SCIENCE CHINA
Information Sciences

• Supplementary File •

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*, P.R.China;*
[2] *Science and Technology on Communication Security Laboratory, Chengdu,* 610041*, P.R.China;*
[3] *Institute of North Electronic Equipment, Beijing,* 100191*, P.R.China;*
[4] *Temasek Laboratories, Nanyang Technological University, Singapore,* 637371*,Singapore*

Appendix A Details of the Payload Logic Design in Figure 1

Component PL(a): When the HTH is triggered, a '1' is stored in a flip-flop M which waits for R_i, the target round of fault injection in $\mathbb{B}$. A signal r_flag, derived from state machine, indicates if the current round is R_i or not. The value of (i, j) is determined by Stage 1. When the trojan is triggered, the j^{th} bit of i^{th} round, i.e., $X_{i,j}$, is flipped due to PL(a), in E_k. This is realized by function f_2 as shown in Fig.1(a).

Component PL(b): The role of PL(b) in Fig.1(a) is to disable the loading of plaintext P_{k+1} during E_{k+1}. This can be done by disabling the plaintext loading register using the clock enable (CE) port. The PL(b) is enabled during the i^{th} round of E_k, through the round flag signal r_flag. Upon activation, the flip-flop S in PL(b) stores '1' to disable the plaintext register for next encryption E_{k+1}. Both the activation flip-flop M and PL(b) flip-flop S are reset by the start signal S_t of encryption E_{k+1}. Note that plaintext loading is before the start of encryption computation. The given circuit imposes an activation window on the trigger of trojan. The HTH is triggered after the start of E_k and before the execution of the i^{th} round of E_k. Outside this window, the trojan is harmless. Disabling plaintext loading only after trojan activation is done by function f_1 and flip-flop S.

Signal r_flag**:** The r_flag is a round flag signal which indicates whether the current round is R_i or not. In another word, r_flag is a tag comparison of the current round index with the target round R_i to activate the trojan. r_flag is generated from the unused secondary output (O5) of the LUT5 (marked as in dashed line in Fig.1(a)) that generates the round counter of the original crypto-block on its primary output (O6). Thus the overhead is null.

Signal S_t**:** The S_t is the start signal of the encryption that is assumed as a global input and used to indicate encryption request.

In practice, the injection is controlled by the two payload modules PL(a) and PL(b). Each module includes two inputs B and C.

The signal B comes from the rising edge of the thermal sensor, which plays a role of enabling the trojan in the specified encryption instead of precisely controlling it in the desired round. Therefore, once the adversary wants to launch an attack, he could heat the chip first and keep the warming process for a long while (multiple seconds).

The signal C comes from the round flag and the signal A comes from the specified bit, both of which are pre-known through the location search process. Under the enabling of B, it is the combination of A and C (more precisely, it is the signal r_flag) that guarantees the precise injection at the exact timing (i.e. the calculation of j^{th} bit in the i^{th} round).

However the adversary needs to know which encryption goes wrong when the rising of the thermal trigger. The affordable timing window of a rising trigger is comparable to that of a full encryption. The adversary records the rough timing of the thermal sensor and the exact starting time of each encryption. He compares the two timing information, together observes which plaintexts are used twice, so that he collects the corresponding correct/incorrect pair of ciphertexts, which is enough for the offline analysis.

[*] Corresponding author (email: he.wei@ntu.edu.sg)