Against Transient-Steady Effect Attack using Time Check Blocks

Jinbao Zhang, Ning Wu, Fen Ge, Fang Zhou
College of Electrical and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China.

***Abstract*—Transient-steady effect (TSE) attack is a new type of fault attack which exploits the phenomenon that the output of a combinational circuit keeps a temporal value for a while before it finally switches to the correct value and breaks the key by injecting a clock glitch to capture the temporal value. In order to efficient against TSE attack, we propose our countermeasure based on a time check blocks (TCB). We first make a theoretical analysis for the TSE attack, and point out the transient value is exists as long as the two circuits in operation with different critical paths. Then we present our resist ideas. The proposed idea is to use the TCB to check whether the clock signal is abnormal and change the output of the circuit if there is a clock glitch be detected, this will make the attacker can not get the correct transient value. Furthermore, we point out that our countermeasure can be used to against attacks which based on injecting a clock glitch, such as typical Fault Sensitivity Analysis (FSA). Experiments are carried out to verify our countermeasure, and the results demonstrate that the proposed TCB can successfully detect the abnormal clock, and our countermeasure can resist TSE attacks effectively.**

Keywords—Transient-Steady Effect; Side channel attacks; Countermeasures; AES

I. INTRODUCTION

With the extensive use of cryptographic chips, the security of cryptographic implementation has drawn much attention. In recent years, side-channel analysis or attacks (SCAs), which based on the analysis of side channel information leaked by cryptographic devices, such as timing information, power consumption, electromagnetic radiations, et al. have been a serious threaten to the security of cryptographic implementation. So, it becomes especially necessary to resist against SCAs. Among various SCAs, Differential Fault Analysis (DFA) [1] is one of the well-known attacks. DFA exploits faulty outputs to estimate internal states of cryptographic modules.

In [2], the Fault Sensitivity Analysis (FSA) was proposed. FSA exploits a dependency between the secret information and the fault sensitivity of cryptographic modules. Paper [3] extended FSA to masked AES implementation by combining it with collision attack. In 2012, the Clockwise Collision Fault Sensitivity Analysis (CC-FSA) attack was presented on unmasked AES [4], the authors pointed out if the inputs of two consecutive cycles are identical in an iterative AES implementation, the setup time of the second cycle is extremely short. Soon after that, in 2013, Wang et al. improved the CC-FSA, and broke a masked serial AES S-box implementation [5]. Ren et al. further improved the FSA, they proposed a new attack called Transient-steady effect (TSE) attack [6], which exploits the phenomenon that the output of a combinational circuit keeps a temporal value for a while before it finally switches to the correct value and breaks the key by injecting a clock glitch to capture the temporal value. All the attacks methods mentioned above are carried out at a fixed glitch frequency, through observe the output of cryptographic circuit after injecting the clock glitch, then crack the key.

Several countermeasures have been devised to avoid these attacks which need inject clock glitch into cryptographic circuits. In 2009, the idea of a setup time violation faults detector was presented [7]. The setup time violation faults detector is based on an artificial critical path made of serial delay elements. Later, a similar approach is proposed in [8]. This countermeasure eliminates dependency between the secret data and the fault sensitivity. The enable (EN) signal is essential to hide the fault sensitivities measured by the attackers. Further, Li et al. proposed the method of the EN signal generation in 2012 [9]. Although these countermeasures can effectively resist these attacks which based on the injection of the clock glitch, they all first need to obtain the critical path time consume, and then to design the clock configuration blocks. This will increase the difficulty of the circuit design.

In this paper, we propose a simple clock signal detection mechanism based on the principle analysis for TSE attacks. We use a time check block (TCB) to determine whether there is an injection abnormal clock, and then judge whether there is an attack occur. Once the attack is detected, the correct output of the circuit wouldn't be obtained.

The rest of this paper is organized as follows. In section II, we first analyze the principle of TSE, and then make a brief introduction to advanced encryption standard (AES) [10], last we describe TSE attack. In section III, we discuss the flawed of the countermeasures against TSE attack mentioned in [6], and propose our TCB countermeasure. We utilize this TCB to judge whether there is an attack occur, so as to resist the TSE attack. In section IV, experiments are carried out to verify the proposed countermeasure. Section V concludes this paper.

978-1-5090-6161-7/17/$31.00 ©2017 IEEE

II. TRANSIENT-STEADY EFFECT ATTACK

A. Basic idea for Transient-Steady Effect Attack

A new type of fault attack called Transient-Steady Effect (TSE) attack was proposed in [6]. Unlike some of the previous fault attacks, e.g., DFA [1], TSE attack exploits the phenomenon that the output of a combinational circuit keeps a temporal value for a while before it finally switches to the correct value. By injecting a clock glitch, the attacker can capture the temporal value as a faulty output to recover the key. Fig. 1 describes the principle of Transient-Steady.

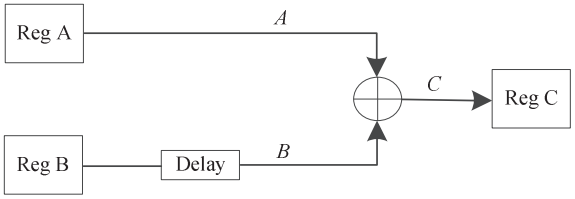

Fig. 1. An example of circuit with different propagation delays

In most standard logic designs, the lengths of data paths in combinational circuits are usually different. As shown in Fig. 1, A and B are the inputs of a combinational circuit. Their propagation delays are denoted as t_A and t_B, and $t_B >> t_A$. The output, denoted as $C = f(A,B)$. For two specific clock cycles, A_1 and B_1 are the inputs in the first cycle. A_2 and B_2 are the inputs in the second cycle. After the rising edge of the second clock, the effects of A_2 and B_2 begin to propagate along the two data paths. When A_2 has impacted all the gates in the circuit after a period of time t ($t_B > t > t_A$), but in this time, the ripple of B_2 has not arrived at the output, so the value of output C will change from $f(A_1,B_1)$ to $f(A_2,B_1)$, if the difference of path delays $d = t_B - t_A$ is large enough, the temporal value $f(A_2,B_1)$ will keep steady for a while. Store the temporal value through injecting a glitch to make the length of the second cycle within the range from t_A to t_B, this is the basic idea for TSE attack.

B. AES

In this paper, AES algorithm is taken as an example to illustrate the fundamentals of TSE attack, as well as the countermeasures.

AES is a symmetric block cipher algorithm that can encrypt and decrypt a 128-bit data with three different key sizes: 128, 192 and 256 bits (respectively called AES-128, AES-192 and AES-256). In this paper, the cryptographic algorithm we focus on is AES-128 because of its popularity and simple description. AES-128 has 10 rounds, and each round is consisted of four operations: SubBytes (SB), ShiftRows (SR), MixColumns (MC), AddRoundKey (ARK), except for the first round and the last round. During the encryption or decryption process, the 16 bytes plaintext is transformed into a 4x4 byte matrix referred to as State.

C. The realization for Transient-Steady Effect Attack

TSE attack introduces the clock glitch in cryptographic circuits to obtain the temporal value mentioned in section *A* to reveal the cipher key. In here, in the interest of conciseness, we illustrate the TSE attack with an unmasked AES circuit.

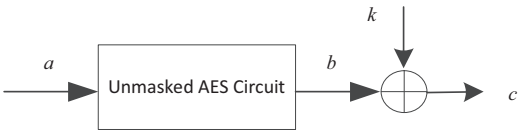

Fig. 2. The data path in the final AES round

As shown in Fig. 2, *a* is the plaintext, *b* is the output of SR for AES tenth round, *k* stands for the key. *c* is the output ciphertext of the cryptographic circuit, so $c = b \oplus k$. We assume the encryption path is longer than the path of key. Actually, many practical AES circuits all meet this assumption.

We let the target unmasked AES circuit computer normally in the first cycle, and inject a clock glitch to create a very short second cycle, as shown in Fig. 3, the delay of encryption path is denoted as t_b, and the delay of key path is denoted as t_k.

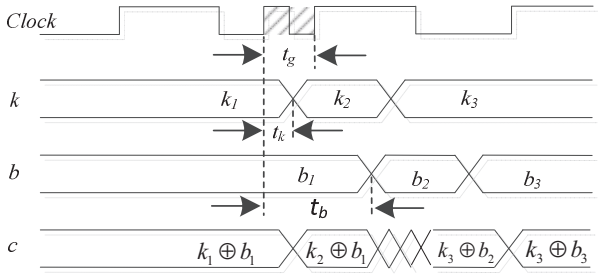

Fig. 3. Sequence diagram of unmasked AES circuit with clock glitch

As shown in Fig. 3, in the first clock, the value of output $c_1 = k_1 \oplus b_1$, after a period of t_k, k_2 arrives at the XOR gate, and the value c_1 switches to a temporal value $c_{1,2} = k_2 \oplus b_1$, and $c_{1,2}$ will stay for the duration time of $t_b - t_k$. With the value of c_1 and $c_{1,2}$, we can get the following relationship:

$$\begin{aligned} c_1 \oplus c_{1,2} &= k_1 \oplus b_1 \oplus k_2 \oplus b_1 \\ &= k_1 \oplus k_2 \\ &= \Delta k_{1,2} \end{aligned} \tag{1}$$

Because c_1 and $c_{1,2}$ are the outputs of the circuit, which the attacker can obtain, so the attacker can break the key.

For masked AES circuit, the analysis process is general similar.