American Journal of Information Science and Computer Engineering
Vol. 6, No. 2, 2020, pp. 6-14
http://www.aiscience.org/journal/ajisce
ISSN: 2381-7488 (Print); ISSN: 2381-7496 (Online)

Differential Power Analysis Attack Using RSA-CRT Algorithm

Xiujun Wang[1, 2, *], Wei Wang[1, 2], Shu Guo[1, 2], Heng Zhang[3], Xiang Li[3], Siyuan Hai[3]

[1]Beijing Software Testing & QA Center, Zhongguancun Software Park, Beijing, China
[2]Beijing Key Laboratory of Software Testing Technology, Zhongguancun Software Park, Beijing, China
[3]School of Information Engineering, China University of Geosciences, Beijing, China

Abstract

RSA-CRT can improve the efficiency of modular exponentiation in RSA algorithm, which is becoming the most widely used encryption algorithm in digital signature and authentication of embedded cryptographic devices, and its security has also received a lot of attention. This paper studies the side channel attack of RSA-CRT algorithm, especially the differential power analysis attacks. The first four bytes of the parameter 'q' used to run the RSA-CRT algorithm on the STM32 chip are restored. When the key is restored, 512-bit random hexadecimal plaintext is sent to the single-chip microcomputer through the serial port loop, and then read the ciphertext processed by the chip through the serial port, feed back to the upper computer and record. At the same time, in the encryption process, the power trace generated by the STM32 chip in the process of encryption and decryption is collected and recorded to the upper computer through the electromagnetic probe, and through the oscilloscope combine the trigger signal to select the specific steps of the chip to process the data. (e.g. modular exponentiation in pre-calculation and modular multiplication in reorganization, etc.). The side channel attack system is combined to analyze the curve. Experimental results show that this method can improve the safety performance of the equipment.

Keywords

RSA-CRT Algorithm, CRT, DPA, Side Channel Analysis

Received: October 13, 2020 / Accepted: November 3, 2020 / Published online: November 27, 2020

1. Introduction

Information security generally refers to protecting the information to be transmitted to avoid potential threats, interference, and attacks, that is, to ensure the stability and security of information transmission. At the same time, with the rapid development of science and technology today, embedded devices have been widely used in logistics management, mobile information collection, barcode scanning and other fields, which have brought great convenience to our production and life. Because of this, the information security of embedded devices is very important. RSA algorithm is currently widely used in public key encryption and electronic commerce[1], The RSA-CRT algorithm is optimized on the basis of the original algorithm, which can perform encryption and decryption operations more efficiently.

Generally, when cracking the secret key of the RSA-CRT algorithm, the experimenter will use exhaustive attacks, mathematical attacks, timing attacks, chosen-ciphertext attacks, etc. These attack methods are all used to restore the key from a mathematical perspective or the main channel direction. This design uses the side channel attack system, which is used for chip side channel testing (including smart IC cards). It includes hardware and software platforms, which

* Corresponding author
E-mail address: wangxj@bsw.net.cn (Xiujun Wang)

can collect and analyze the power consumption of the chip and electromagnetic radiation signals. Differential power analysis (DPA) is a type of side channel attack, it collects the power leakage curve generated by the embedded devices in the process of password-related operations, assumes the median value. of the algorithm, and then uses the corresponding correlation coefficient analysis to decode the key used in the encryption and decryption process of the embedded device. From results, the attack capability of this attack method is much better than the traditional analysis method.

In the practical application of RSA-CRT algorithm, there are two realization forms: Garner form and Gauss form. At present, there are many attacks in the form of Garner , for example, the SPA attack proposed by Novka is implemented in the form of Garner, and the key is broken through the eigenvalue analysis of subtraction calculation; Marc Witteman's DPA attack on the implementation of the RSA-CRT algorithm Garner.

DPA attack refers to the comparative analysis and utilization of the electromagnetic power consumption of the chip in embedded devices and the correlation coefficient between the encryption key applied therein. The reason why the DPA attack can receive attention and research is that the attack is non-invasive, so the intruder can destroy the embedded system without producing traces.

For this kind of attack, currently feasible defense schemes are: increasing the mask to makes the acquisition of the power consumption curve more difficult; using smart IC card, because each time the card is used, the IC card will generate a corresponding new key from the old key that was used previously, thus rendering the key acquired by the password attack worthless. At present, the mobile internet represented by smart home, etc. and various applications are advancing rapidly, making embedded devices and other hardware devices gradually enter the lives of the public. The wide application of these devices makes their actual security problems. Drones and Samsung Galaxy S7 have been experimentally proved to have certain security risks, and there is a risk of leakage of users' private information.

2. Methods

2.1. The Principle of RSA-CRT Algorithm

The RSA algorithm is the most widely used "asymmetric encryption algorithm" in the world [2]. It was designed by three outstanding mathematicians in 1977 and named with the initials of three people. This algorithm is mathematically secure, the longer you choose the key, the harder it is to crack. The RSA algorithm is secure because it is based on the difficulty of factorizing extremely large integers. RSA algorithm is much less secure if there is an algorithm that can implement fast factorization. However, until now, only people have implemented the exhaustive cracking of the short key RSA algorithm, so the information processed by the crypto equipment cannot actually be cracked under the condition of ensuring the key length. Based on this nature of the algorithm, RSA has been adopted as a standard by famous international standards organizations such as SWIFT, ISO, and ITU [3].

From the perspective of the security of the RSA algorithm, the longer the key is, the more secure the whole encryption and decryption process will be. Therefore, people usually choose a longer key, which makes the encryption and decryption operations of the device take a long time. But in practical applications, Chinese Remainder Theorem (CRT) [4] is selected to speed up the calculation in most cases.

When optimizing the RSA algorithm, first pre-calculate the intermediate parameters of the algorithm:

$$V_p \equiv C^d \bmod p;$$

$$V_q \equiv C^d \bmod q;$$

$$X_p \equiv q \times q^{-1} \bmod p;$$

$$X_q \equiv p \times p^{-1} \bmod q;$$

$$M \equiv \left(V_p \times X_p + V_q \times X_q\right) \bmod n;$$

Furthermore, Fermat's theorem can be used to simplify the calculation:

$$V_p \equiv C^d \bmod p \equiv C^d \bmod (p-1) \bmod p;$$

$$V_q \equiv C^d \bmod q \equiv C^d \bmod (q-1) \bmod q;$$

Since there are many complex modular exponentiation operations for large prime numbers in the calculation of the RSA algorithm, it is usually chosen to combine with CRT to improve its operation speed. After the combination of the CRT and the RSA algorithm, the speed of encryption and decryption process using RSA-CRT algorithm can be increased by about four times when the same length of key is used and the ciphertext with the same content and format is processed. Generally, RSA algorithm can be optimized in two forms: RSA-CRT algorithm in Gauss mode and RSA-CRT algorithm in Garner mode [5]. RSA-CRT algorithm in the form of Garner. The details are as follows:

Input: Positive integer M: $M = \prod_{i=1}^{t} m_i > 1$, and $\gcd(m_i, m_j) = 1$ for all i and j. And the remainder of x in the prime number field is $(v_1, v_2 \cdots v_j)$.

$$U = m_j - 1 (\bmod m_i)$$

$$C_i = u \times C_i (\bmod m_i)$$

i from 2 to t, j from 1 to $i-1$;

$$U = (v_i - x) \times C_i \bmod m_i;$$

$$x = x + u \prod_{j=1}^{i-1} m_j;$$

Output: integer x.

For the RSA algorithm, the operation formula of Garner combined with Garner is:

$$C = S_p + \left(\left((S_p - S_q) \times qinv \bmod p\right) \times p\right) \bmod N \quad (1)$$

The RSA-CRT algorithm in the form of Gauss is directly obtained by using the CRT:

$$C = \left(S_p \times q \times qinv + S_q \times p \times pinv\right) \bmod N \quad (2)$$

By comparing the two equations (1) and (2), we can find that the Gauss method requires more modular multiplication operations during the calculation process, and the calculation requires more resources, so the Garner method is used by more people.

2.2. DPA Attack on RSA-CRT Algorithm

With the popularity of cryptographic devices, there are a variety of attacks on cryptographic devices. From the perspective of the time spent in the attack, the instruments used in the attack, and the specific knowledge involved, the methods of attack are different. In terms of the performance of cryptographic devices, it can be divided into two forms: passive attacks and active attacks [6].

In the case of passive attack, the embedded device still performs normally according to its algorithm in the process of encryption and decryption, while the attacker guesses the key by observing the physical characteristics of the embedded device (riming time of algorithm, electromagnetic power consumption curve generated during encryption, etc.) [7], and finally restores the original key through mathematical analysis. When the device is actively attacked, the attacker will adjust some parameters in the algorithm, which will affect the intermediate value of the algorithm. Through the input of this control algorithm and the abnormal operation of the device, the key will be analyzed and restored [8].

From the point of view of the "interface" used to attack embedded devices, it can be divided into intrusive attack, non-invasive attack and semi-invasive attack [9]. Among them, the most powerful attack and the most effective is the intrusive attack. However, in actual experiments, due to multiple limitations of equipment and implementation conditions, this attack method is not suitable for use in many cases. At present, there are few researches on intrusive attack of cryptographic devices at home and abroad. Semi-invasive attacks also cost more and have higher requirements on the specific internal structure and expertise of the chip.

In contrast, non-intrusive attacks are more widely used. This method will not have much impact on the process of encryption and decryption devices. It only needs to use the interface of the device and does not produce any traces. In addition, non-intrusive attacks have low requirements for devices and can be carried out on many devices [10]. Therefore, in fact, they pose a huge threat to the encryption and decryption process of cryptographic devices.

Side channel attack is a non-invasive attack, and it is also a kind of passive attack. There are three kinds of side channel attacks: power analysis attack, timing attack and electromagnetic attack. Power analysis attack is the most used one, which has significant amplification effect, lower experimental cost and higher implementation feasibility. DPA has received the most attention. The advantage of DPA is that the attacker does not necessarily need to know the detailed knowledge of the attacked device, but only needs to know which encryption algorithm is used by the device. Therefore, DPA is gradually becoming the most used power analysis attack.

DPA requires analysis of the power consumption curve of the data processed by the device, and the generation of the power consumption curve is based on CMOS [11] technology. CMOS is a logical structure, and some logic elements are based on CMOS. Compared with other components, the performance of these components is relatively small in static consumption, but they are continuously affected by output signals. When the output signal changes, there will be a large energy consumption. And this kind of dynamic energy consumption has a great dependence on the calculated parameters. During the execution of DPA, the collected power trace reflects the energy consumption of the device during a specific step, and the correlation between the processed parameters and the curve can be further analyzed based on this. The core of DPA is that cryptographic equipment has different input and intermediate values and energy consumption in different calculation steps during data processing. Therefore, the key can be recovered by analyzing the power consumption curve generated by the device.

In addition, DPA can restore the key used by the device when the power trace is affected by external noise. For example, suppose there is a smart IC card user who collects a large number of power traces generated during the process of deposit and withdrawal, the key used by the IC card can be restored by DPA attack on the collected data when necessary in the future.

In most cases, DPA attacks are based on the same process system. which generally consist of five steps, as shown in Figure 1: In the first step of the process, the function $f(d, k)$ is

selected. d represents the non-constant data published at the beginning of the experiment, and k is the content of the key. Normally, d selects plaintext or ciphertext. The second step of the process records the power consumption information of the embedded device when processing data. For the power trace with X data packets and length Y, it can be recorded in the form of a matrix, usually recorded as a matrix W of size $X \times Y$, in most cases, a trigger signal will be used to align the power consumption curve to ensure that the power trace represents the same operation. In the DPA attack, the third step is usually called the key assumption. For the recorded of X times encryption process and the possible value vector $k = (k_1 \cdots k_k)$, the operator can calculate $f(d, k)$ and further get the matrix Z with the size of X×K, and finally locks which column of the matrix Z is processed by the embedded device during X times of encryption, so as to determine k_{ck}.

And then at the fourth step, the previously obtained matrix Z will be mapped to the hypothetical energy consumption value matrix H [12]. The most common energy models that map Z to H are the Hamming distance model and the Hamming weight model. In this experiment, the Hamming weight model is adopted. Generally, the Hamming distance model is preferred during the experiment, and the Hamming weight model is selected only when the Hamming distance is not suitable. The algorithms implemented on hardware mostly use the Hamming distance model, while the algorithms implemented by software mostly use the Hamming weight model.

After the matrix Z is mapped to the matrix H, the last step is performed [13]. Compare the hypothetical energy consumption value corresponding to each key hypothesis with the power trace recorded at each location. Finally, a matrix C of size $K \times Y$ is obtained, whose elements respectively represent the comparison results of each column of the matrix H and the matrix W. Then search for the maximum value of each element in matrix C to determine the correct key index and the position of the corresponding point in the curve, so that the experimenter can get the key used in the process of encryption and decryption of embedded device .

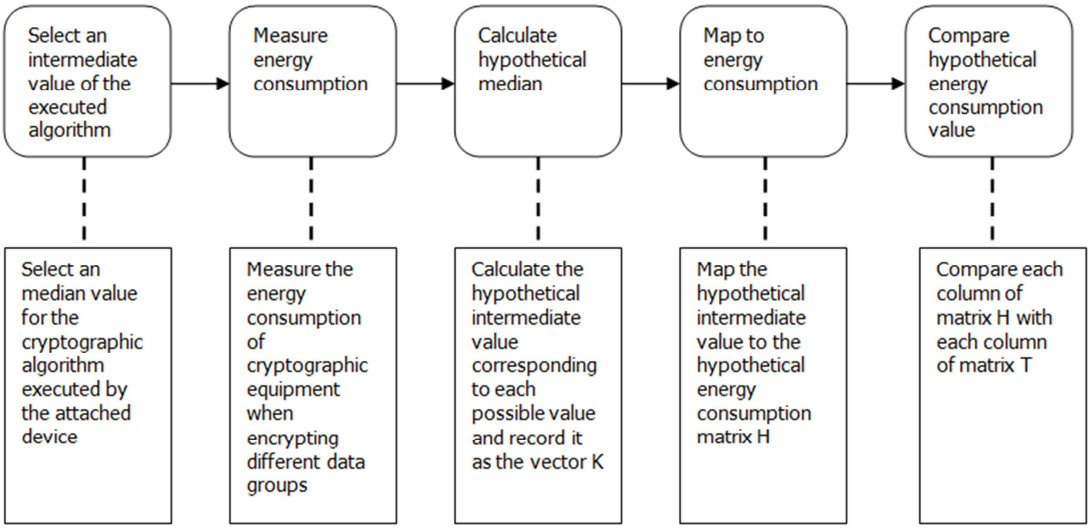

Figure 1. General flow chart of DPA attack.

When the RSA algorithm is combined with the CRT, it can be marked by applying a trigger signal to the single-chip microcomputer during the pre-calculated modular exponentiation operation and the modular multiplication step of the recombination operation, and then the device can collect electromagnetic information. Add a trigger signal when entering the decryption function to avoid interference from other steps of the modular multiplication operation. The specific code is as follows:

```
int status;
uint8_t pkcs_block[RSA_MAX_MODULUS_LEN];
uint32_t modulus_len, pkcs_block_len;
modulus_len = ( sk -> bits + 7 ) / 8;
if(in_len > modulus_len)
return ERR_WRONG_LEN;
//rising edge
GPIO_SetBits(GPIOA, GPIO_Pin_1);
status = private_block_operation(pkcs_block, &pkcs_block_len, in, in_len, sk);
GPIO_ResetBits(GPIOA, GPIO_Pin_1);
//falling edge
×out_len = 64;
memcpy((uint8_t ×)out, (uint8_t ×)&pkcs_block[0], ×out_len);
```

```
//Add a trigger signal to mark the specific curve position during the reorganization to launch the attack
bdigits = bn_digits(b, digits);
cdigits = bn_digits(c, digits);
for(i=0; i<bdigits; i++) {
if(flag==3&&i==1){
// Add a rising edge signal, when starting to process the second byte.
GPIO_SetBits(GPIOA, GPIO_Pin_2);
}
if(flag==3&&i==2){
GPIO_ResetBits(GPIOA, GPIO_Pin_2);
}
// Add a falling edge signal, when starting to process the third byte.
t[i+cdigits] += bn_add_digit_mul(&t[i], &t[i], b[i], c, cdigits);
}
bn_assign(a, t, 2×digits);
```

On the oscilloscope, the experimenter can select a specific part of the curve as the target of attack by observing the position of the trigger signal. As shown in Figure 2, two trigger signals are set to confirm the part of the electromagnetic radiation signal curve where the device processes the first byte.

2.3. Experimental Steps

Step 1. Using Keil 5, the parameter-set C language code (see the appendix) is burned into the STM32, and the code should retain the full COS of the original chip, i.e. without any shielding and modification, as well as defenses such as masks. The single-chip microcomputer is linked to the upper computer through the serial port, and configure the serial port, the oscilloscope and the upper computer are connected by the network cable, And through TCP/IP to make it in the same network segment to achieve communication.

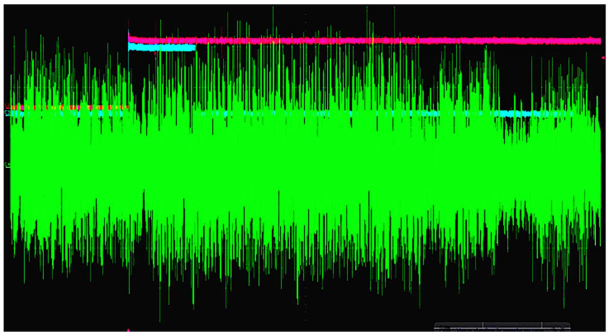

Figure 2. Graph of the first byte position.

Step 2. After the configuration is complete, run the RSA-CRT algorithm on the software equipped with the side channel attack system, and this information click here to collect and analyze the information to attack the script.

Step 3. Repeat the collection of electromagnetic radiation signal curve. Set the trigger signal in the source code. During the operation of the STM32 single-chip microcomputer, the oscilloscope is triggered through the chip pin to collect the signal. The oscilloscope completes the electromagnetic radiation signal curve at the time of the operation of the cryptographic device through the sampling circuit. By adjusting the position of the chip, observe the waveform when the electromagnetic information is the most obvious on the oscilloscope to determine the electromagnetic acquisition location. When collecting, it is necessary to increase the sampling frequency above 250M and use the trigger delay.

After setting the oscilloscope trigger source, click "Repeat Acquisition", the STM32 MCU will send the trigger signal, and the oscilloscope will capture the trigger signal to complete the acquisition of the power consumption signal. For repeated acquisition, in the pop-up dialog box, select the curve save path. When collecting curves, it is necessary to collect a large number of electromagnetic radiation signal curves, so that the correlation estimation result can be more accurate when analyzing the curve.

Step 4. Open the collected curve on the software. According to the trigger signal added in the code, we can determine which part of the power consumption curve is the modular multiplication part that needs attention, and which part is the modular exponentiation part. The power consumption curve interface parameter description is shown in Table 1.

Step 5. Align the power consumption curve. Select the middle spike section for alignment. It should be noted that alignment and filtering themselves are operations without any correlation, and both can be exchanged.

Step 6. Resample the power consumption curve. Before the attack, integrate the power consumption curve in each clock. Therefore, when Resample, it is necessary to know clearly the clock of the target to be attacked. Generally, the clock frequency can be analyzed through spectrum analysis. After obtaining the driving clock of the attack target, use this frequency or a multiple of the frequency for re-sampling processing, which can achieve the effect of integrating the power consumption curve in the clock.

Step 7. Attack the curve. Use RsaCrt Analysis analysis method to analyze the power consumption curve.

Table 1. The description of interface parameters.

Parameter name	Meaning
Starting curve	Select the first curve
Ending curve	Select the last curve
initiation point	Select the first sample point to be processed
end point	Select the last sample point to be processed
start	The position of the first data unit for calculating correlation data
Number of units	Number of data units for calculating correlation
Unit bit number	Data unit for calculating correlation, 1 means 1 bit, 8 means 1 byte
Public keys E	The value of the key bit of the experiment that needs to be used in the correlation calculation
Modulus N	The value of modulus N in RSA algorithm
Numbers of multithreading	Number of threads for multitasking

3. Results

3.1. Attack Test on 512-bit Plaintext

3.1.1. Analysis of the Second Byte of Parameter Q

When collecting, the upper computer will send 256 (or 512) hexadecimal ciphertext to the serial port in bytes in a cycle, and then read the encrypted plaintext from the serial port.

Adjust the position of the trigger signal in the source code, determine on the oscilloscope the part of the curve where the cryptographic device processes the second byte, and then collect the electromagnetic radiation signal.

Use hamonics to filter out the power consumption noise caused by the external clock, filter, and then align and resample.

The results after the attack are as follows:

Best correlation

(1) Candidate key: [0x87] The correlation value is [0.04787129] at point [12470]

(2) Candidate key: [0xd5] The correlation value is [0.04758301] at point [14276]

(3) Candidate key: [0xa4] The correlation value is [0.047361] at point [5554]

(4) Candidate key: [0x6f] The correlation value is [0.04673735] at point [25815]

(5) Candidate key: [0x31] The correlation value is [0.04552473] at point [47072]

(6) Candidate key: [0x73] The correlation value is [0.04542644] at point [7516]

(7) Candidate key: [0x47] The correlation value is [0.04537982] at point [80239]

(8) Candidate key: [0xcd] The correlation value is [0.0452986] at point [92802]

(9) Candidate key: [0xbd] The correlation value is [0.0449816] at point [7583]

(10) Candidate key: [0x6] The correlation value is [0.04442649] at point [27135]

(11) Candidate key: [0xba] The correlation value is [0.04437806] at point [37860]

(12) Candidate key: [0x78] The correlation value is [0.04426808] at point [7516]

(13) Candidate key: [0x77] The correlation value is [0.04396128] at point [7518]

(14) Candidate key: [0xc8] The correlation value is [0.04373333] at point [37501]

(15) Candidate key: [0x65] The correlation value is [0.04364035] at point [8142]

(16) Candidate key: [0xbe] The correlation value is [0.04362432] at point [37850]

(17) Candidate key: [0xc6] The correlation value is [0.04356924] at point [7588]

(18) Candidate key: [0x33] The correlation value is [0.04338891] at point [2485]

(19) Candidate key: [0x4b] The correlation value is [0.0430641] at point [94658]

(20) Candidate key: [0x4f] The correlation value is [0.04304987] at point [67417]

The above is the candidate key analyzed by RsaCrtAnalysis. Note that the public keys and modulus need to be filled in the menu. Because the modulus is used to generate intermediate results and do correlation calculations, the public key is to traverse search for q, and the private key can be calculated through the public key. Use the menu to attack directly. In the previously found prime bytes box, the FA byte is filled in. The actual analysis does not need to fill in the byte. This menu directly analyzes the result of the power consumption curve. In this experiment, in order to analyze the characteristics of power leakage, fill in the FA byte and observe the subsequent attack results. In the result, "87" appears very close to the second byte "88" of q. From the experimental results, a certain amount of electromagnetic information leakage occurred at this position on the chip.

3.1.2. Analysis of the Third Byte of Parameter Q

By adjusting the source code of the STM32 chip and changing the trigger signal, we can lock on the oscilloscope the part where the device processes the third byte in the curve.

Perform the same filtering, resampling, static alignment and other processing on the collected electromagnetic radiation curve

about the third byte, and then analyze the processed curve.

The results of the attack are as follows:

Best correlation

(1) Candidate key: [0xaa] The correlation value is [0.06703768] at point [4834]

(2) Candidate key: [0xaf] The correlation value is [0.06567978] at point [2187]

(3) Candidate key: [0xa1] The correlation value is [0.06250887] at point [4962]

(4) Candidate key: [0x3b] The correlation value is [0.061524] at point [4995]

(5) Candidate key: [0x3c] The correlation value is [0.06139879] at point [2187]

(6) Candidate key: [0x3e] The correlation value is [0.06107543] at point [2187]

(7) Candidate key: [0xca] The correlation value is [0.05947958] at point [5631]

(8) Candidate key: [0xf4] The correlation value is [0.05911126] at point [4834]

(9) Candidate key: [0xf6] The correlation value is [0.05893514] at point [4849]

(10) Candidate key: [0x3] The correlation value is [0.05888109] at point [1492]

(11) Candidate key: [0x50] The correlation value is [0.05878312] at point [1345]

(12) Candidate key: [0xf5] The correlation value is [0.05838756] at point [4787]

(13) Candidate key: [0x68] The correlation value is [0.057971] at point [666]

(14) Candidate key: [0x47] The correlation value is [0.05764499] at point [1318]

(15) Candidate key: [0xf3] The correlation value is [0.05742567] at point [1836]

(16) Candidate key: [0x59] The correlation value is [0.05726628] at point [683]

(17) Candidate key: [0x5e] The correlation value is [0.05709403] at point [683]

(18) Candidate key: [0x4] The correlation value is [0.05696862] at point [1492]

(19) Candidate key: [0x79] The correlation value is [0.05682706] at point [4962]

(20) Candidate key: [0x63] The correlation value is [0.05645072] at point [4945]

3.1.3. Analysis of the Fourth Byte of Parameter Q

Re-modify the source code to lock the part of the curve that processes the fourth byte.

After processing the electromagnetic radiation curve part of the collected fourth byte, analyze the curve again,

The results of the analysis are as follows:

Best correlation

(1) Candidate key: [0x7b] The correlation value is [0.06962758] at point [5859]

(2) Candidate key: [0x7c] The correlation value is [0.06374717] at point [2671]

(3) Candidate key: [0x79] The correlation value is [0.06344516] at point [3026]

(4) Candidate key: [0x4f] The correlation value is [0.06127578] at point [1101]

(5) Candidate key: [0xca] The correlation value is [0.06126913] at point [759]

(6) Candidate key: [0x61] The correlation value is [0.0611571] at point [156]

(7) Candidate key: [0xb2] The correlation value is [0.06064361] at point [4145]

(8) Candidate key: [0x1e] The correlation value is [0.06034289] at point [1097]

(9) Candidate key: [0x8f] The correlation value is [0.0600964] at point [4571]

(10) Candidate key: [0xac] The correlation value is [0.05989924] at point [4145]

(11) Candidate key: [0xa8] The correlation value is [0.05983479] at point [1694]

(12) Candidate key: [0xd3] The correlation value is [0.05976565] at point [2650]

(13) Candidate key: [0xe7] The correlation value is [0.05946249] at point [3497]

(14) Candidate key: [0xf1] The correlation value is [0.05944555] at point [1164]

(15) Candidate key: [0x6c] The correlation value is [0.05944065] at point [3009]

(16) Candidate key: [0x57] The correlation value is [0.05943002] at point [4632]

(17) Candidate key: [0xfd] The correlation value is [0.05942644] at point [587]

(18) Candidate key: [0x73] The correlation value is [0.05938115] at point [3536]

(19) Candidate key: [0x24] The correlation value is [0.05934943] at point [3632]

(20) Candidate key: [0xdc] The correlation value is [0.05934076] at point [4273]

4. Discussions

DPA attacks are currently the most widely used, and the main reason is that the power consumption of the current cryptographic equipment when processing data depends on the median value of the cryptographic algorithm executed by the equipment. So the limitations of DPA are very obvious. Experimenters can improve the safety of the equipment by eliminating the data dependence of the equipment. The core idea of the current popular defense schemes is also to minimize the correlation between the electromagnetic information generated by the equipment and the parameters used. Based on this idea, DPA attacks can be restricted from two perspectives: hiding technology can be used in hardware, and masking technology can be used in software. Figure 3 illustrates the principle of reducing data correlation from these two perspectives.

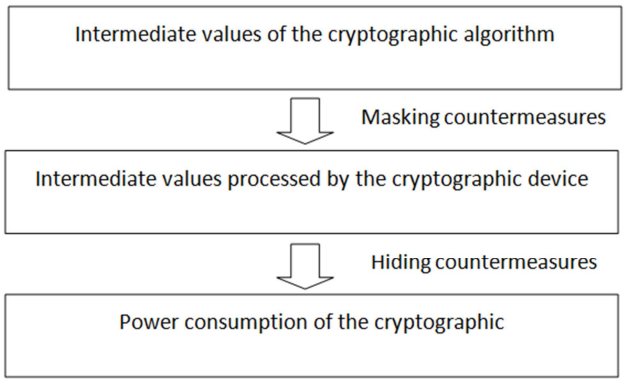

Figure 3. Basic concepts of hiding countermeasures and masking countermeasures.

From the perspective of hardware analysis, the essence of using hiding technique is to minimize the data dependence of the electromagnetic information generated by the device [14]. That is to say, the steps of the device during the encryption and decryption process should be randomized, or the electromagnetic information generated by the device should be changed, so that data dependence can no longer be a breach of the device key. To achieve this, the feasible operation methods are divided into two categories: modify the equipment from the hardware so that the electromagnetic information leaked by the equipment during data processing is completely consistent, or change the electromagnetic information generated by the equipment to random, no matter which one can greatly eliminate the data dependence of electromagnetic information. But in the specific experiment process, it is unrealistic to completely erase the data dependence, and the equipment will always be affected by the input data to a certain extent.

The hiding countermeasures implemented by hardware usually judges its performance by two standards: whether the electromagnetic information leakage generated by the cryptographic device in different clock cycles obeys random distribution; or whether the cryptographic device can generate completely consistent results during each specific operation electromagnetic information leakage. In addition, the hiding technique implemented by changing the electromagnetic leakage of the hardware will not affect the intermediate value calculated by the device during the encryption and decryption process of the data. This measure focuses on changing electromagnetic leakage to prevent DPA.

When implementing a DPA attack, one of the important steps is to statically align the collected power traces, that is, to make the same part of different power traces represent the same operation. When static alignment processing is not possible, a large number of curve collections are required. Therefore, DPA defense can be carried out from this perspective, and the sequence of operations performed by the device during data processing can be disrupted, or randomly added in the entire process operations that are irrelevant and do not affect the processing results. The higher the randomness of the operation in the entire process, the better the overall security can be guaranteed. Hiding technique is mainly to greatly weaken or even eliminate the correlation between the electromagnetic information leaked by the cryptographic device when processing data and the intermediate value of the specific calculation. In this case, even if the experimenter collects a large number of equipment electromagnetic radiation signal curves, it is still difficult to analyze and obtain information that is helpful to restore the algorithm key.

The masking technology implemented from the software perspective is very different from the hardware aspect. When the masking countermeasure is implemented [15], the parameters in the algorithm calculation process will be combined with the "mask" for processing. The mask is randomly selected by the device and changes accordingly in different operations. Therefore, the mask is sealed to the outside world. The parameters will be combined with the "mask" to produce the masked intermediate value. And this intermediate value will greatly increase the safety of the previous parameters. When implementing this defense strategy, it is a crucial aspect to ensure that all intermediate values generated by the device when processing data are always in a sealed state. For example, when the device needs to perform exclusive OR operation on two parameters during operation, the result of the exclusive OR operation must be sealed.

Commonly used masking techniques include Boolean mask, arithmetic mask, etc. When the device runs certain algorithms, it may include two operation modes. In this case, a combination of Boolean mask and arithmetic mask is required. In asymmetric encryption algorithm such as the RSA algorithm, addition or multiplication mask are usually selected. This operation is also called "blinding". The two commonly used plans are "index blinding" for processing index and "message blinding" for processing plaintext.

5. Conclusions

This article uses STM32 to run the RSA-CRT encryption algorithm. The upper computer cyclically sends the randomly generated hexadecimal ciphertext to the single-chip microcomputer through the serial port, and at the same time continuously collects the electromagnetic information leakage generated by the device during the encryption process, and then uses the side channel attack system to filter, resample, align and analyze the collected power traces, and finally tries to crack the key. From the experimental results, the software restores the second byte of the parameter q of the algorithm, and the candidate key that appears indicates that there is a leak of information that can be passed at this position of the chip. Finally, two feasible DPA countermeasures are explained, which can improve the safety performance of the equipment.

Acknowledgements

Supported by State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCH 201806.

References

[1] Feng Yan. (2013). Research on smart card attack and defense based on AES algorithm (Master's thesis, Beijing Jiaotong University).

[2] Edward David Moreno, Leila C. M. Buarque, Floruio Natan & Ricardo Salgueiro (2016). Impact of Asymmetric Encryption Algorithms in a VANET. 11(12), 1118-1131.

[3] Lu Pengyu. (2012). Research and design of general authorization service based on cloud computing and rule engine (Master's thesis, Beijing University of Posts and Telecommunications).

[4] Jagdish C. Patra and Cedric Bornand(2010). A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression. 20(6), 1597-1611.

[5] Song Nan. (2015). Research on data security of multi-campus all-in-one card system (Master's thesis, Xi'an University of Architecture and Technology).

[6] Du Zhenyu, Liu Fangzheng and Li Yihong. (2019). APT attack path prediction based on HMM. System Engineering and Electronic Technology (04), 826-834.

[7] Zhi Jingsong. (2017). Research and design of HDCP2.2 transmitter in HDMI (Master's thesis, Beijing University of Technology).

[8] Shi Meng. (2013). Application of LLL algorithm in RSA security analysis (Master's thesis, PLA Information Engineering University).

[9] Yu Yanyan. (2012). Lightweight block cipher algorithm collision energy attack (Master's thesis, Shandong University).

[10] Li Zengju, Shi Ruhui, Wang Jianxin, Li Chao, Li Haibin and Shi Xinling. (2016). Selected plaintext attack on Gauss form CRT-RSA based on DPA. Chinese Journal of Cryptography (02), 202-210.

[11] Zhou Yi, Wang Lei, Li Jun, Yang Xuelei, Gan Fengyuan, Zhao Yingxuan... and Li Wei. (2020). CMOS-compatible mid-infrared multi-channel photonic crystal sensor (English). Journal of Infrared and Millimeter Waves (03), 279-283.

[12] Li Zengju, Peng Qian, Shi Ruhui, Li Chao, Ma Zhipeng and Li Haibin. (2016). Selected plaintext attack on CRT-RSA algorithm. Chinese Journal of Cryptography (05), 447-461.

[13] Gan, Zhang, King Khan.(2019). An improved differential power analysis against random process interrupts. Journal of the Chinese Institute of Engineers 42(2): 127-131.

[14] Xu Shubin, Jia Zhe and Zhang Haifeng. (2018). Network feature dynamic hiding technology based on SDN. Journal of Communications (S2), 28-34.

[15] Xu Pei and Fu Li. (2016). Software masking scheme to prevent differential power analysis attacks. Computer Application Research (01), 245-248.