A Differential Power Analysis Attack on Dynamic Password Token Based On SM3 Algorithm

Limin Guo, Qing Li, Lihui Wang, Zhimin Zhang, Dan Liu, Weijun Shan
Shanghai Fudan Microelectronics Group Company Limited, Shanghai, 200433, China
E-mail: guolimin@fmsh.com.cn

***Abstract*—Dynamic password technology is one of the most widely utilized methods for identity authentication. The security of dynamic password system depends on the cryptographic strength of the underlying hash function. And SM3 is the only standard hash algorithm of China. However, most cryptographic algorithm implementations are vulnerable against side channel attacks. But specific side channel attacks on dynamic password token based on SM3 hash function have not been given so far. This paper presents a differential power analysis attack on dynamic password token based on SM3 algorithm. SM3 hash algorithm is based on the mixing of different algebraic operations, such as XOR and addition modulo 2^{32}, thus the proposed DPA attack is mainly against these basic group operations. Experimental results are given by attacking an implementation of generating dynamic password using SM3 algorithm in a smart card, which demonstrate the feasibility of such attacks described in this paper.**

Keywords—Dynamic password token; SM3; DPA;

I. INTRODUCTION

Various cryptographic algorithms are prevalent applied in economy, military, Government and so on to provide information security. Identity authentication is the basis of the entire information security system. It is mainly used to detect and exclude the unauthorized accesses. Static password authentication scheme is utilized for identity authentication, since it is simple and easy to use. However, it can be intercepted or impersonated easily and is vulnerable to replay attacks *et al*. Dynamic password technology is an enhancer to traditional passwords and avoids a number of shortcomings that are associated with static password based authentication. At present, synchronous dynamic password and asynchronous dynamic password are two dominating dynamic password techniques. Whereby, synchronization dynamic password technology are classified into two categories: time-synchronized dynamic password and event- synchronized dynamic password. One of the most widely used methods to generate dynamic password is based on time synchronization.

The security of dynamic password system relies on the cryptographic strength of the underlying hash function. Nowadays, the widely used hash functions are RSA, SHA-1, MD4, MD5 and so on [1] [2]. Wang *et al*. have presented powerful attacks on MD5 and SHA-1, which allow finding collisions efficiently [3] [4]. Thus, it is a potential risk for using these hash algorithms to generate dynamic password.

SM3 [5] hash algorithm is released by China's Office of Security Commercial Code Administration, and certificated as the only standard hash algorithm of China in 2010. Futhermore, it is recommended to use SM3 hash function to generate dynamic password in China. Thus, it is of great importance to study on the security of dynamic password token based on SM3 algorithm. Many side channel attacks on hash algorithm have been reported already. Many hash algorithms are based on the mixing of different algebraic operations. Lemke *et al*. [6] introduced a side channel attack on the basic group operations, such as XOR, addition modulo 2^n and modular multiplication using multi-bit selection functions. Okeya *et al*. [7] [8] evaluated the security of HMAC algorithm based on block-cipher based hash functions. McEvoy *et al*. [9] discussed a differential side-channel attack on an implementation of the HMAC algorithm that uses the SHA-2 hash function family. But as far as we know, the resistance of SM3 to side channel attacks still remains uncertain. In this paper, we propose a first order DPA attack on dynamic password token based on SM3 algorithm. Furthermore, we provide attack results on a software implementation of generating the dynamic password using SM3 algorithm. The rest of this paper is organized as follows. Section II introduces the background theory regarding the SM3 algorithm, the dynamic password algorithm, and DPA attacks. Section III presents a DPA attack on dynamic password token based on SM3 algorithm. Attack results are given in Section IV. Finally, we conclude in Section V.

II. BACKGROUND

The following section will briefly cover the background theory, required for understanding this work. First, we give an overview of SM3 algorithm, dynamic password algorithm and followed by a brief introduction to differential power analysis.

A. SM3 hash algorithm descryption

Full description of the SM3 hash algorithm can be found in the official OSCCA standard [1]. SM3 maps a message of length l ($l < 2^{64}$) bits to produce a message digest of 256 bits. The SM3 algorithm essentially consists of three stages: (i) message padding and parsing; (ii) message expansion; (iii)message compression.

Message Padding and Parsing. The binary message to be processed is appended with a single bit "1" followed by zeros until the padded message bit length equivalent to $448 \bmod 512$. The original message length is then appended as a 64-bit binary number. The resultant message is then parsed into n 512-bit blocks, denoted as $B^{(i)}$, for $0 \le i \le n-1$. These $B^{(i)}$ message blocks are iteratively passed to the message expansion stage.

Message Expansion. In this stage, each $B^{(i)}$ is expanded into 132 32-bit words W_j, for $0 \le j \le 67$ and W_j', for $0 \le j \le 63$. Each 512-bit $B^{(i)}$ block from the message padding and parsing stage is viewed as sixteen 32-bit words denoted as W_j, for $0 \le j \le 15$ and generate other 32-bit words according to equations given by:

$$W_j = P_1\left(W_{j-16} \oplus W_{j-9} \oplus \left(W_{j-3} \lll 15\right)\right) \oplus \left(W_{j-13} \lll 7\right) \oplus W_{j-6}, 16 \le j \le 67 \quad (1)$$

$$W\_j\text{^}2 = W\_j \oplus W\_(j+4), 0 \le j \le 63 \quad (2)$$

where $P_1(X) = X \oplus (X \lll 15) \oplus (X \lll 23)$ and $x \lll k$ denotes a circular rotations of x by k positions to the left.

Message Compression. The SM3 compression function *CF* utilizes eight 32-bit word registers labeled A, B, C, D, E, F, G, H, which are initialized to the 256-bit initial value at the beginning of each call to the hash function. Sixty-four iterations of following operations are sequentially utilized for computing the hash value:

$$SS1 \leftarrow \left((A \lll 12) + E + \left(T_j \lll j\right)\right) \lll 7 \quad (3)$$

$$SS2 \leftarrow SS1 \oplus (A \lll 12) \quad (4)$$

$$TT1 \leftarrow FF_j(A,B,C) + D + SS2 + W_j^{2} \quad (5)$$

$$TT2 \leftarrow GG_j(E,F,G) + H + SS1 + W_j \quad (6)$$

$$D \leftarrow C \quad (7)$$

$$C \leftarrow B \lll 9 \quad (8)$$

$$B \leftarrow A \quad (9)$$

$$A \leftarrow TT1 \quad (10)$$

$$H \leftarrow G \quad (11)$$

$$G \leftarrow F \lll 19 \quad (12)$$

$$F \leftarrow E \quad (13)$$

$$E \leftarrow P_0(TT2) \quad (14)$$

where T_j are constants defined as below:

$$T_j = \begin{cases} 79cc4519 & 0 \le j \le 15 \\ 7a879d8a & 16 \le j \le 63 \end{cases}$$

and FF_j, GG_j, P_0 are given by:

$$FF_j(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z & 0 \le j \le 15 \\ (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z) & 16 \le j \le 63 \end{cases}$$

$$GG_j(X,Y,Z) = \begin{cases} X \oplus Y \oplus Z & 0 \le j \le 15 \\ (X \wedge Y) \vee (\bar{X} \wedge Z) & 16 \le j \le 63 \end{cases}$$

$$P_0(X) = X \oplus (X \lll 9) \oplus (X \lll 17)$$

A 256-bit intermediate hash value $V^{(i+1)}$ is calculated after 64 iterations:

$$V^{(i+1)} \leftarrow ABCDEFGH \oplus V^{(i)}$$

The SM3 algorithm iteratively processes all N data blocks, and computes a fixed length data, namely the final 256-bit message digest.

B. Dynamic password generation algorithm based on SM3

Dynamic password token is a hardware used to generate and display the dynamic password. The Dynamic password generation algorithm combines a secret key with the current timestamp using a cryptographic hash function to generate a one-time password. According to the one time password application of cryptography algorithm of China, *K* is a seed key of length not less than 128 bits, and *ID* is a changeable message input of length no less than 128 bits [11].

The dynamic password generation algorithm based on SM3 hash algorithm essentially consists of three steps listed below:

(i) S = SM3(*K*|*ID*), where *S* is the 256-bit output value of SM3 hash function;

(ii) OD=Truncate(S), where *OD* is the 32-bit output of the truncated function and Truncate be a function that selects 32 bits from the result of SM3 in a defined manner;

(iii) P=OD %(10^N), where *N* is the desired number of digits of the one-time password and *N* is no less than 6.

P is the final dynamic password generated by the dynamic password token based on SM3 algorithm.

C. DPA

Side channel attacks can exploit all kinds of information which are unintentionally emitted during the computation of a cryptographic algorithm for detecting the secret stored in the device. Because the amount of power used by a device is influenced by the data being processed, power consumption measurements contain information about calculations of a cryptographic implementation. An attacker can exploit the dependency between the power consumption of a device and the processed data in order to recover secret intermediates, such as keys of cryptographic algorithms.

One of the most widely used side channel attacks is the Differential Power Analysis (DPA). Differential Power Analysis (DPA) computes hypotheses of the power consumption for each input and key candidate and compares them to the recorded power consumption of the device. Such a hypothesis can be computed by calculating the Hamming weight (HW) of a processed value. Finding a suited intermediate value, which reveals information about the key, is specified as leakage analysis. In order to compare the calculated hypothesis to the measured power consumption, methods like the Pearson correlation or the difference in means can be used [10].

III. ATTACKING DYNAMIC PASSWORD TOKEN BAESD ON SM3 ALGORITHM

In this section, we present an attack on dynamic password token based on SM3 algorithm using DPA. Note that this attack allows recovery of the seed key itself. In this paper, we focus on power analysis, especially DPA. However, the attack

is not limited to DPA, other side channel attacks, such as timing attack or electromagnetic analysis, are also applicable.

A. Goal of the attack

In order to explain the attacks, we will state the hypothesis functions used in the conducted DPAs. We assume the following setting: seed key *K* is secret and fixed, and *ID* is public and changeable. Furthermore, we assume that the attacker can measure the power consumption while the device is calculating the one time password. Without loss of generality, we can assume that the size of the seed key and message satisfy |*K*| = |*ID*| = 128. Therefore, the device will run the compression function *CF* only once. We can launch a DPA attack on the message expansion function when the variable *ID* is introduced and combined with *K*. On the other hand, another DPA attack is applicable to the message compression function in order to recover *K*. Consequently, the goal of the attacker is to recover the secret key *K*.

B. Attack on message expansion function

According to the message padding and parsing stage, we know that W_0, W_1, W_2, W_3 make up the 128-bit seed key *K*, and W_4, W_5, W_6, W_7 constitute the 128-bit message input *ID*. Thus, W_0, W_1, W_2, W_3 are fixed and unknown variables, and W_4, W_5, W_6, W_7 are known and changeable.

According to equations (2), the variable$\square W_0^{'}\square$must be calculated in round 0, and the calculation involves W_0 and W_4. Note that W_0 is fixed, W_4 is known and changeable, therefore, a DPA attack is applicable. By selecting $\mathrm{HW}(W_0^{'}) = \mathrm{HW}(W_0 \oplus W_4)$ as hypothesis function and making hypotheses about W_0, we can recover it using a first-order DPA.

In order to recover W_1, W_2, W_3, we can select $\mathrm{HW}(W_1^{'}) = \mathrm{HW}(W_1 \oplus W_5)$, $\mathrm{HW}(W_2^{'}) = \mathrm{HW}(W_2 \oplus W_6)$ and $\mathrm{HW}(W_3^{'}) = \mathrm{HW}(W_3 \oplus W_7)$ as hypothesis function respectively. By making hypotheses about W_1, W_2, W_3 separately, we can recover these three words.

The four 32-bit secret values, i.e. *K* are thus recovered, using four first-order DPA attacks.

C. Attack on message compression function

We use the subscript t, $0 \le t \le 63$ to signify the round number, e.g. A_0 refers to the value of *A* at the beginning of round 0 of the compression function *CF*, etc. At the very start of *CF*, eight 32-bit word registers A, B, C, D, E, F, G, H are initialized by *IV*. So $A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0$ are known and constant values. And the goal of this DPA attack is to recover the four values: W_0, W_1, W_2, W_3. The detailed attack process is described as follows.

1. According to equations (3) to (14), the variable$\square \mathrm{TT1}_0\square$must be calculated in round 0. $\mathrm{TT1}_0$ is a sum with 4 items, and can be rewritten as:

$$\mathrm{TT1}_0 = \theta_0 + \mathrm{W}_0^{'}$$

where

$$\theta_0 = FF_0(A_0, B_0, C_0) + D_0 + SS2_0$$

Note that θ_0 is a fixed and known constant, $W_0^{'}$ is unknown and changeable, therefore, a DPA attack is applicable. By selecting $\mathrm{HW}(TT1_0) = \mathrm{HW}(\theta_0 + (W_0 \oplus W_4))$ as hypothesis function and making hypotheses about W_0, we can recover W_0 and calculate the corresponding values of $TT1_0$.

On the other hand, with reference to equations (10), we can get $A_1 = TT1_0$. Alternatively, another attack strategy is selecting $\mathrm{HD}(A_0, A_1) = \mathrm{HW}(A_0 \oplus TT1_0)$ as hypothesis function and making hypothesis about W_0 . We can recover W_0 in this way as well.

2. According to the above attack, we can compute $TT1_0$, so we know$A_1, B_1, C_1, D_1, E_1, F_1, G_1, H_1$ before round 1. From equations (3) and (14), we know $TT1_1$ must be calculated in round 1. Using the similar method described above, we can make hypothesis about W_1 and then recover it. This allows the attacker to compute $TT1_1$.

3. After attack step 1 and step 2 above, we can compute $TT1_1$, so we know$A_2, B_2, C_2, D_2, E_2, F_2, G_2, H_2$ before round 2. From equations (3) and (14), we know $TT1_2$ must be calculated in round 2. In the same way, we can make hypothesis about W_2 and then recover it. This allows the attacker to compute $TT1_2$.

4. Follow the above attack, we can compute $TT1_2$. Then we know $A_3, B_3, C_3, D_3, E_3, F_3, G_3, H_3$ before round 3. From equations (3) and (14), we know $TT1_3$ must be calculated in round 3. Similarly, we can make hypothesis about W_3 and then recover it.

The four 32-bit secret values, i.e. *K* are thus recovered, using four first-order DPA attacks by attacking the compression function.

Extending the DPA attacks described in section III B and section III C to other length of *K* and *ID* are trivial and need not to be repeated here.

IV. Attack on Software Implementation

A. Testing platform

To examine the proposed DPA attack in section III, we conducted experiments on a software implementation of dynamic password token based on SM3 hash function. Experimental setup consists of a PC, a power tracer, a smart card and a LeCroy oscilloscope. The smart card is an 8-bit 80251 microprocessor with a software implementation of dynamic password generation algorithm based on SM3, and does not include any countermeasures against side channel analysis.

Without loss of generality, we assume that |*K*| = |*ID*| = 128. The working frequency of the smart card is 4MHz and the sampling frequency is 500MHz. Traces for the message expansion and first four rounds of the compression function were captured while 5,000 random messages were being processed. The aligning and re-sampling of the data preprocessing are done after data sampling. Fig.1. represent the traces after processing.

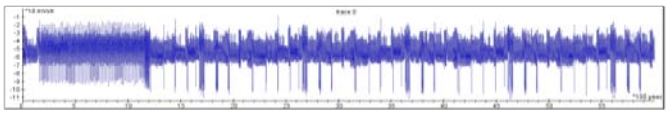

Fig. 1. Traces after processing

B. Attack results

We have realized our attack in Inspector 4.5, including the acquisition module and the analysis module. The power model was built using an 8-bit key guess at a time. This choice was motivated by a practical search space (256 different key guesses). There is a preferred direction for DPA starting from the least significant bit.

Fig.2 and Fig.3 shows the result of our attack on W_0 against the message expansion function. Since the seed key K is 0x1234567890abcdef1234567890abcdef, thus the most significant byte of W_0 is 0x12 and the second most significant byte of W_0 is 0x34. It only takes 5000 traces that 0x12 and 0x34 comes top in the 256 candidate values respectively.

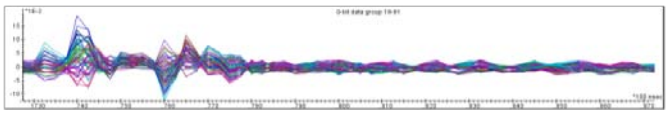

Fig. 2. Result of our attack on the most significant byte of W_0

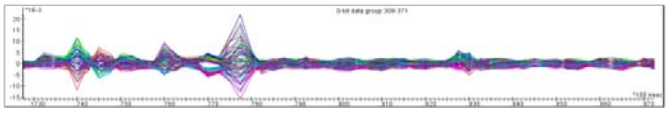

Fig. 3. Result of our attack on the second most significant byte of W_0

The attack result against the message compression function is described below. Fig.4 shows the result of our attack on the least significant byte of W_0. The least significant byte of W_0 is 0x78 and it takes 5000 traces that 0x78 comes top in the 256 candidate values. Fig.5 shows that the second least significant byte of W_0 emerges top in the 256 candidates.

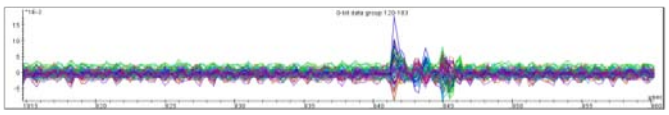

Fig. 4. Result of our attack on the least significant byte of W_0

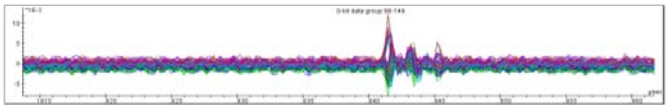

Fig. 5. Result of our attack on the second least significant byte of W_0

V. Conclusions

A first-order DPA attack on dynamic password token based on SM3 hash algorithm was proposed, and the attacks were verified with a software implementation. We have shown that the attacker can recover the seed key by attacking the message expansion function or the message compression function respectively. Further work will focus on designing other forms of side channel attack, such as template attack and chosen-plaint attack. Besides, secure dynamic password token based on SM3 hash algorithm against side channel attacks also needs further research.

Acknowledgment

This work is supported by National Science and Technology Major Project on “Core Electronic Devices, High-end General Chips and Fundamental Software Product” (No. 2014ZX01032401-001) .

References

[1] R.L. Rivest, “The MD5 message-digest algorithm, ”

[2] FIPS 180-1. Secure hash standard, NIST, US Department od Commerce, Washington D.C., Spinger-Verlag, 1996.

[3] X.Y. Wang and H.B. Yu, “How to break MD5 and other hash functions,” Proceedings of the 24th annual international conference on Theory and Applications of Cryptographic Techniques, pp.19-35, 2005.

[4] X.Y. wang, Y.Q. l. Yin, and H.B. Yu, “Finding collisions in the full SHA-1,” Advances in Cryptology - CRYPTO 2005, vol. 3621, pp. 17-36, 2005.

[5] China’s Office of Security Commercial Code Administration: Sepecification of SM3 Cryptographic Hash Function (2010) (in Chinese), http://www.oscca.gov.cn/UpFile/20101222141857786.pdf

[6] K. Lemke, K. Schramm and C. Paar, “DPA on n-bit sized boolean and arithmetic operations and its application to IDEA, RC6,and the HMAC-Construction,” Cryptographic Hardware and Embedded Systems, vol. 3156, pp. 205-218, 2004.

[7] K. Okeya, “Side channel attsacks against HMACs based on block-cipher based hash functions,” Information Security and Privacy, vol. 4058, pp.432-443, 2006.

[8] K. Okeya and T. Iwata, “Side channel attsacks on message authentication codes,” Security and Privacy in Ad-hoc and Sensor Networks, vol. 3813, pp.205-217, 2005.

[9] R. McEvoy, M. Tunstall, C.C. Murphy, and W.P. Marnane, “Differential power analysis of HMAC based on SHA-2, and countermeasures,” Information Security Applications, vol. 4867, pp. 317-332, 2007.

[10] S. Mangard, E. Oswald, and T. Popp, Power anlysis attacks – revealing the secrets of smart cards. Springer, 2007.

[11] China’s Office of Security Commercial Code Administration: One time password application of Cryptographic algorithm (2012) (in Chinese)