IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 31, 2018, accepted September 3, 2018, date of publication September 17, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2870663

Side-Channel Attack on a Protected RFID Card

RIXIN XU[1,2], LIEHUANG ZHU[1], AN WANG[1,3], XIAOJIANG DU[4],
KIM-KWANG RAYMOND CHOO[5,6], (Senior Member, IEEE),
GUOSHUANG ZHANG[7], AND KEKE GAI[1]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2] State Key Laboratory of Cryptology, Beijing 100878, China
[3] Key Laboratory of Network Assessment Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[4] Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[5] Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[6] Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA
[7] Science and Technology on Information Assurance Laboratory, Beijing 100072, China

Corresponding author: An Wang (wanganl@bit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872040, in part by the National Cryptography Development Fund under Grant MMJJ20170201, in part by the Beijing Natural Science Foundation under Grant 4162053, and in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-17-009.

ABSTRACT Side-channel attack is a known security risk to smart cards, and there have been efforts by smart card manufacturers to incorporate side-channel attack countermeasures. In this paper, we study a widely used smart card that uses the 3DES algorithm. First, a platform is setup to extract the power consumption information from the electromagnetic wave. Based on the findings from the initial analysis, we determine that the card is equipped with a ‘‘head and tail protection’’ mechanism. Second, a chosen-plaintext power analysis with a complexity of 2^{16} is proposed, which is designed to recover the second round key from the power leakage in the third round. Then, a slicing-collision-algebraic attack is presented, which decreases the complexity to 2^6 rapidly. The experiments show that after collecting 2 0000 power traces (in approximately 200 s), only $2^6 \times 8$ key guesses and another 177 searches (about 300 seconds) are sufficient in recovering the 56-bit source keys of DES successfully. In other words, we demonstrate how the security of the 3DES card can be easily compromised, using side-channel attacks. Finally, we recommend that the head and tail protection should extend to the first and last four rounds, at the minimal, in order to be side-channel attack resilience.

INDEX TERMS Cryptoanalysis, smart card attack, side-channel analysis, power analysis.

I. INTRODUCTION

Smart cards are widely used in our society, and generally consist of a microprocessor, I/O interface, and memory. Smart cards allow the processing of data, provision of access control and performing other computing functions. Smart cards implemented with cryptography algorithms have been utilized in the scenarios such as identity authentication, mobile payments, and mobile communications.

Since power analysis attack on cryptographic devices, such as smart cards, was first proposed in 1999 by Kocher *et al.* [1], the potential for side-channel attacks has been widely studied. For example, in 2005, Carluccio *et al.* [2] revealed the security flaw of a contactless smart card, in the sense that they demonstrated how one can conduct an electromagnetic side-channel attack against the smart card. In 2007, Oren and Shamir [3] also demonstrated that remote side-channel attacks can be carried out even when the attacker is several meters away from the target card.

As we all know, smart cards are prevalent smart devices, which are crucial components of the Internet of Things and sensor networks. Several representative papers have studied the security issues about these fields, such as key management [4], [5], access control [6], defending the phishing attack [7] or the hardware security [8]. At the same time, more and more researchers also have paid their attention to side-channel attacks to the smart cards. MiFare Classic, another widely used smart card, was shown to be vulnerable to reverse engineering and man-in-the-middle attack in 2009 [9]. Kasper *et al.* [10] also demonstrated how one can break the security of several commercial RFID cards by exploiting their physical characteristics. Oswald and Paar [11] also compromised the security of the MiFare DESFire card successfully in 2011. Two years later in 2013, the attack of Oswald *et al.* was extended to a remote implementation of one meter [12]. Other more advanced power analysis on contactless card have been presented in the literature, and examples include Correlation Power Analysis (CPA) [13], Differential Clustering Analysis (DCA) [14], Mutual Information Analysis (MIA) [15] and others presented in [16]–[19]. Researchers have also extended side-channel analysis to personal computers (PCs) [20] and smart phones [21], [22].

Power analysis is the most common side-channel attack approach, which establishes a ''correspondence'' between the intermediate value of the encryption process and power fluctuation. Once such a correspondence is established, the attacker can attempt to recover the key of the chip. Attackers often mount the attack from the head or the tail round of a block cipher, because they can obtain the plaintext and the ciphertext, which correspond respectively to the input of the first round and the output of the last round. The known plaintext or ciphertext can help the attacker guess the intermediate values of the head or tail round and then establish the correlation to the power traces. Similarly, when we study the security of a smart card, the ''head or tail'' method is also a popular choice. Clearly, smart card manufacturers have vested interest to design and incorporate countermeasures to mask or hide information (i.e., to avoid information leakage) [23], [24].

In this paper, we studied a widely used 3DES smart card and determine that this card showed the following features: it is a contactless RFID card with some kind of unknown protection, which enables the card to avoid information leakage during the first two rounds and the last two rounds in every DES process. This means that the ''head or tail'' method does not work because there is no correlation between power consumption and intermediate values of the head or tail round. However, DES has been known to be insecure since the 1990's, since the key size is too small to withstand attacks from existing computing devices. However, 3DES is still widely used, partly due to its resilience to brute-force attacks. Some papers also have noticed this kind of countermeasures which can get rid of the power leakage from the first and last round. Reparaz and Gierlichs [25] proposed a method of combining DPA and differential cryptanalysis to recover the key, using the power leakage from the 3rd round of Feistel cipher. But this work only showed the simulation experiments on an 8-bit microcontroller, while our target is a widely used card. At the same time, [25] needs 2^{28} hypothesis which is too more than the hundreds hypothesis of our work. With the same SNR, the failure rate of our work is only 25% of our work.

In this paper, we revealed that the 3DES smart card was insecure by showing how to successfully recover the 3DES key. Our contributions are summarized as follows:

- Unlike empirical power analysis targeting the head or tail round, we present a new framework to directly target the third round. This framework can mitigate the avalanche effect of the first two rounds by chosen-plaintexts, and yet obtain the status of the third round. This allows us to mount the attack on the third round directly, just like the first round. It is impractical to recover the key using conventional power analysis, but using this new framework, we only need to search a space of 2^{16} to obtain the key.
- We design a bitwise-absolute-correlation distinguisher. It is similar to the Hamming distance leakage model, but we only need to guess the final status of the register if the initial status is unknown but static. It reduces the required searching workload significantly, from 2^{16} to 2^{6}.
- We give a combination solution consisting of power analysis, collision attack, and algebraic attack. Based on the ''2^6'' distinguisher as mentioned above, this combined solution enables us to locate the unknown 56-bit key with fewer attempts.
- Our method proves that it still works even if the protection covers the first and last three rounds.

In the next section, we present the relevant preliminaries, including experiment setup. Section III explains how we evade the avalanche effect and compromise the first two rounds' protection. Section IV describes our new distinguisher and the combined attack solution. Section V presents a potential countermeasure to our attacks. The last section concludes this paper.

II. PRE-ANALYSIS OF A PROTECTED 3DES CARD

A. NOTATIONS

DES contains 16 rounds. For the i-th ($1 \leq i \leq 16$) round, we use L^{i-1} and R^{i-1} to denote the two 32-bit halves of its input. Accordingly, L^i and R^i represent both the output of i-th round and the input of $(i+1)$-th round. The i-th round key is denoted as K^i. We use the subscript to denote one or a few specific bits of an intermediate value, for example, K^2_{44} indicates the 44th bit of the 2nd round key. For several bits of a value, we use the subscript with a pair of angle brackets to represent them as a whole. For instance, $K^2_{32,44,46\sim48}$ are 5 bits total and this set can be denoted as $K^2_{<5>}$. This kind of notations is suitable for the cases that we do not care about which bits they exactly are but the amount of them.

Each round can be concluded as $L^i = R^{i-1}$ and $R^i = L^{i-1} \oplus F\left(R^{i-1}, K^i\right)$. The $F(\cdot)$ brings the ''confusion'' and ''diffusion'' effect, which are also referred as the ''avalanche effect,'' to the encryption. DES exhibits a strong avalanche effect. In order to see this, we can exemplify this by trying to guess a bit of R^2 with an unknown master key. Every bit of R^2 is identified by $R^1_{<6>}$ and $K^2_{<6>}$, while every bit of the former is also identified by $R^0_{<6>}$ and the corresponding $K^1_{<6>}$. This shows that if we want to identify even only one bit of the R^2, we should gather all the values of these 42 bits ($K^2_{<6>}$ and $K^1_{<36>}$).

The avalanche effect also constrains the application of the power analysis. In general, it is inevitable to traverse all the hypothetical values of a master key chunk when an adversary mounts a power analysis, no matter it is CPA, MIA, DCA and so on. A method will be treated as infeasible if the entropy of the chunk is too large (larger than 2^{40}). As we see previously, because of the avalanche effect, trying to attack the first or the

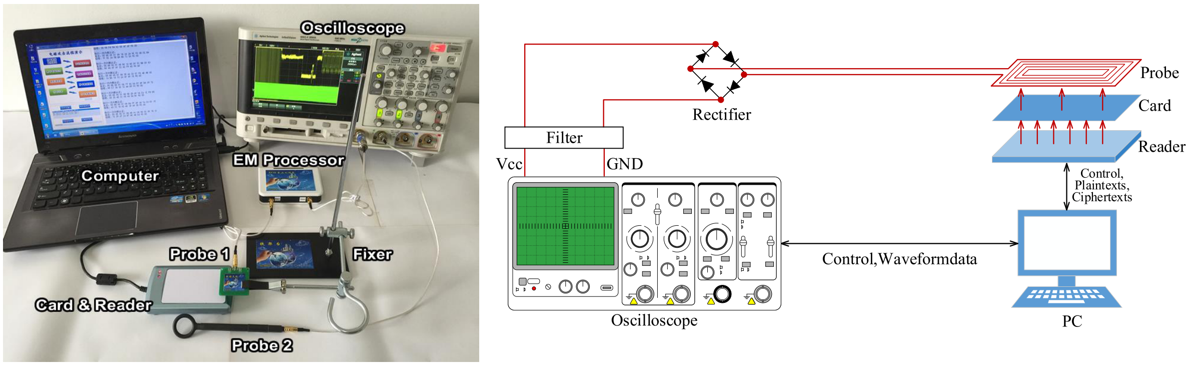


FIGURE 1. An overview of our platform.

last round has been the only choice to an adversary. Based on this, the manufacturers also put the effort on protecting these two rounds. We can refer this as the "head and tail protection."

B. EXPERIMENT SETUP AND POWER EXTRACTION

Our platform and its illustration are shown in Fig. 1. We make a 4-loop 40mm × 40mm antenna as the probe, and connect it to the oscilloscope. The reader provides AC power to the RFID card by emitting 13.56MHz electromagnetic waves, which are firstly consumed by the card, and then the rest can be sensed by the probe.

The encryption flow is actually a sequence of instructions. The power consumption varies during the flow as the Hamming weight of the intermediate values stored by register varies. For every encryption process, what the probe senses can be shown by the oscilloscope as a "power trace." This trace is also 13.56MHz, but it does not exactly equal the AC power. This is because the power carried by a wave is fixed, and when an operation consumes more power, the probe senses less accordingly. Otherwise, the probe will sense more. This phenomenon can be represented in a form of that, if an operation consumes more power, the amplitude of the power trace will be less. All this implies that the peaks and troughs of the traces reflect the power consumption the encryption process. To make full use of all the information, the original EM traces collected by the probe will be converted by a rectifier circuit to an equivalent form of power signal before they are transferred to the oscilloscope.

C. LOCATING THE LEAKAGE ON THE TRACE

Before or after the encryption, a smart card usually does some other jobs, such as receiving the plaintext from the reader, transmitting the data between CPU and encryption co-processor, and sending the ciphertext to the reader. This means that we need to exactly locate the encryption moment during the card's entire workflow. According to the correlation between the power consumption and the data being processed, we divided all 10000 plaintexts and ciphertexts into 8 bytes. Then we computed the correlation coefficients between the Hamming weight of all these bytes and the 10000 power traces, and a number of peaks were accordingly obtained. We got a conclusion that all the instants corresponding to the peaks are exactly the moments that the card is processing the plaintexts or the ciphertexts. To be specific, the interval of 200∼1600 is the time of receiving the plaintexts from the reader and the interval of 7300∼8700 is the time of transmitting ciphertexts back to the reader. We roughly took the interval from the plaintext curves' last peak (5750) to the ciphertext curves' first peak (6299) as the encryption procedure. This result of this locating experiment is shown in Fig. 2.

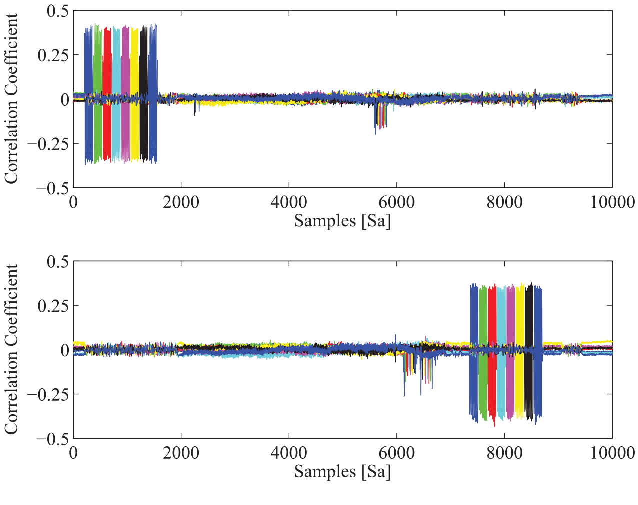


FIGURE 2. Locate the encryption interval.

After locating the encryption interval on the power traces, we begin to calculate the intermediate values of N plaintexts ($N = 10000$ in the actual experiment) encrypted by an identical known key. During the encryption, the register without protection will store 17 values. They are $L^i \parallel R^i$, where $0 \leq i \leq 16$. The N times encryption can form 17 lists. Every list contains N elements, they are $L^i(0) \parallel R^i(0), L^i(1) \parallel R^i(1), \cdots, L^i(N-1) \parallel R^i(N-1)$, where $0 \leq i \leq 16$. We also can denote the list as $HW_{list}(i)$ by calculating the Hamming weight of each element. At the same time, this no-protection register will transient 16 times, corresponding the 16 rounds of Encryption. The status of the register transient from $L^i \parallel R^i$ to $L^{(i+1)} \parallel R^{(i+1)}$, where $0 \leq i \leq 15$. There are 16 lists like this, every list contains N elements. Every list can also be denoted as HD_{list}.

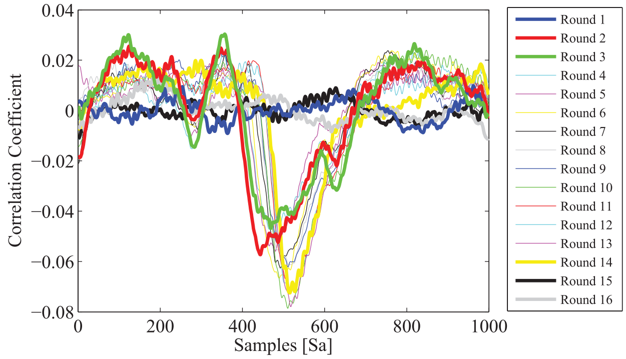


FIGURE 3. **16 rounds' power leakage of the card.**

At first, we failed to find out any leakage when we calculated the correlation between the power traces and all the 17 HW_{list}s. However, we came to success when we tried the 16 HD_{list}s. We observed that there are significant leakages when the card is working from the 2nd round to 14th round, but there is no leakage when it is working at the first round and last 2 rounds. We resampled the 500 points (5750∼6299) of Fig. 2. The result is shown as 250∼749 of Fig. 3.

D. THE CHALLENGE THAT THE MANUFACTURER BRINGS TO US

Now we can give a summary about the card's leakage model:

- The leakage of the card matches the Hamming distance model. The power consumption caused by the transient is obviously correlated to the Hamming distance between $L^{i-1} \parallel R^{i-1}$ and $L^i \parallel R^i$.
- The manufacturer may deploy some mask countermeasures on the first 2 rounds and last 2 rounds. Apparently, there is no power leakage during the 1st round. But why is there significant power leakage during the 2nd round?

As $\mathrm{HD}(R^1, R^2)$ equals to $\mathrm{HD}(L^2, L^3)$, now we use m to denote the mask. When we calculated the correlation coefficient between $HD_{list}(2)$ and the power traces, $HD_{list}(2)$ can be actually represented by $\mathrm{HD}(L^1 \oplus m, L^2) + \mathrm{HD}(L^2, L^3)$ too. $\mathrm{HD}(L^1 \oplus m, L^2)$ is random as the register does not store L^1 but may store $L^1 \oplus m$. Therefore, when we calculated the correlation between $HD_{list}(2)$ and power traces, what we get is actually the correlation between the power traces and the information of $\mathrm{HD}(L^2, L^3)$ with noise. Of course we can treat $\mathrm{HD}(L^1 \oplus m, L^2)$ as noise because it is a random value. If there is no protection deployed on the 2nd round, the 1st round will demonstrate significant leakage as the 2nd round does. The same reason also applies to the last 2 rounds.

III. PENETRATING THE PROTECTION OF THE FIRST 2 ROUNDS BY CHOSEN-PLAINTEXTS

This section fisrt shows the basic idea of [25]. It offers a method of evading the avalanche effects of the first 2 rounds in a way of chosen-plaintexts, so the attackers can "climb over" the first 2 rounds and attack the 3rd round directly. Then an improved method is proposed to reduce the unknown space from 2^{16} to 2^{10}, and the experiment proved its feasibility.

A. BASIC IDEA

It is infeasible to guess the transient of a specific latch from a bit of R^1 to the same bit of R^2 when the 2nd round ends. This is because we can not guess any bit of R^2, which is the result of the avalanche effect. But we have known that the avalanche effect of the 1st round is caused by both R^0 and K^1, and K^1 is a fixed value. So, how about each R^0 of the N plaintexts also equal to a fixed value?

Then the F-function of the 1st round will output a fixed value too, and this also means the avalanche effect of the 1st round lost its functionality. Therefore, the basic idea of [25] to evade the avalanche effect of the 1st round encryption by chosen-plaintexts is that these N specific plaintexts all share a common characteristic, which makes that when they are processed after IP. They make:

1. Every L^0 is a random value at the range $[0, 2^{32} - 1]$;

2. Every R^0 equals to a 32-bit constant value. For simplicity, we set each bit of this value as 0 and denote it as $\{0\}^{32}$. After the 1st round, the $L^1 \parallel R^1$ equals to $\{0\}^{32} \parallel \left(L^0 \oplus F\left(\{0\}^{32}, K^1\right)\right)$.

This leads that the F-function's output is unknow but static. We use Ω to denote this value, then the output is eventually transformed to $\{0\}^{32} \parallel \left(L^0 \oplus \Omega\right)$.

It is not only the 1st round's output but also the 2nd round's input. Repeat the above derivations, then the input of the 2nd round will be denoted as $\left(L^0 \oplus \Omega\right) \parallel \left(\{0\}^{32} \oplus F\left(L^0 \oplus \Omega, K^2\right)\right)$.

We focus on the transients between R^1 and R^2, as they are equal to the 3rd round's transients from $L^2 = L^0 \oplus \Omega$ to $L^3 = F\left(L^0 \oplus \Omega, K^2\right)$.

The basic idea of [25] can also be illustrated as Fig. 4.

As Fig. 4 shows, we can focus on the 4-bit transition between R^1 and R^2. We can illustrate this by using the 7th S-box of the 2nd round:

1. The 6-bit input of the 7th S-box equals $R^1_{24\sim29} \oplus K^2_{37\sim42}$, and $R^1_{24\sim29}$ equals $L^0_{24\sim29} \oplus \Omega_{24\sim29}$. To hypothesize the 4-bit output, an attacker needs to guess $\Omega_{<6>}$ and $K^2_{<6>}$. For every guess, the attack can get the 4-bit hypothetical output of the F function of the 2nd round, they are $R^2_{7,12,22,32}$;

2. When the register is about to store $R^2_{7,12,22,32}$, 4 latches will transit from $R^1_{7,12,22,32}$ to $R^2_{7,12,22,32}$. Therefore, the attacker should guess another 4-bit ($\Omega_{7,12,22,32}$) to hypothesize $R^1_{7,12,22,32}$.

This means that, in order to hypothesize $R^1_{<4>}$ and $R^2_{<4>}$ which are stored in the same 4 latches, an attacker only needs

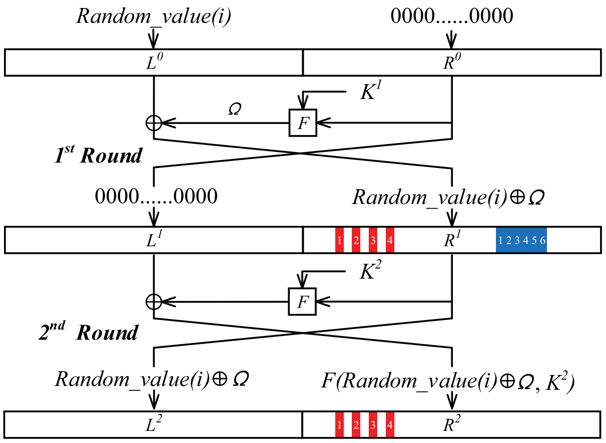


FIGURE 4. Basic idea of evading the avalanche effect ($1 \leq i \leq N$).

to guess 16 bits. This is a feasible solution to overcome the difficulties caused by the avalanche effect.

So, we can mount a CPA by using the Hamming distance between $(R^1_7, R^1_{12}, R^1_{22}, R^1_{32})$ and $(R^2_7, R^2_{12}, R^2_{22}, R^2_{32})$.

B. AN INTUITIVE ATTACK PROCEDURE

Based on the basic idea, we give the generic steps of this attack are as follows:

1) Generate N plaintexts meeting the following conditions when they are processed after the IP permutation:
   a) L^0 is a random value at the range $\left[0, 2^{32}-1\right]$;
   b) R^0 equals to $\{0\}^{32}$.
2) Encrypt these N plaintexts, and collect N corresponding power traces.
3) For a specific S-box of the 2nd round, traverse all the possible values of $\beta = \Omega_{<4>} \parallel \Omega_{<6>} \parallel K^2_{<6>}$. For each possible β, we perform the following steps:
   a) For the i-th plaintext $(1 \leq i \leq N)$, we use $L^0_{<6>}$, $K^2_{<6>}$ and $\Omega_{<6>}$ to hypothesize the $R^2_{<4>}$, use $L^0_{<4>}$ and $\Omega_{<4>}$ to hypothesize the $R^1_{<4>}$. After hypothesizing these N pairs of $R^1_{<4>}$ and $R^2_{<4>}$, we can eventually get a vector containing N elements. These N elements are Hamming distance of every pair of $R^1_{<4>}$ and $R^2_{<4>}$
   b) We calculated the correlation coefficient between this hypothetical vector and the power traces, and record the result as ρ_β.
4) Find the most significant peak on the ρ_β curve, then the corresponding β is the correct guess.

Finally, an attacker can repeat step 1~4 and other seven $K^2_{<6>}$s will be recovered. Then the attacker will get all the 48-bit K^2.

Attacking the 3rd round directly is infeasible by empirical methods, but this method makes this kind of attack practical. It uses the chosen-plaintexts to evade the avalanche effect of first 2 rounds successfully. But this method still exposes two drawbacks:

1) It is inefficient, as the experiment proved that it took about 22 hours to traverse every possible β belongs to a specific S-box;
2) The 2^{16} possibilities of β may result into too many "ghost peaks."

C. IMPROVED ATTACK BASED ON CONJUGATE GUESS

Now we take the 7th S-box for example again. Its 6-bit input values are $R^1_{24\sim29} \oplus K^2_{24\sim29}$, and $R^1_{24\sim29}$ are $L^0_{24\sim29} \oplus \Omega_{24\sim29}$.

Accordingly, the 7th S-box's 6-bit input values are actually $L^0_{24\sim29} \oplus \Omega_{24\sim29} \oplus K^2_{24\sim29}$.

This leads to a conclusion that $R^1_{24\sim29}$ are not determined by either $\Omega_{24\sim29}$ or $K^2_{24\sim29}$, but are determined by the result of $\Omega_{24\sim29} \oplus K^2_{24\sim29}$ in a bitwise sequence. We use $\alpha_{<6>}$ to denote the result of $\Omega_{24\sim29} \oplus K^2_{24\sim29}$ in a bitwise sequence, and the 7th S-box's 6 bits change into $L^0_{24\sim29} \oplus \alpha_{24\sim29}$.

This means that we only need to guess the 6-bit $\alpha_{24\sim29}$, instead of guessing the 12-bit $\Omega_{24\sim29}$ and $K^2_{24\sim29}$. Therefore, we shrink the β's space from 16 bits ($\beta = \Omega_{<4>} \parallel \Omega_{<6>} \parallel K^2_{<6>}$) to 10 bits ($\beta = \Omega_{<4>} \parallel \alpha_{<6>}$)

When we finish all the 8 S-boxes, we can get a 48-bit α and a 32-bit Ω. We expand the Ω to a 48-bit value because it will be processed by the E expansion before it enters to the S-box, and every bit of Ω will be rearranged in a new sequence to compose this new value. We denote this 48-bit value as Ω^E. Finally, K^2 can be easily gotten as $K^2_i = \alpha_i \oplus \Omega^E_i$, where $1 \leq i \leq 48$.

We give the generic steps of this attack as follows:

1) Generate N plaintexts meet the following conditions when they are processed after IP:
   a) L^0 is a random value at the range $\left[0, 2^{32}-1\right]$;
   b) R^0 equals to $\{0\}^{32}$.
2) Collect N power traces while the card is encrypting these N plaintexts with an unknown key.
3) For a specific S-box of the 2nd round, traverse all the possible values of $\beta = \Omega_{<4>} \parallel \alpha_{<6>}$. For each possible β, we perform the following steps $(1 \leq i \leq N)$:
   a) For the i-th plaintext $(1 \leq i \leq N)$, we use $L^0_{<6>}$ and $\alpha_{<6>}$ to hypothesize the $R^2_{<4>}$, and use $L^0_{<4>}$ and $\Omega_{<4>}$ to hypothesize the $R^1_{<4>}$. After hypothesizing these N pairs of $R^1_{<4>}$ and $R^2_{<4>}$, we can eventually get a vector containing N elements. These N elements are Hamming distance of every pair of $R^1_{<4>}$ and $R^2_{<4>}$;
   b) We calculated the correlation coefficient between this hypothetical vector and the power traces, and record the result as $\rho\,(\beta)$.
4) Find the most significant peak on the $\rho\,(\beta)$ curve, then the corresponding β is the correct guess. Therefore, $\alpha_{<6>}$ and $\Omega_{<4>}$ can be obtained too.
5) Repeat the steps 1~4 until we get the rest $\alpha_{<42>}$ and $\Omega_{<24>}$.
6) Expand the Ω as it is processed by E expansion to get the 48-bit Ω^E.
7) Calculate the 48-bit $K^2 = \alpha \oplus \Omega^E$.

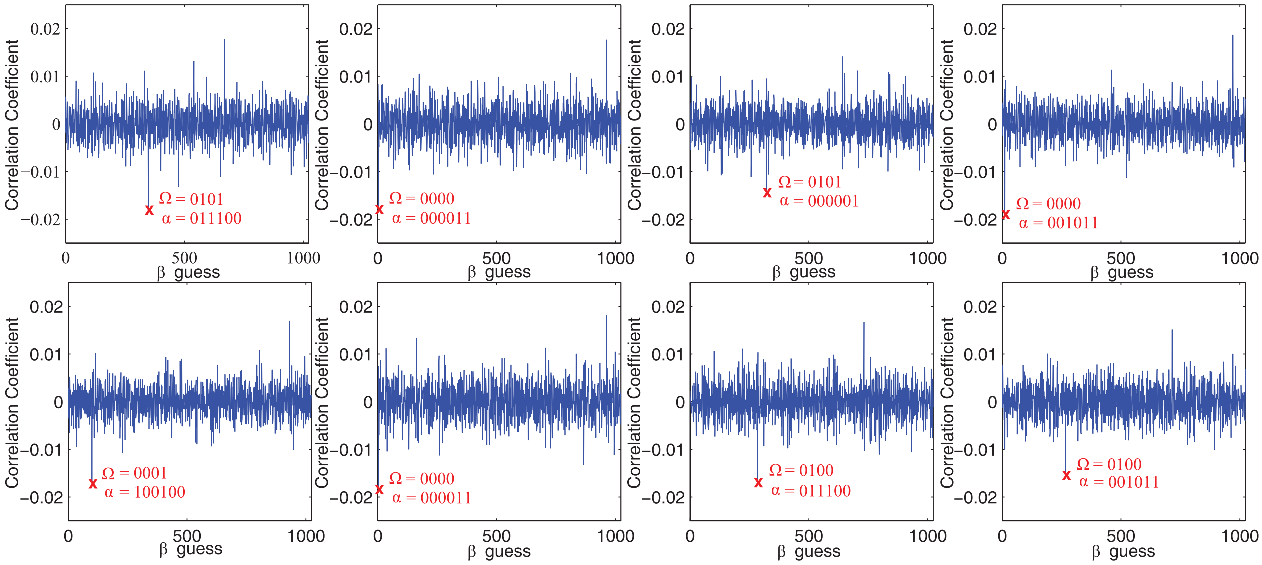


FIGURE 5. The 48-bit α and 32-bit Ω. The 32-bit Ω is presented by the sequence of the corresponding S-box.

This means that we just need to try 2^{10} times at most then we can recover the 48-bit K^2.

D. ATTACK EXPERIMENT

We collected the power traces of 18000 plaintexts' encryptions, and used MATLAB to implement this attack. Then the corresponding β were recovered, and the corresponding α and Ω are obtained too. Next, we calculate $K_i^2 = \alpha_i \oplus \Omega_i^E$, where $1 \le i \le 48$, to recover the 48-bit K^2:

$$011100, 000011, 000011, 101100,$$
$$010100, 000010, 001100, 000111.$$

At last, we searched the rest 8 bits and recovered the 56-bit source key $\widetilde{K}$ successfully. We used 18 seconds to collect enough traces. Every S-box cost us about 162 seconds on average to get the correct corresponding $K^2_{<6>}$, then we cost about 21 minutes ($8 \times 162 = 1296$) totally.

IV. A SLICING-COLLISION-ALGEBRAIC ATTACK BASED ON HAMMING DISTANCE LEAKAGE MODEL

We do not think that the complexity of 2^{10} in Sec. III-C is efficient enough. Besides this, there are still too many disturbance choices of that method. Because of this, we propose a new distinguisher. It can recover the correct $\alpha_{<6>}$ without $\Omega_{<4>}$, this makes the complexity of key search shrink from 2^{10} to 2^6.

A. BITWISE-ABSOLUTE-CORRELATION DISTINGUISHER IN A SLICING PERSPECTIVE

For each guess of $\beta = \Omega_{<4>} \parallel \alpha_{<6>}$, when we calculate $\mathrm{HD}\big(R^1_{<4>}, R^2_{<4>}\big)(i)$, we actually calculate the $\mathrm{HD}\big(L^0_{<4>} \oplus \Omega_{<4>}, R^2_{<4>}\big)(i)$, and this equals to $\mathrm{HW}\big(L^0_{<4>} \oplus \Omega_{<4>} \oplus R^2_{<4>}\big)(i)$, where $1 \le i \le N$.

Then we know that before the register transits to $R^2_{<4>}$, it stores $L^0_{<4>} \oplus \Omega_{<4>}$. We don't know this value because we don't know $\Omega_{<4>}$. As the method mentioned at III-C, we will face a workload of 2^{10}, as the $\Omega_{<4>}$ is included. Since $\Omega_{<4>}$ is a constant value, then can we ignore it, or just guess 2^6 of choices?

The results of $L^0_{<4>}(i) \oplus \Omega_{<4>} \oplus R^2_{<4>}(i)$ can be divided into 4 lists. Take $R^2_{7,12,22,32}$ as an example again, these 4 lists are $L^0_7(i) \oplus \Omega_7 \oplus R^2_7(i)$, $L^0_{12}(i) \oplus \Omega_{12} \oplus R^2_{12}(i)$, $L^0_{22}(i) \oplus \Omega_{22} \oplus R^2_{22}(i)$ and $L^0_{32}(i) \oplus \Omega_{32} \oplus R^2_{32}(i)$, where $1 \le i \le N$. Every element of these 4 lists contains only 1 bit.

Let us focus on these 4 lists again, if we correctly guess the $\alpha_{<6>}$ and $\Omega_{<4>}$, and if we calculate the correlation coefficients between these 4 lists and the power traces, then all of ρ_7, ρ_{12}, ρ_{22} and ρ_{32} will show a significant peak. But what will happen if we guess the correct $\alpha_{<6>}$ but a wrong $\Omega_{<4>}$? If we just correctly guess the $\alpha_{<6>}$ but guess a wrong $\Omega_{<4>}$, these 4 lists will turn to be Table 1.

TABLE 1. 4 lists if we incorrectly guessed the 12th bit of $\Omega_{<4>}$.

7	12	22	32
$R^2_7(1) \oplus L^0_7(1) \oplus \Omega_7$	$R^2_{12}(1) \oplus L^0_{12}(1) \oplus \overline{\Omega}_{12}$	$\ldots$	$\ldots$
$R^2_7(2) \oplus L^0_7(2) \oplus \Omega_7$	$R^2_{12}(2) \oplus L^0_{12}(2) \oplus \overline{\Omega}_{12}$	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
$R^2_7(N) \oplus L^0_7(N) \oplus \Omega_7$	$R^2_{12}(N) \oplus L^0_{12}(N) \oplus \overline{\Omega}_{12}$	$\ldots$	$\ldots$

This means that all the one-bit values of the 12-column will flip from the correct value of 0 to 1 or from 1 to 0, and this leads to the result that the correlation between 12-column and power traces will be $-\rho(12)$. The absolute of the result is still correct. This tells us that all we have to do is just to get the right $\alpha_{<6>}$. That is because, even if we guess a wrong $\Omega_{<4>}$, the correlation peak will be as same as the right guess, but towards an opposite direction. This allows us to guess only 6-bit $\alpha_{<6>}$ and neglect the 4-bit $\Omega_{<4>}$. Besides this, we do not need to care about the values before the transition although we use the Hamming distance leakage model.

"Slice" the result of $\mathrm{HD}\left(R^2_{<4>}, L^0_{<4>} \oplus \Omega_{<4>}\right)(i)$ into 4 columns, so we can use the information in a new perspective. Then, this enables us to get rid of guessing the $\Omega_{<4>}$, because it does not affect the result of the attacks.

This new distinguisher is illustrated as follows:

1) Generate N plaintexts meet the following conditions when they are processed after IP:
    a) L^0 is a random value at the range $\left[0, 2^{32}-1\right]$;
    b) R^0 equals to $\{0\}^{32}$.
2) Collect N power traces while the card is encrypting these N plaintexts with an unknown key.
3) For a specific S-box of the 2nd round, traverse all the possible values of $\beta = \alpha_{<6>}$. For each possible β, we perform the following steps ($1 \le i \le N$):
    a) For the i-th plaintext, we use $L^0_{<6>}$ and $\alpha_{<6>}$ to calculate the corresponding 4 bits of $R^2_{<4>}(i)$, which can be denoted as $R^2_a, R^2_b, R^2_c, R^2_d$. Now we get the pair of $\left(L^0_{<4>}, R^2_{<4>}\right)(i)$.
    b) After this, we get a list containing N elements which are $\mathrm{HD}\left(L^0_{<4>}, R^2_{<4>}\right)(i)$.
    c) Divide this "HD_list" into 4 sub-lists, they are $L^0_a \oplus R^2_a$, $L^0_b \oplus R^2_b$, $L^0_c \oplus R^2_c$ and $L^0_d \oplus R^2_d$.
    d) We calculated four correlation coefficients between the four hypothesis lists and the power traces, they are ρ_a, ρ_b, ρ_c and ρ_d. Then we calculate $\rho(\beta) = |\rho_a| + |\rho_b| + |\rho_c| + |\rho_d|$.
4) Find the $\rho(\beta)$ with the most significant peak. The corresponding β is the correct guess.
5) Repeat the steps 1~4 until we get the rest $\alpha_{<42>}$.

We collected 20000 power traces by the specific plaintexts. The experiment proves this method's high efficiency and the ability to tolerate disturbance. We cost about 36 seconds on average to recover every 6-bit α, and there are only 63 disturbance choices. The result was generated by MATLAB and can be shown in Fig. 6.

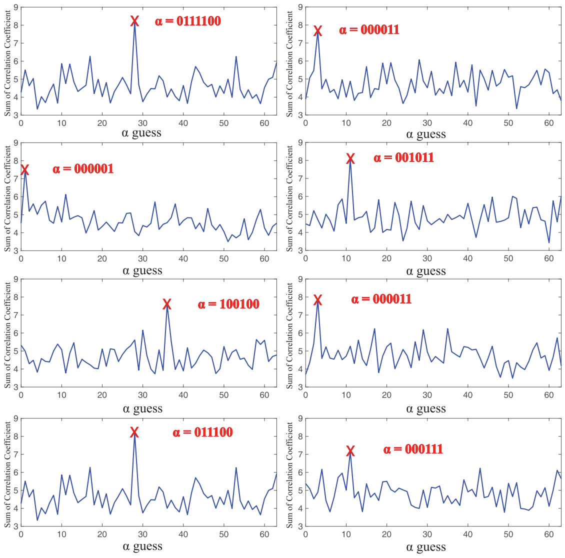


FIGURE 6. Recover the 48-bit α.

B. COLLISION ATTACK

The collision attack is trying to extract some equivalence relations between some bits of the 56-bit source key, in order to reduce the unknown bits. As the characteristic of the E-expansion, we will use it to reduce 16 unknown source key bits.

The 32-bit Ω turns to be a 48-bit Ω^E, as illustrated in Fig. 7. For example, after the E-expansion, Ω_4 and Ω_5 will be arranged into new places of Ω^E:

$$\Omega^E_5 = \Omega^E_7 = \Omega_4, \quad \Omega^E_6 = \Omega^E_8 = \Omega_5.$$

Then we can give two equations:

$$\alpha_5 \oplus K^2_5 = \alpha_7 \oplus K^2_7, \quad \alpha_6 \oplus K^2_6 = \alpha_8 \oplus K^2_8.$$

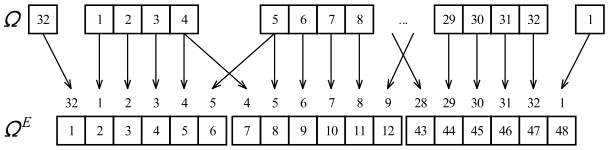


FIGURE 7. The 32-bit Ω after E-expansion.

This means that if we have α, we just need to guess one of K^2_5 and K^2_7, then we will get the other. This same thing also can be applied to K^2_6 and K^2_8.

By the collision brought by E-expansion, we can further reduce unknown 16-bit key guesses.

C. ALGEBRAIC ATTACK

We use $\widetilde{K}$ to denote the 56-bit source key. After the key generation, the K^1's and K^2's 48 bits correspond to the $\widetilde{K}$'s:

$$K^1 : \left[\widetilde{K}_9, \widetilde{K}_{45}, \widetilde{K}_{30}, \widetilde{K}_{53}, \widetilde{K}_{43}, \widetilde{K}_{15}, \ldots, \widetilde{K}_{28}\right],$$
$$K^2 : \left[\widetilde{K}_2, \widetilde{K}_{38}, \widetilde{K}_{23}, \widetilde{K}_{46}, \widetilde{K}_{36}, \widetilde{K}_8, \ldots, \widetilde{K}_{21}\right].$$

We take α_6 for an example, α_6's value is known and: $\alpha_6 = K^2_6 \oplus \Omega^E_6 = \widetilde{K}_8 \oplus \Omega_5$.

As Ω_5 is the output of the 8th S-box in the 1st round, so we can give the equations about α_6 and 6 bits of source key, they are:

$$\alpha_6 = \widetilde{K}_8 \oplus \Omega_5 = \widetilde{K}_8 \oplus Sbox\left(R^0_{<6>} \oplus \widetilde{K}_{<6>}\right)_1 \oplus L^0_5,$$
$$R^0_{<6>} = R^0_{1,28\sim32},$$
$$\widetilde{K}_{<6>} = \widetilde{K}_{40,13,12,55,49,28}.$$

$Sbox(\cdot)_1$ denotes the first bit of the output of the S-box. Accordingly, we can set up 48 equations to represent all 48-bit α.

With the help of the collision attack, we've known that $\alpha_6 = K^2_6 \oplus K^2_8 \oplus \alpha_8$ and can set up another 16 equations like this. Consequently, we will have 64 equations to get 56-bit unknown $\widetilde{K}$. We use the mini-SAT to find the possible

TABLE 2. Evaluation and comparison of all the 4 methods.

	Classic CPA	Sec. III-B	Sec. III-C	Sec. IV
Unknown space	2^{36}	2^{16}	2^{10}	2^{6}
Power traces	Infeasible	18000	18000	20000
Online time (seconds)	Infeasible	180	180	200
Offline time (seconds)	Infeasible	>10000 × 8	$162 \times 8 = 1296$	$36 \times 8 + 3 = 291$
Further procession (seconds)	Infeasible	256 searches	One XOR and 256 searches	177 searches

solutions, and finally, we get 177 results. This means that this was a successful attack because we can just search these 177 possible results to identify the 56-bit source key $\widetilde{K}$.

D. COMPARISON OF ALL ABOVE METHODS

We give the comparison of all the above methods. The first method is classic CPA, but it is infeasible because there are too many unknown bits (Sec. II-A). The other three methods have been mentioned at Sec. III-B, Sec. III-C and this section respectively. We need to collect about 18000 power traces to gather enough information to recover the source key, and the third method only uses 162s around to recover the corresponding 6-bit α of an S-box. And the last row shows the new distinguisher's efficiency, it only needs about 36s to complete the same work of the third method. Our experiment is made on a PC with an Intel i5-650 processor and 12GB memory, and the evaluation is shown in Table 2.

E. RECOVERING THE 168-bit KEY OF 3DES

The target smart card adopts 3DES with a 168-bit key. When we are dealing with the 3DES, we use correlation power analysis again. For each of the 177 possibilities, to every one of the N plaintexts, we calculate the hypothetical values of all the output of the 16th round of 1st DES. We calculated the correlation between all these values and the N power traces, then the hypothetical value with the most significant peak is the correct one. After this, we have recovered all the 56 bits of the first key. Now we can generate the input of the 2nd as we want and recover the 2nd key by repeating this experiment. We will recover the 168-bit key of 3DES eventually.

V. RESISTING OUR ATTACK

To resist our attack, we should make sure how many rounds our attack can penetrate. Now, we try to attack the 4th round by using the HD(L^3, L^4) information leakage. This means we should try to establish the correlation between the power consumption and HD(R^2, R^3).

We give an example by using 4-bit leakage of the 4th round to show the steps instead of giving the generic steps as follows.

1) Generate N plaintexts, for each plaintext, it meets the following conditions after it is processed after the initial permutation:
   a) All the 6 bits of $L^0_{24\sim29}$ equal 0. For each bit of the rest $L^0_{<26>}$, it is random and equals to either 0 or 1. We denote all these 26-bit long random values as $L^0_{<26>}(1), \ldots, L^0_{<26>}(N)$;
   b) R^0 equals to $\{0\}^{32}$;
2) Collect N power traces while the card is encrypting these N plaintexts with an unknown key;
3) We use Ω to denote the result of $F\left(R^0, K^1\right)$, then:
   a) L^1 equals to $\{0\}^{32}$;
   b) The 32-bit of R^1 are the results of bitwise XORed of L^0 and Ω, and L^0 is known;
4) Then we will get the input and output of 3rd round:
   a) L^2 is the copy of R^1;
   b) For $R^2_{7,12,22,32}$, their values are the result of $L^1 \oplus F(L^0_{24\sim29} \oplus \Omega_{24\sim29} \oplus K^2_{37\sim42})$;
   c) For $R^3_{7,12,22,32}$, we can get them from L^2, $R^2_{24\sim29}$ and $K^3_{37\sim42}$.

According to the 4-bit Hamming distance leakage, we can mount CPA as mentioned in the above sections, and recover the corresponding key.

However, our chosen-plaintext method cannot attack the 5th round directly because the avalanche effect significantly confuses the relationship between the source key and the intermediate value leakage in the 5th round. So, we suggest that the head and tail protection must cover the first and last four rounds.

VI. CONCLUSION

Side-channel attacks will be of ongoing interest, particularly in the Internet of Things (IoT) environment.

In this paper, we demonstrated that when a distinguisher was applied to the Hamming distance leakage model, using a combination of different attack methods can break the security of the smart card. Specifically, our approach directly targeted either the third or fourth round of a DES encryption. This also proved the vulnerability of the "head and tail" protection.

REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology—CRYPTO*. Springer, 1999, pp. 388–397.

[2] D. Carluccio, K. Lemke, and C. Paar, "Electromagnetic side channel analysis of a contactless smart card: First results," in *Proc. ECrypt Workshop RFID Lightweight Crypto*, 2005.

[3] Y. Oren and A. Shamir, "Remote password extraction from RFID tags," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1292–1296, Sep. 2007.

[4] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.

[5] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.

[6] X. Hei, X. Du, S. Lin, and I. Lee, "PIPAC: Patient infusion pattern based access control scheme for wireless insulin pump system," in *Proc. INFOCOM*, Apr. 2013, pp. 3030–3038.

[7] L. Wu, X. Du, and J. Wu, "Effective defense schemes for phishing attacks on mobile computing platforms," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6678–6691, Aug. 2016.

[8] Y. Cheng, X. Fu, X. Du, B. Luo, and M. Guizani, "A lightweight live memory forensic approach based on hardware virtualization," *Inf. Sci.*, vol. 379, pp. 23–41, Feb. 2017.

[9] F. D. Garcia, P. van Rossum, R. Verdult, and R. W. Schreur, "Wirelessly pickpocketing a mifare classic card," in *Proc. 30th IEEE Symp. Secur. Privacy*, May 2009, pp. 3–15.

[10] T. Kasper, D. Oswald, and C. Paar, "EM side-channel attacks on commercial contactless smartcards using low-cost equipment," *Inf. Secur. Appl.*, pp. 79–93, 2009.

[11] D. Oswald and C. Paar, "Breaking mifare DESFire MF$_3$ICD40: Power analysis and templates in the real world," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2011, pp. 207–222.

[12] T. Korak and T. Plos, "Applying remote side-channel analysis attacks on a security-enabled NFC tag," in *Proc. Cryptographers' Track RSA Conf.* Springer, 2013, pp. 207–222.

[13] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Springer, 2004, pp. 16–29.

[14] L. Batina, B. Gierlichs, and K. Lemke-Rust, "Differential cluster analysis," in *Cryptographic Hardware and Embedded Systems*, vol. 5747. Springer, 2009, pp. 112–127.

[15] B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel, "Mutual information analysis," in *Cryptographic Hardware and Embedded Systems*. 2008, pp. 426–442.

[16] C. Tu, L. Zhang, Z. Liu, N. Gao, and Y. Ma, "A practical chosen message power analysis approach against ciphers with the key whitening layers," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Springer, 2017, pp. 415–434.

[17] A. Wang, Y. Zhang, W. Tian, Q. Wang, G. Zhang, and L. Zhu, "Right or wrong collision rate analysis without profiling: Full-automatic collision fault attack," *Sci. China Inf. Sci.*, vol. 61, no. 3, p. 032101, 2018.

[18] Q. Wang, A. Wang, L. Wu, and J. Zhang, "A new zero value attack combined fault sensitivity analysis on masked AES," *Microprocess. Microsyst.*, vol. 45, pp. 355–362, Sep. 2016.

[19] E. Oswald, S. Mangard, C. Herbst, and S. Tillich, "Practical second-order DPA attacks for masked smart card implementations of block ciphers," in *Proc. Cryptographers' Track RSA Conf.* Springer, 2006, pp. 192–207.

[20] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "ECDH key-extraction via low-bandwidth electromagnetic attacks on PCs," in *Proc. Cryptographers' Track RSA Conf.* Springer, 2016, pp. 219–235.

[21] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive physical side channels," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1626–1638.

[22] P. Belgarric, P.-A. Fouque, G. Macario-Rat, and M. Tibouchi, "Side-channel analysis of weierstrass and koblitz curve ECDSA on Android smartphones," in *Cryptographers' Track RSA Conf.* Springer, 2016, pp. 236–252.

[23] M. Nassar, Y. Souissi, S. Guilley, and J.-L. Danger, "RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2012, pp. 1173–1178.

[24] A. Gornik, I. Stoychev, and J. Oehm, "A novel circuit design methodology to reduce side channel leakage," in *Security, Privacy, and Applied Cryptography Engineering*. 2012, pp. 1–15.

[25] O. Reparaz and B. Gierlichs, "A first-order chosen-plaintext DPA attack on the third round of DES," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Springer, 2017, pp. 42–50.

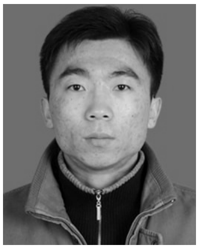
RIXIN XU received the B.S. degree in software engineering from the Harbin Institute of Technology and the M.S. degree in software engineering from Peking University. He is currently pursuing the Ph.D. degree in computer science with the Beijing Institute of Technology. His research interests include side-channel analysis and IoT security.

LIEHUANG ZHU is currently a Professor with the Department of Computer Science, Beijing Institute of Technology. He is selected into the Program for New Century Excellent Talents in University from the Ministry of Education, China. His research interests include Internet of Things, cloud computing security, and Internet and mobile security.

AN WANG was born in 1983. He received the Ph.D. degree from Shangdong University in 2011. From 2011 to 2015, he held a post-doctoral position at Tsinghua University. He is currently with the Beijing Institute of Technology. His main research interests include side-channel analysis, embedded systems, and cryptographic implementation.

XIAOJIANG DU received the B.S. and M.S. degrees in electrical engineering (Automation Department) from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park in 2002 and 2003, respectively. He is currently a tenured Full Professor and the Director of the Security and Networking Laboratory, Department of Computer and Information Sciences, Temple University, Philadelphia, USA. His research interests are security, wireless networks, and systems.

KIM-KWANG RAYMOND CHOO (SM'15) received the Ph.D. degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA), and has a courtesy appointment at the University of South Australia. He is a fellow of the Australian Computer Society. In 2015, he and his team won the Digital Forensics Research Challenge organized by the Germany's University of Erlangen-Nuremberg. He was a recipient of the 2008 Australia Day Achievement Medallion, the British Computer Society's Wilkes Award in 2008, the Fulbright Scholarship in 2009, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the ESORICS 2015 Best Research Paper Award, the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for tenured Faculty, and the IEEE TrustCom 2018 Best Paper Award. In 2016, he was named the Cybersecurity Educator of the Year–APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn). He serves on the Editorial Board of *Computers & Electrical Engineering*, *Computers & Security*, *Cluster Computing*, the IEEE Access, the IEEE Blockchain Newsletter, the IEEE Cloud Computing, the *IEEE Communications Magazine*, the IEEE Transactions on Big Data, *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, *PLoS ONE*, and *Soft Computing*.



GUOSHUANG ZHANG was born in 1982. He received the M.S. degree from the Zhengzhou Information Science and Technology Institute in 2009. He is currently a Research Assistant with the Science and Technology on Information Assurance Laboratory. His main research interests include lattice-based cryptography and cryptanalysis.

KEKE GAI received the B.Eng. degree from the Nanjing University of Science and Technology, the M.E.T. degree from The University of British Columbia, the M.S. and M.B.A. degrees from Lawrence Technological University, and the Ph.D. degree in computer science from the Department of Computer Science, Pace University, New York, NY, USA. He is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He has published over 90 peer-reviewed journals or conference papers, over 30 journal papers (including ACM/IEEE Transactions), and over 50 conference papers. His research interests include cloud computing, cyber security, combinatorial optimization, edge computing, and blockchain. He is involved in a number of professional/academic associations, including the IEEE and ACM. He was a recipient of five IEEE Best Paper Awards: the IEEE SSC'16; the IEEE CSCloud 2015; the IEEE BigDataSecurity 2015; the IEEE TrustCom 2018; and the IEEE HPCC 2018, and two IEEE Best Student Paper Awards (SmartCloud'16 and HPCC 2016) by IEEE conferences in recent years. His paper about cloud computing has been ranked as the Most Downloaded Articles of the *Journal of Network and Computer Applications*.