SCIENCE CHINA
Information Sciences

• LETTER •

March 2018, Vol. 61 039106:1–039106:3
https://doi.org/10.1007/s11432-017-9179-4

Optimal model search for hardware-trojan-based bit-level fault attacks on block ciphers

Xinjie ZHAO[1], Fan ZHANG[2,3*], Shize GUO[1] & Zheng GONG[4]

[1]*Institute of North Electronic Equipment, Beijing* 100000*, China;*
[2]*College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou* 310027*, China;*
[3]*School of Computing, National University of Singapore, Singapore* 117417*, Singapore;*
[4]*School of Computer Science, South China Normal University, Guangzhou* 510631*, China*

Received 22 April 2017/Accepted 7 June 2017/Published online 5 January 2018

Citation Zhao X J, Zhang F, Guo S Z, et al. Optimal model search for hardware-trojan-based bit-level fault attacks on block ciphers. Sci China Inf Sci, 2018, 61(3): 039106, https://doi.org/10.1007/s11432-017-9179-4

Dear editor,

Fault analysis is a very powerful technique used to break cryptographic implementations. In particular, bit-level fault attacks (BLFAs), where one or a few isolated bits are flipped to inject faults, are among the most efficient of the lot. Because it requires both precise fault injection and sophisticated key extraction, a BLFA is very difficult to conduct in practice. However, if the underlying cryptographic hardware is maliciously modified, a BLFA can be easily achieved. This recent security threat is popularly known as a hardware Trojan horse (HTH) [1]. An HTH is a byproduct of the very popular and economically necessary outsourcing trend in the semiconductor industry. A well-designed HTH can precisely inject any type of bit-level fault. The corresponding attack is called a hardware-Trojan-based bit-level fault attack (HTH-BLFA).

In [2], an HTH was designed to flip a nibble or byte of cryptographic states. However, the fault model in [2] is simply adopted from those well studied in differential fault analysis (DFA) [3] on block ciphers. A desired fault model has a significant influence on the overall attack. Identifying the optimal fault model is crucial to the cryptanalysis efficiency of BLFAs for further investigation.

In this letter, we first depict the optimal fault model in a BLFA. Then we propose three metrics that could effectively enhance the optimal model search. Four steps are elaborated for a practical HTH-BLFA. Finally, we use PRESENT-80 [4] implemented on SASEBO-GII to prove our technique. The HTH is triggered only once to inject one nibble fault and the 80-bit key space can be reduced to $2^{20.1}$ on average.

Optimal fault model for an HTH-BLFA. In a block cipher $\mathbb{B}$, P, C, C^*, m, n, and r denote the plaintext, correct ciphertext, faulty ciphertext, block size, S-box size, and the total number of rounds, respectively. For simplicity, we assume only one type of S-box is used throughout the whole cipher. Let λ denote the number of S-box lookups in one round. Then, $\lambda = m/n$. $X_{i,j}$ is a one-bit intermediate state where i is the index of the round ($1 \leqslant i \leqslant r$) and $j = \{d_1, \ldots, d_k\}$ is a set of indexes for k bit flips in the state ($1 \leqslant d_1, \ldots, d_k \leqslant m$). Then, the fault model can be described as $\mathbb{F} = X_{i,j} = \{X_{i,d_1}, \ldots, X_{i,d_k}\}$ by assuming all bit flips are to the same round i. Given a specific value of k, the optimal model is denoted as $\mathbb{F}_o = X_{I_o,J_o}$, where I_o is the round index and J_o is the bit index for $\mathbb{F}_o$. The search process is to find a specific assignment $\{I_o, J_o\}$ among all possible $\{i, j\}$ with which $\mathbb{F}_o$ is considered as the best choice for the subsequent HTH design and the offline cryptanalysis. Evaluating whether a model is optimal can be considered from two as-

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

© Science China Press and Springer-Verlag GmbH Germany 2018

pects. The first is the attack complexity, such as the data complexity (i.e., the number of injections, N), the time complexity (i.e., the solving time T), and the memory complexity (i.e., the number of key enumerations, δ). The second is the key guessing metric, such as the remained key entropy $\phi(K)$ (where $\phi(K) = 0$ means that the value of K is recovered and the search space of K is only one).

Metrics of the model search. Because of the stealthiness of the HTH, small k and i are preferred. The choice of k depends on the adversary's sophistication of HTH design and offline analysis. A small i is also desired because an injection to the deeper round will propagate more faults to the ciphertext and bring more informative entropy for key analysis. Because of the efficiency of the offline cryptanalysis, small values of N, T, δ, and $\phi(K)$ are desired. In this letter, a single fault injection ($N = 1$) with affordable analysis time and memory is targeted. However, it is difficult to balance the trade-off among of N, T, δ, and $\phi(K)$ because both the faults in the final ciphertext and the propagation paths are highly overlapped. It is important to define some metrics to accelerate the search process.

For a specific model $\mathbb{F}$, an HTH-based fault injection can be simulated as software-level bit flips in advance, which is referred to as an instance. Let q denote the index of the injection round. Let A_i denote the number of active S-boxes in the ith round. Whether $\mathbb{F}$ is optimal or not highly depends on two metrics: η_q, the average number of A_r, and D_q, the depth of the fault. $D_q = r-q+1$.

From the view of information theory, η_q should be maximized. The more faults that are propagated to the final round, the more entropy related to the subkey is provided for the offline analysis, resulting in a small $\phi(K)$. To maximize η_q, a straightforward solution is to move the injection round as deep as possible. However, this will also increase the difficulty of the offline analysis, resulting in an increase of N, T, δ, and $\phi(K)$. From the view of attack complexity, a smaller value of D_q is preferable. In this case, the representation of fault leakages and the problem solving become easier, so that N, T, and δ are small. Both η_q and D_q should simultaneously qualify the performance of the optimal model search. In this letter, we introduce a combinational metric $\theta_q = \frac{\eta_q}{D_q}$ that denotes the speed of the fault propagation in the last D_q rounds.

HTH-BLFA procedure.

Step 1. Determine I_o. For each possible round i, we randomize the set of j and simulate the fault injection for L times. Then, η_q, D_q, and θ_q can be calculated. The optimal injection round I_o is the round i with maximal θ_q.

Step 2. Determine J_o. In the I_oth round determined in Step 1, we check each candidate of j and simulate the fault injection. J_o is determined as the j where η_q is maximized for the same D_q.

Step 3. Verify $\mathbb{F}_o = X_{I_o,J_o}$. Different techniques can be applied, e.g., DFA [3] or algebraic fault analysis (AFA) [5]. For each $\mathbb{F}_o$, DFA requires manual cryptanalysis on the fault propagation path and is efficient for heavy block ciphers. AFA combines algebraic analysis with DFA and can be conducted automatically for different $\mathbb{F}_o$. It is efficient for lightweight block ciphers. In this letter, we apply AFA to verify $\mathbb{F}_o$. In AFA, the bit flips specified by $\{I_o, J_o\}$ are simulated by modifying the source code of the targeted cipher $\mathbb{B}$. The algebraic equations for both $\mathbb{B}$ and $\mathbb{F}_o$ are built with an automatic tool. Finally, a machine solver is applied to solve the equations. The solver can output all satisfiable solutions and their corresponding solving times. $\mathbb{F}_o$ with small values of N, T, and $\phi(K)$ are preferred.

Step 4. Design and implement the HTH. A lightweight HTH can be designed to inject the fault specified by $\mathbb{F}_o$ into a circuit. A typical HTH is composed of payload logic (PL) to modify the integrated circuit and trigger logic (TL) to activate the PL. The PL flips k bits of the original circuit. In field-programmable gate array (FPGA) platforms, it can be implemented as k or fewer XOR gates with look-up tables. Once the correct and faulty ciphertexts, C and C^*, are collected, the adversary can conduct AFA to extract the key.

Application to PRESENT-80. We take the block cipher PRESENT-80 [4] as an illustration. It is a 31-round cipher with a 64-bit block size, an 80-bit key size, and a 4-bit S-box size ($m = 64$, $n = 4$, $\lambda = 16$, and $r = 31$). To make as many active S-boxes as possible in the last round, four bits of one S-box output (one nibble) are flipped. Thus $k = 4$ and $j = \{4\alpha, 4\alpha+1, 4\alpha+2, 4\alpha+3\}$ $(0 \leqslant \alpha < 16)$.

In Step 1, for each i we randomly choose α for 100000 times with different plaintexts (for each instance, $N = 1$) and calculate A_i.

The results are shown in Figure 1(a). When q is close to $r = 31$, η_q is much smaller, which means that $\phi(K)$ is larger. When $q < 28$, the values of A_i in the last two or three rounds are all close to 15, which means that the time complexity T and the memory complexity δ are quite large for key elimination. Figure 1(b) presents θ_q, the fault propagation speed in the last D_q rounds. θ_{28} is the maximum. Thus, $I_o = 28$.

In Step 2, when $I_o = 28$, for each α, we collect 100000 instances and calculate η_{28}. The results

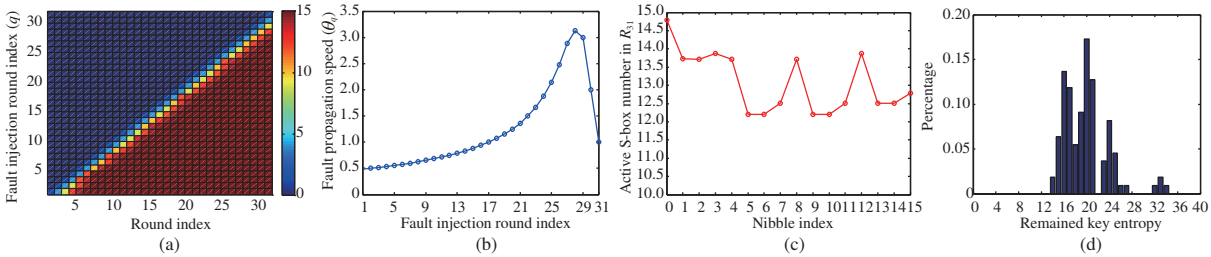

Figure 1 (Color online) Optimal model search for an HTH-BLFA on PRESENT-80. (a) Faulty nibble number in each round (A_i); (b) determining the optimal injection round I_o by θ_q; (c) determining the optimal injection location J_o ($I_o = 28$); (d) statistics of $\phi(K)$ ($I_o = 28, J_o = 0, 1, 2, 3$).

are shown in Figure 1(c). When $\alpha = 0$, the value of η_{28} is maximized. This is mainly caused by the diffusion and confusion features of PRESENT-80 for different fault nibbles. Thus, $J_o = \{0, 1, 2, 3\}$.

In Step 3, to evaluate N, T, and $\phi(K)$, we convert the block cipher PRESENT-80, C and C^* into equations and conduct AFA. The machine solver is CryptoMiniSAT v2.9.4, which runs on a PC with an Intel Core i5-4460 at 3.2 GHz and 4 GB of RAM. The attack is repeated 100 times with $N = 1$. The statistics of $\phi(K)$ are shown in Figure 1(d), where $\phi(K)$ is in the range $[14.2, 34]$ and the average value is 20.1. The average solving time is 30 min.

In Step 4, both the HTH and PRESENT-80 are implemented on SASEBO-GII with a 65-nm Virtex-5 FPGA. The PL is carefully designed to flip four bits, i.e., $X_{28,0}$, $X_{28,1}$, $X_{28,2}$, and $X_{28,3}$, and output the faulty ciphertext. The PL disables the loading operation of the new plaintext and encrypts the previous plaintext again to output the correct ciphertext. Since $N = 1$, the TL is implemented by exploiting the temperature sensor in SASEBO-GII. The Trojan will be activated only once when the temperature is $> 42°$C, which is easily accomplished by using a hair dryer.

Conclusion. This letter details how to search the optimal fault model for HTH-BLFAs on block ciphers. The proposed technique is verified with an application to the PRESENT-80 block cipher with a substitution-permutation network (SPN) structure. Meanwhile, we also applied these metrics to the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES). For the AES, just one single-bit fault injection on the 8th round can reduce the 128-bit key to a 1.03-bit key on average. For the DES, just one single-bit fault injection on the 12th round can reduce the 56-bit key to a 5.58-bit key on average. Details of the DES attack are provided in Appendix A.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61309021, 61472357, 61571063, 61572028), China Scholarship Council (Grant No. 201606325012), and Project of Science and Technology of Guangdong Province (Grant Nos. 2014A030313439, 2016B010125002).

Supporting information Appendix A. 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 Bhunia S, Hsiao M, Banga M, et al. Hardware trojan attacks: threat analysis and countermeasures. Proc IEEE, 2014, 102: 1229–1247
2 Johnson A, Saha S, Chakraborty R, et al. Fault attack on AES via hardware Trojan insertion by dynamic partial reconfiguration of FPGA over ethernet. In: Proceedings of the 9th Workshop on Embedded Systems Security, New Delhi, 2014. 1–8
3 Biham E, Shamir A. Differential fault analysis of secret key cryptosystem. In: Proceedings of the 36th International Cryptology Conference. Berlin: Springer, 1997. 513–525
4 Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, 2007. 450–466
5 Zhang F, Zhao X J, Guo S Z, et al. Improved algebraic fault analysis: a case study on piccolo and applications to other lightweight block ciphers. In: Proceedings of the 4th International Conference on Constructive Side-Channel Analysis and Secure Design, Paris, 2013. 62–79