/10.1007/s11432-018-9818-y

Theoretical analysis of persistent fault attack

Fan ZHANG[1,2,3,4], Guorui XU[1,3,4*], Bolin YANG[1], Ziyuan LIANG[1,3,4] & Kui REN[3,4]

[1]*College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China;*
[2]*State Key Laboratory of Cryptology, P.O.Box 5159, Beijing 100878, China;*
[3]*School of Cyber Science and Technology, Zhejiang University, Hangzhou 310027, China;*
[4]*Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou 310027, China*

Received 30 December 2018/Accepted 1 March 2019/Published online 10 February 2020

Citation Zhang F, Xu G R, Yang B L, et al. Theoretical analysis of persistent fault attack. Sci China Inf Sci, 2020, 63(3): 139102, https://doi.org/10.1007/s11432-018-9818-y

Dear editor,

Persistent fault attack is a new type of fault attack that was proposed in CHES 2018 [1]. Its success relies on the so-called persistent fault which persists from one encryption to another and disappears when the target device reboots. The practical attack is illustrated in [1] at both software and hardware levels, however, the corresponding persistent fault analysis (PFA) has not been thoroughly investigated. The contribution of this study focuses on the theoretical analysis of the persistent fault attack. This study also proposes a new and generic method which could put the analysis of previous cases into one nutshell.

PFA. The dual modular redundancy (DMR) countermeasure is designed to prevent traditional fault attacks [2]. However PFA can bypass both redundant encryption based DRM (REDMR) and inversive decryption based DMR (IDDMR) as shown in [1]. When a persistent fault injected in an S-Box used in AES-128, the output ciphertexts can be different in the statistical perspective, even with the existence of DMR countermeasures.

In the simplest case, the fault leads to an impossible value in the specific output bytes of substitution layer. Since this output will be XORed by a fixed round key in multiple encryptions, this impossible byte will be easily detected in the faulty ciphertexts. The adversary can focus on the distribution of possible values for specific faulty ciphertext byte, which is no longer an even distribution due to the induced persistent fault [1]. If enough ciphertexts are collected, the difference among those values can be distinguished using statistical methods, and the corresponding key byte for the last round key can be extracted.

Previous work. Suppose the total number of possible values for substitution table is η where $\eta = n - 1 = 2^b - 1$. b is the block size and n is the table size. For AES-128 S-Box, $b = 8$ and $n = 256$. The values ranges from `0x01` to `0xff`, assuming the fault is injected to the first element of S-Box. Zhang et al. [1] assume that the bytes in faulty ciphertexts are chosen from η values with equal probability of $\frac{1}{\eta}$, due to the randomness of plaintexts and the avalanche effect of AES.

Suppose the adversary can collect i ciphertexts after the encryption. Parts of them are faulty. Let θ_i be the "average" number of different values that he has observed for one specific ciphertext byte. According to [1], θ_i can be calculated as in (1) which is detailed in Appendix A.

$$\theta_i = \frac{1 - q^i}{1 - q}, \quad \text{where } q = \frac{\eta - 1}{\eta}. \tag{1}$$

There are still some problems that have not been addressed in [1]. First, it only stated that θ_i will converge to $\eta = 255$ eventually, but did not give a theoretical expectation value of i when θ_i equals η for the first time. The minimal number of traces on average required to extract that key byte is of more interests. Second, in fact, θ_i is the expectation of a series of random variables that are not independent to each other. So the

* Corresponding author (email: xugr@zju.edu.cn)

expectation should not be simply added together as in [1]. Moreover, the probability of each value is not the same. For two special values, the corresponding probabilities are 0 and $\frac{2}{n}$, indeed.

Problem statement. Suppose one original value of the target S-Box is s_0, which changes to s_1 because of the induced fault. The goal of this study is to give a theoretical calculation of i, i.e., the average number of ciphertexts that are required to identify s_0 or s_1. More precisely, the expectation of i is desired, which also shows how difficult the attack it is, to some extent.

The problem can be categorized into three cases according to the probability distribution introduced in [1], which are summarized in Figure 1(a). Case 1 shows the exact distribution discussed in last part and also as the vanilla case in [1]. s_1 has the same probability as the rest, while s_0 never appears. Note that there are no countermeasures. Case 2 shows the scenario where countermeasures, such as no ciphertext output (NCO) or zero value output (ZVO) [1], are deployed. Without loss of generality, the probability for s_1 is $\frac{2}{n}$, while that for the rest is $\frac{1}{n}$ instead of $\frac{1}{n-1}$ in Case 1. Case 3 shows a more advanced scenario where IDDMR countermeasure is deployed and a random ciphertext output (RCO) [1] will be given. s_0 still appears in the ciphertext bytes with a probability of $\frac{1-p}{n}$ while s_1 holds a probability of $\frac{1+p}{n}$. p is the probability threshold to differentiate either s_0 or s_1 from the rest.

Case	s_0	s_1	s_2	$\cdots$	s_{n-1}
1	0	$1/(n-1)$	$1/(n-1)$	$\cdots$	$1/(n-1)$
2	0	$2/n$	$1/n$	$\cdots$	$1/n$
3	$(1-p)/n$	$(1+p)/n$	$1/n$	$\cdots$	$1/n$

(a)

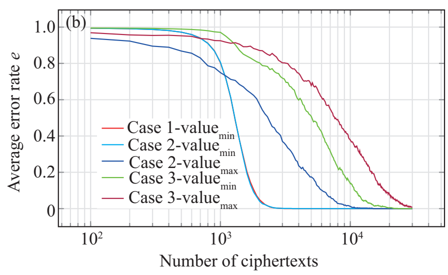

Figure 1 (Color online) (a) Distribution of values in ciphertexts; (b) error rate and number of ciphertexts for all cases.

Theoretical analysis. Among all cases, s_0 has probability difference from others. The attack strategy is to identify the special values when increasing the number of ciphertexts.

Case 1. This case can be interpreted as the classic coupon collector's problem (CCP) [3] which can be described as following: a company issues different types of coupons, each type having a certain probability of being issued. The CCP asks for the expected number of coupons that need to be gathered before a full collection is obtained.

In fact, the ciphertexts can be equivalently treated as the coupons in CCP, and the values observed in the ciphertext bytes can be viewed as the types of those coupons.

Suppose t_j denotes the number of additional ciphertexts required after $j-1$ different values have been observed. T denotes the total number of the ciphertexts required when all possible values have been observed. $T = \sum_{j=1}^{n-1} t_j$ and $t_1 = 1$.

Since the distribution is uniform, when the $(j-1)$-th value has been observed, the next new value comes with a fixed probability $p_j = \frac{n-j}{n-1}$. Also, variables t_j are independent. The expectation of T denoted as $E(T)$, i.e., the expected number of ciphertexts required, can be calculated as in (2). Note that H_n is the n-th harmonic number and $H_n = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}$. More details about (2) are listed in Appendix B. When $n = 256$, the expected number of ciphertexts is about 1565.88.

$$E(T) = (n-1) \times H_{n-1}. \qquad (2)$$

Case 2. In this case, the probability of s_1 is doubled against those of the rest. This problem can be viewed as a general version of CCP. The expected number of ciphertexts can be calculated using similar means proposed in [4], as shown in

$$E(T) = \sum_{k=1}^{n-2} (-1)^{k+1} \left(\binom{n-2}{k} \frac{n}{k} + \binom{n-2}{k-1} \frac{n}{k+1} \right) + (-1)^n. \qquad (3)$$

For a large n, the formula for $E(T)$ in (3) is hard to compute. Using the idea of generating function (see Appendix C), Eq. (3) can be transformed into (4) as follows. Note that Eq. (4) is a precise calculation for $E(T)$ and can be directly computed much easier. When $n = 256$, the expected number of ciphertexts is about 1560.70.

$$E(T) = (-1)^n - n \int_0^1 \frac{(1-x)^{n-2} - 1}{x} \mathrm{d}x + n \int_0^1 (x(1-x)^{n-1} - (-1)^n x^{n-1}) \mathrm{d}x. \qquad (4)$$

Case 3. In this case, PFA can defeat IDDMR with RCO [1]. When the fault is detected, the output will be random, which makes every value occurs with a non-zero probability in faulty ciphertexts.

To recover the key, the algorithm in [1] is to perform the statistics by gradually increasing the number of ciphertexts until the frequency of some values reached an empirical threshold. When the threshold is set to an appropriate value, the extracted key has a large probability to be the real key used in the encryption.

Different from [1], this study proposes a new method to find s_0 and s_1 without thresholds: simply find the bytes which holds the smallest (or largest) frequency as shown in Algorithm 1. Note that the outputs of Algorithm 1 (i.e., $\text{value}_{\min}$ and $\text{value}_{\max}$) may not be the ones associated with the correct key, whose probability can be denoted as the error rate e_i. It is too difficult to formalize a unified formula for $E(T)$ with a fixed e_i as (4). Therefore, this computation of e_i is approximated via a simulation, which will reveal the relation between the number of ciphertexts and e_i.

Algorithm 1 Pseudo code to distinguish two special values

Require: $n, j, \text{value}[i]$ $(0 \leqslant i \leqslant j)$;
count$[n] \Leftarrow [0..0]$; $\text{value}_{\min} \Leftarrow 0$; $\text{value}_{\max} \Leftarrow 0$;
for $i \Leftarrow 0$ **to** $j-1$ **do**
    count[value$[i]$] $\Leftarrow$ count[value$[i]$] $+ 1$;
end for
for value $\Leftarrow 0$ **to** $N-1$ **do**
    if count[value] $\leqslant$ count[$\text{value}_{\min}$] **then**
        $\text{value}_{\min} \Leftarrow$ value;
    else if count[value] $>$ count[$\text{value}_{\max}$] **then**
        $\text{value}_{\max} \Leftarrow$ value;
    end if
end for
return $\text{value}_{\min}, \text{value}_{\max}$.

More importantly, this method is generic to be applied to three cases. All corresponding statistical analysis can be unified under one nutshell. In the first two cases, if the adversary collects enough ciphertexts, most of values will be eliminated as impossible ones. The last remained value (s_0 or s_1) is uniquely determined, which must be the correct one. But in Case 3, the error rate cannot be 0. For instance, in one attack, a value s_k may be associated with the smallest frequency. But s_k may not be equal to $\text{value}_{\min}$. However, the confidence of the probability that s_k equals can be evaluated.

Simulation. To further verify the theoretical result of our analysis, we implemented a simulation of the problem. The goal of the simulation is to find the relationship between the average error rate and the number of ciphertexts. For a fixed number of ciphertexts, Algorithm 1 can be executed for many times to compute the average error rate.

Figure 1(b) shows the relationship between the error rate and the number of the ciphertexts required to recover the first key byte. With about 3×10^4 ciphertexts, the error rate e_i is reduced to 0 (or close to 0 approximately) in all curves. Note that those curves for Case 3 are independent of p because no thresholds are used. This figure shows that the attack's ultimate success based on enough ciphertexts that are collected. The total number of required ciphertexts (at 10^4 level) is quite reasonable and acceptable in practice.

Case 1 has only one curve since the values of $\frac{1}{n-1}$ are applied to all s_i except s_0 and no maximal value can be distinguished from others as for s_1. Both Cases 2 and 3 have two curves in corresponding to $\text{value}_{\min}$ and $\text{value}_{\max}$, respectively.

$\text{value}_{\min}$s in Cases 1 and 2 have little difference, as the distribution is almost the same. Both the simulation and the computed expectation show that Case 1 is a good estimation of Case 2. With similar expectation and error rate curve, Case 1 is much easier to be computed and derived.

Results of Cases 2 and 3 show that finding the value which has the smallest probability is a better way to extract the key effectively and efficiently. This preference can be applied in the real PFA.

As for Case 3, when the number of ciphertexts is about 1.3×10^4, e can be less than 3%; when the number is about 1.2×10^4, e is 6.2% as shown in Figure 1(b). The error rate is small enough to extract the key in practice. This is consistent to the results of practical attacks mentioned in [1], which states that $12000 \sim 13000$ ciphertexts are required.

Conclusion. The study focuses more on the theoretical calculation of those complexities and probability used in [1]. Our results are consistent to those in [1] but with more formal computation extended from the well-known CCP problem.

Acknowledgements This work was supported in part by Open Fund of State Key Laboratory of Cryptology (MMKFKT201805), Zhejiang Key R&D Plan(2019C03133), Major Scientific Research Project of Zhejiang Lab (2018FD0ZX01), Young Elite Scientists Sponsorship Program by CAST(17-JCJQ-QT-045), Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhang F, Lou X X, Zhao X J, et al. Persistent fault analysis on block ciphers. IACR Trans Cryptograph Embed Syst, 2018, 2018: 150–172
2 Joye M, Tunstall M. Fault Analysis in Cryptography. Berlin: Springer, 2012
3 Ferrante M, Saltalamacchia M. The coupon collector's problem. Mater Matemàtics, 2014, 2014: 1–35
4 Flajolet P, Gardy D, Thimonier L. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. Discrete Appl Math, 1992, 39: 207–229