SCIENCE CHINA
Information Sciences

• HIGHLIGHT •

April 2017, Vol. 60 048102:1–048102:3
doi: 10.1007/s11432-016-0233-0

Low-cost design of stealthy hardware trojan for bit-level fault attacks on block ciphers

Fan ZHANG[1,2], Xinjie ZHAO[3], Wei HE[4*], Shivam BHASIN[4] & Shize GUO[3]

[1]*College of Information Science and Electrical Engineering, Zhejiang University, Hangzhou* 310027*, China;*
[2]*Science and Technology on Communication Security Laboratory, Chengdu* 610041*, China;*
[3]*Institute of North Electronic Equipment, Beijing* 100191*, China;*
[4]*Temasek Laboratories, Nanyang Technological University, Singapore* 637371*, Singapore*

Received April 10, 2016; accepted August 12, 2016; published online January 21, 2017

Citation Zhang F, Zhao X J, He W, et al. Low-cost design of stealthy hardware trojan for bit-level fault attacks on block ciphers. Sci China Inf Sci, 2017, 60(4): 048102, doi: 10.1007/s11432-016-0233-0

Fault analysis is a very powerful technique to break cryptographic implementations. In particular, bit-level fault analysis (BLFA), where faults are injected by flipping one or a few isolated bits, are among the most efficient of the lot. BLFA requires both precise fault injection capabilities and sophisticated key extraction skills. Algebraic fault analysis (AFA) [1] is a good analysis technique for BLFA. Compared with differential fault analysis (DFA), AFA relies on the automation from machine solvers. Since it fully utilizes the leakages along propagation paths, it can extract the whole key when there is only one or a few bits infected, and when the injection is into the much deeper rounds. In practice, it is very difficult to inject precise bit-level faults and expensive equipments are indeed required. However, if the underlying cryptographic hardware is maliciously modified, BLFA can be easily achieved. This recent security threat is popularly known as Hardware trojan horse (HTH) [2]. HTH is a by-product of much popular and economically necessary outsourcing trend in semiconductors. A well designed HTH can precisely inject any type of faults to enable AFA and bypass detections, by having low cost and with low activation rate.

This article presents a case study on the design of a highly efficient HTH which can use offline fault analysis to extract the secret key with single activation. The motivation of this work is to (1) push the limits of fault injection to as low as ONE, however, guarantee enough entropy for offline analysis, and (2) demonstrate techniques to design extremely low-cost and stealthy HTH in cryptographic hardware. Compared with the laser-induced fault injection, the advantages of our attack are the precision of the injection, the low cost and its non-invasion. The subsequent AFA is also practical.

Attack model. The considered attack model focuses on a cryptographic intellectual property (IP) with advanced protections like sensors from an untrusted IP vendor or system integrator. Deploying physical sensors alongside cryptographic IP is a common practice in industrial designs. The vendor is capable of inserting a small but functional HTH to enable BLFA. Under such practical scenarios, we demonstrate the design of a HTH to perform efficient AFA. The prototype is on a Xilinx FPGA device implementing a cryptographic IP $\mathbb{B}$. An adversary A can either modify the RTL or the corresponding logic elements in the post place and route netlist. Inserting a HTH in RTL is straightforward and obvious. We demonstrate the versatility of the

* Corresponding author (email: he.wei@ntu.edu.sg)
The authors declare that they have no conflict of interest.

technique. The initial design is done by the original manufacturer. The adversary A only has the access to the XDL file and no access to the design stage. In this case study, we use a temperature sensor to trigger the trojan that can be affected by a cheap equipment like hair dryer. The adversary A has no control on input plaintexts and can only observe final ciphertexts.

HTH design. The ideal location of inserting the HTH is determined by AFA. A HTH has two components: trigger logic (TL) and payload logic (PL). The TL is realized by a temperature sensor while the PL is a simple `XOR` gate. Few extra gates are deployed to attain the conditions required for AFA, as described as follows.

Stage 1: Location selection. Let $X_{i,j}$ be a one-bit state in $\mathbb{B}$ which is a block cipher of r rounds with m-bit block size and n-bit key size. i is the index of the round R_i. j is the index of possible bit flip. $1 \leqslant i \leqslant r$, $0 \leqslant j < m$. Two properties are desired: (i) The reduced key search space $\phi(K)$ after the injection to $X_{i,j}$ should be minimized. (ii) $X_{i,j}$ should be in a deeper round to maximize the fault propagation and to evade the detection. AFA is used to enumerate every possible (i, j) and search for the optimal location for the HTH. The attempts are conducted in advance, which can guide the following logic designs and reduce costs. Each enumeration consists of three steps: (1) Inducing a fault. For a given fault model $\mathbb{F}$, a one-bit fault is induced to the jth bit in the ith round in the block cipher $\mathbb{B}$ with a simulator. (2) Constructing the equations for $\mathbb{B}$ and $\mathbb{F}$. An automatic tool is customized to generate algebraic equations for both the cipher and the fault. The details can be referred to [3]. (3) Solving the equations. A machine solver is applied to interpret the system of equations and retrieve the secret key.

Since AFA is executed as machine-based automation, all possible key candidates will be eventually checked along the fault propagation path. The utilization of fault leakages is maximized. The automation shows its advantage over traditional manual analysis, such as DFA, especially when the analysis goes into the deeper round (e.g., i decreases), or there are tremendous combinations to analyze (e.g., all (i, j)s). Since the workloads are more on the solver itself, AFA can possibly output the secret key with a single bit fault and limit the injections to be only ONE.

Stage 2: Trigger logic design. HTH trigger signal exploits the temperature sensor present in the system monitor of the Xilinx FPGA [4]. As the application is security sensitive, the system monitor is assumed to be instantiated already for preventive sensors. It produces a 10-bit temperature output in the range $[-273°\text{C}, +230°\text{C}]$. The normal operating temperature is 25°C, i.e., roughly $605 = (1001011101)_b$.

Stage 3: Payload logic design. For clarity, the kth encryption and plaintext are denoted as E_k and P_k. A pair of correct and faulty ciphertext should be collected for the same random plaintext. To fulfil this, the schemed trojan system consists of two payload components: PL(a) serves to inject the bit fault in R_i during E_k; PL(b) temporarily disables the loading operation of new plaintext for E_{k+1}. If the fault is injected to E_k, P_{k+1} will not be loaded into the plaintext registers. Consequently E_{k+1} will continue to encrypt with P_k again, but without injecting fault. This payload allows to fulfill the conditions in Stage 1 with a single execution of HTH. The trojan is activated by the rising edge of the alarm signal from the sensor system—Temperature signal (T_s). Once chip temperature rises above the assigned threshold, T_s will rise from 0 to 1, and activate the trojan by storing a '1' in the flip-flop M. More details about PL(a) and PL(b) can be found in the supplementary file.

Cost-optimization implementation. 6-input LUT is the mainstream look-up-table architecture widely used from the 65 nm Virtex-5 FPGAs to the 20 nm Ultrascale FPGAs. In these devices, the fundamental logic unit, slice, contains four LUTs. A single LUT is able to implement either one Boolean equation with up to 6 inputs, or two Boolean equations with no more than 5 different input signals in total. As illustrated in Figure 1(a), the payload gate f_1(B,C) and f_2(A,B,C) has 2 and 3 inputs, respectively. Occupied LUTs with 3 or less used inputs can be found by searching in XDL. In this stage, the two payload gates can be merged with two arbitrarily occupied LUTs with no more than 3 used inputs, by just modifying the corresponding slice instances on XDL. Since the trojan LUTs are merged in existing logic, the eventual cost is just 2 extra registers.

Experiment. Our design and attack are verified on SASEBO-GII board soldering a 65 nm Virtex-5 FPGA for side channel attack evaluation [5]. And the target cipher is DES where $r = 16$, $m = 32$ if only the left part is considered, and $n = 56$.

In Stage 1, a machine solver, CryptoMiniSAT v2.9.4, runs on a PC with Intel Core i5-4460 @3.2 GHz and 4 GB RAM. One bit fault is injected to the left part of the DES internal state in R_i. To determine i, each round is tested for 400 simulations where j is randomly selected. With only one injection, the solver could not output any satisfiable solution within 24 h for any R_i except $i = 11, 12$. $\phi(K)$ is calculated as the $\log_2$ based number of

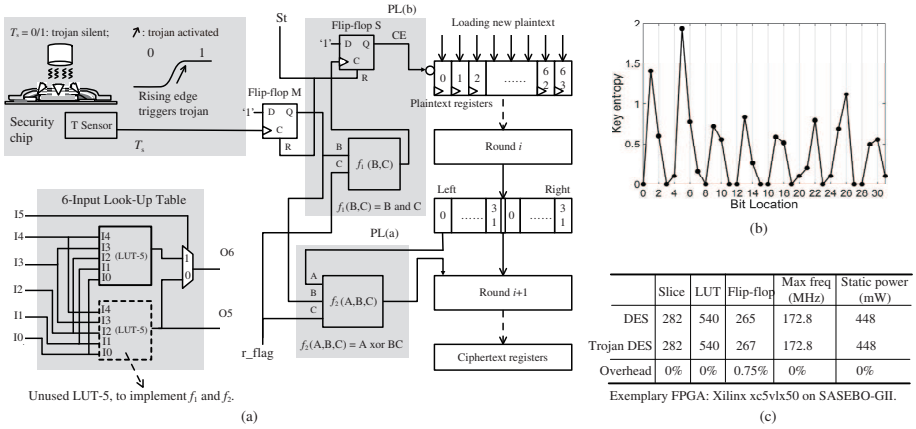

Figure 1 Hardware trojan design based on single-bit AFA. (a) The temperature triggered trojan system; (b) comparison of different bit locations in R_{11} (Tmax=600 s); (c) overhead report of inserted HTH.

total solutions when the solver outputs within 10 min. Since $\phi(K)$ in R_{11} is smaller than that in R_{12} and HTH in the deeper round can evade the detection, therefore $i = 11$. To determine j in R_{11}, every bit is tested 400 times. The statistics are shown in Figure 1(b) where y-axis is $\phi(K)$. Without loss of generality, we can agree that $X_{11,0}$ is one of the optimal locations where $\phi(K) = 0$.

In Stage 2, we directly use the rising edge of 7th bit of temperature output as HTH trigger, i.e., when the temperature is higher than the threshold $640 = (1010000000)_b$ or 42°C. A simple \$5 hair dryer can achieve this. This trigger then activates the payload to perform fault injection. In Stage 3, since the adversary A controls the temperature and knows the ciphertext, he can deduce the pair of correct and faulty ciphertext for E_k and E_{k+1} respectively from the timing of temperature change and the round execution.

Figure 1(c) shows the cost/performance comparison of the original and the trojan inserted DES. Since the single gate in PL(a) is merged with used LUTs from DES logic, the only extra overhead is two flip-flops. The frequency stays unchanged after the trojan insertion. We only estimate the static power as the trojan does not consume dynamic power when it is silent.

Conclusion. A novel low-cost design of hardware trojan is proposed for bit-level fault attacks. AFA is used to search for the optimal location in advance. As for DES implementation, a single bit flip on the most significant bit in the 11th round requires only one injection. The payload logic is carefully designed to collect the pair of correct and faulty ciphertexts required for offline AFA analysis, and to merge the LUTs into existing DES implementation. The trigger logic relies on a temperature sensor. Experiments report a 0.75% additional cost in flip-flops for DES implemented on a 65 nm Virtex-5 FPGA. The proposed design raises the threat from untrusted vendors to industrial IP design, and the attack can even be conducted by adversaries with a household hair dryer.

Acknowledgements This work was supported in part by National Basic Research Program of China (973 Program) (Grant No. 2013CB338004), National Natural Science Foundation of China (Grant Nos. 61173191, 61272491, 61309021, 61472357, 61571063), Zhejiang University Fundamental Research Funds for the Central Universities (Grant No. 2015QNA5005), and Science and Technology on Communication Security Laboratory (Grant No. 9140C110602150C11053).

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 Courtois N T, Ware D, Jackson K. Fault algebraic attacks on inner rounds of DES. In: Proceedings of eSmart European Smart Card Security Conference, Sophia Antipolis, 2010. 22–24
2 Bhunia S, Hsiao M, Banga M, et al. Hardware trojan attacks: threat analysis and countermeasures. Proc IEEE, 2014, 102: 1229–1247
3 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: Constructive Side-Channel Analysis and Secure Design. Berlin: Springer, 2013. 62–79
4 Xilinx. Virtex-5 FPGA system monitor. UG192. Version 1.6, 2008
5 Katashita T, Satoh A, Sugawara T, et al. Development of side-channel attack standard evaluation environment. In: Proceedings of IEEE European Conference on Circuit Theory and Design, Antalya, 2009. 403–408