SCIENCE CHINA
Information Sciences

• LETTER •

March 2019, Vol. 62 039105:1–039105:3
https://doi.org/10.1007/s11432-018-9488-2

Side channel attack of multiplication in GF(q) – application to secure RSA-CRT

Sen XU[1], Weija WANG[1], Xiangjun LU[1], Zheng GUO[1], Junrong LIU[1] & Dawu GU[2,1*]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China;*
[2]*Shanghai Institute for Advanced Communication and Data Science, Shanghai* 200241*, China*

Received 4 February 2018/Accepted 15 June 2018/Published online 18 October 2018

Citation Xu S, Wang W J, Lu X J, et al. Side channel attack of multiplication in GF(q) – application to secure RSA-CRT. Sci China Inf Sci, 2019, 62(3): 039105, https://doi.org/10.1007/s11432-018-9488-2

Dear editor,

In the field of side-channel attacks (SCAs), the security of both block ciphers and public key cryptosystems is widely studied after the seminal differential power analysis (DPA) [1]. The straightforward implementations of both types of cryptographic algorithms are vulnerable to SCAs. The former attack methods are based on a comparison of actual leakages and a prior leakage model. The typical one is the correlation power analysis (CPA) [2], which is widely used in both practical attacks and security evaluation. Researchers are dedicated to constructing secure RSA-CRT (Chinese reminder theorem) algorithms to mitigate both SCAs. The first step is to construct a secure modular exponentiation counteracting the simple power analysis (SPA). Two categories exist, namely, Shamir's family and the self-secure exponentiation countermeasure. Both categories employ the checksum to counteract the Bellcore attack. The second step is to thwart the CPA. According to the attack principle, two targets need to be blinded. The first one is the multiplication of the recombination phase. The second one is the data transferring of multiplicands of the multiplication. Certain secure algorithms are proposed using the similar principles, such as the algorithms in [3, 4]. However, are these countermeasures secure enough? Herein, we investigate the answer to this question in the presence of the side-channel leakage. More importantly, we provide a comprehensive study on this topic targeting on the implementation with common side-channel protections. Specifically, we focus on the blinding and bit-flipping countermeasure in the multiplication procedure and data transferring, respectively.

Methodology. The leakage model is very important in the fields of SCAs, which describes the leakage method of manipulated vales on a specific hardware platform. The typical CPA-based SCA methods are based on a common leakage model assumption, such as the Hamming weight, bit model, zero value, and Hamming distance. Based on these leakage models, a comparison between models and acquired side channel information can be executed for filtering the incorrect secret key. Generally, we believe that data manipulating on an embedded device causes a noisy observation of the Hamming weight of the intermediate values. For the manipulated value $z \in \mathrm{GF}(2^n)$, the leakage model is assumed to follow the observation of $\mathcal{L}(z)$:

$$\mathcal{L}(z) = \alpha \mathrm{HW}(z) + \varepsilon \tag{1}$$

with an independent noise satisfying $\varepsilon \sim \mathcal{N}(0, \sigma)$, and $\mathrm{HW}(z)$ denotes the Hamming weight of z. $\mathcal{N}$ is the Gaussian distribution. The leakage model is practical in various hardware platforms currently. Two typical methods can be utilized for obtaining

* Corresponding author (email: dwgu@sjtu.edu.cn)

Algorithm 1 Prime byte recovery algorithm

Require: $x^t = \{x^t_{n-1}, x^t_{n-2}, \ldots, x^t_i\}$, where $x^t_i \in \mathcal{I}^t_0$ and $x^t_{i-1} \in \mathcal{I}^t_1$, $p = \{p_{n-1}, p_{n-2}, \ldots, p_{i+1}\}$, previous prime byte set S_{pre} where $p_{i+1} \in S_{\text{pre}}$, result $r^t = \{r^t_{2n-1}, \ldots, r^t_n\}$;
Ensure: S_{p_{i+1},p_i};
1: **for** $t = 0$ to n **do**
2: **for all** $p_{i+1} \in S_{\text{pre}}$ **do**
3: **for** prime $= 0$ to 255 **do**
4: Index $\Leftarrow 1$; ▷ flag
5: $p = \{p_{n-1}, p_{n-2}, \ldots, p_{i+1}, \text{prime}\}$;
6: **for all** $x^t_i \in \mathcal{I}^t_0$ **do**
7: **for all** $x^t_{i-1} \in \mathcal{I}^t_1$ **do**
8: $x^t = \{x^t_{n-1}, x^t_{n-2}, \ldots, x^t_i\}$; ▷ obtain previous input bytes
9: $\{\text{PreviousByte}, \text{CurrentByte}\} = x^t \times p$; ▷ obtain current and previous product result values
10: **if** CurrentByte $\leqslant r^t_{2n-i} - 1$ && PreviousByte $\equiv r^t_{2n-i+1}$ && Index **then**
11: $A[p_{i+1}][\text{prime}] += 1$; ▷ compare intermediate value and r^t, count all possible prime bytes
12: Index $\Leftarrow 0$;
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: **end for**
18: **end for**
19: $S_{p_{i+1},p_i} \Leftarrow \max(A^{\text{prime}}_{p_{i+1}})$. ▷ obtain prime byte results

$\text{HW}(z)$ with leakage $\mathcal{L}(z)$, namely the template and variance methods [5].

Recombination phase of RSA-CRT involves an arithmetic multiplication between two secret big numbers, which includes a fixed prime (as the secret parameter of the RSA-CRT). The phase can be computed with

$$\begin{aligned} S &= \text{CRT}(S_q, S_p) \\ &= (((S_q - S_p) \cdot i_q) \bmod q) \cdot p + S_p, \end{aligned} \quad (2)$$

where S_p, S_q are two exponentiation results corresponding to p and q, respectively. $i_q = p^{-1} \bmod q$. From the partial product, the security of the RSA-CRT relies on the hardness of the hidden multiplier problem over $\text{GF}(q)$ [6].

Definition 1 (Hidden multiplier problem over $\text{GF}(q)$)**.** Let $N = p \times q$, where p and q are two n-byte-long big primes. Let $\ell \in \mathbb{N}$. Given a sequence $(\mathcal{L}^i, \mathcal{R}^i)_{1 \leqslant i \leqslant \ell}$, where $\mathcal{L}^i = \text{LM}(x^i) + \varepsilon_i$, $\varepsilon_i \leftarrow \mathcal{N}(0, \sigma)$, $x^i \leftarrow \text{GF}(q)$ and $\mathcal{R}^i = \text{HB}(x^i \cdot p)$, p is recovered.

$\text{HB}(*)$ denotes half of the most significant bytes of $*$. The leakage $\mathcal{L}^i$ is the side channel leakage in a leakage model $\text{LM}(*)$, which can be the Hamming weight or the generic leakage model. ε is the leakage noise.

Our primary concern is the method to solve the Definition 1 with the practical leakage model. The whole procedure is realized using Algorithm 1. The leakage model indicates a deterministic partition of the input data. The voting procedure can be employed by simple counting as shown in line 11. The counting results indicate that the occurrences of all correct prime bytes reach n. We reduce the carries to two choices, which means that we cannot uniquely confirm the current prime byte, typically $|S_{\text{pre}}| > 1$ under limited n inputs. Intuitively, the computation of the algorithm will be exponentially incremented even if the prime byte candidates remain two. Fortunately, we emphasize that the recovery procedure of p_i can uniquely confirm the previous byte p_{i+1}, as shown in line 19. Subsequently, the divide-and-conquer procedure is completed.

Results and countermeasures. We first present the experimental results of our attack. Subsequently, we describe the countermeasures for high-level secure RSA-CRT countermeasures. As described in Figure 1(a), we simulate various noisy situations with bit-flipping countermeasure. The guess entropy decrease quickly in even in 50% noisy inputs. Figure 1(b) shows the evaluation of a secure implementation with the Hamming weight leakage, which is obtained by both templates and variance methods. The experimental results show that the traditional bit-flipping and blinding multiplication are still vulnerable to our method.

Furthermore, certain secure RSA-CRT algorithms can also be attacked by our method. For example, in the Boscher et al. [4] algorithm, the researchers utilized three CRT computations in attempting to reduce the relationship between the signature of intermediate values, namely S'^1, S'^2, and S'^3, where $S = S'^1 \cdot r$ and r is a random number. However, these splits are not complete and we found $S'^3 = M^{2^l} \bmod pq$, where M is a plain text and l is bit size of the prime. Our attack method still functioned based on $S'^3 = x'^3 \cdot p + S'^3_p$ even with the bit-flipping countermeasures. However, we also claim that the signature split is still the

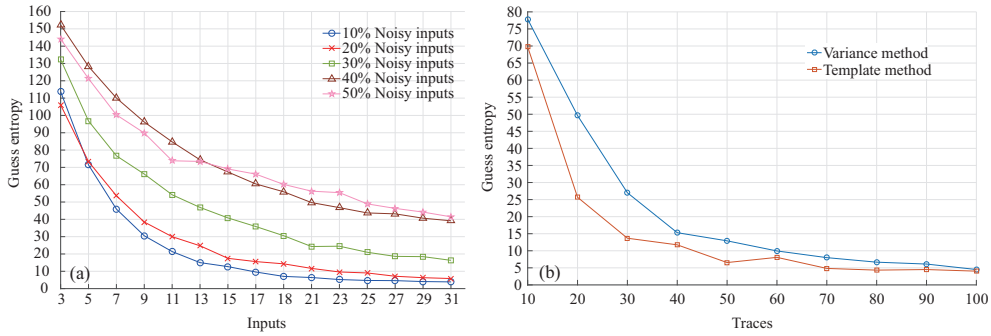

Figure 1 (Color online) (a) Evaluation and (b) practical results of bit-flipping countermeasure with various noisy inputs.

key to thwart both the CPA and our attack.

The key step of both attacks is to utilize the fact that the half of the most significant bytes of the modular exponentiation is exposed to adversary. However, in various secure algorithms, researchers have split the recombination phase into several steps to counteract both Bellcore and CPA. Giraud proposed blinded recombination formulas to prevent Bellcore attacks [7], which provides us with the insight to construct a high-level secure implementation. The blinded recombination multiplies the modulus with a k-bit random prime s. S_q and S_p are the calculated modulo $p \cdot s$ and $q \cdot s$, respectively, and the signature is recombined to $S = M^d \bmod pq$ using the blinded formula:

$$\begin{aligned} S' &= \mathrm{CRT}_{\mathrm{blinded}}(S_p, S_q) \\ &= ((((S_p - S_q) \bmod sp) \cdot i_q \bmod sp) \cdot q + S_q) \\ &= cpq + S, \\ S &= S' \bmod pq, \end{aligned} \tag{3}$$

where c is unknown random number.

In (3), the designers successfully divided the final result S into two steps. The first one is the intermediate signature result S'. Subsequently, the final results can be obtained by $S = S' \bmod pq$ where $S' = cpq + S$. Hence, we cannot utilize half of the most significant bytes for discrimination. Moreover, researchers have attempted to obtain various deterministic one-way functions to blind the computational procedure of S with $F(S') = S$, such as the Kiss et al. [8] and Kim et al. [9] algorithms. The functions render it easy to compute S from S'. It is difficult to reveal S' from S because of the unknown random numbers. These functions prevent both the CPA and our methods.

Conclusion. We investigated various secure RSA-CRT algorithms to evaluate their practical security with the observation of side-channel traces. Our attack method aims to reveal the hidden prime involved in the multiplication of the recombination phase. Blinding and bit-flipping are equipped in the procedure of multiplication and data transferring, respectively. Both countermeasures mitigate the traditional correlation power analysis. Our attack method can still hinder the secure RSA-CRT implementation. We also found that certain secure RSA-CRT algorithms are not secure enough when facing our attack. Eventually, we suggest suitable countermeasures to be employed in the RSA-CRT implementation to protect the recombination phase.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. U1536103, 61402286, 61472249, 61602239, 6157-2192, 61472250), and Minhang District Cooperation Plan (Grant No. 2016MH310).

References

1 Kocher P C, Jaffe J, Jun B. Differential power analysis. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 1999. 15–19
2 Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model. In: Proceedings of International Workshop on Cryptographic Hardware and Embedded Systems, Cambridge, 2004. 16–29
3 Boscher A, Naciri R, Prouff E. CRT RSA algorithm protected against fault attack. In: Proceedings of International Conference on Information Security Theory and Practices, Heraklion, 2007. 229–243
4 Boscher A, Handschuh H, Trichina E. Blinded fault resistant exponentiation revisited. In: Proceedings of Fault Diagnosis and Tolerance in Cryptography, Lausanne, 2010
5 Clavier C, Reynaud L. Improved blind side-channel analysis by exploitation of joint distributions of leakages. In: Proceedings of International Conference on Cryptographic Hardware and Embedded Systems, Taipei, 2017. 24–44
6 Xu S, Lu X J, Zhang K Y, et al. Similar operation template attack on RSA-CRT as a case study. Sci China Inf Sci, 2018, 61: 032111
7 Giraud C. An RSA implementation resistant to fault attacks and to simple power analysis. IEEE Trans Comput, 2006, 55: 1116–1120
8 Kiss Á, Krämer J, Rauzy P. Algorithmic countermeasures against fault attacks and power analysis for RSA-CRT. In: Proceedings of Constructive Side-Channel Analysis and Secure Design, Graz, 2016. 111–129
9 Kim S K, Kim T H, Han D G, et al. An efficient CRT-RSA algorithm secure against power and fault attacks. J Syst Softw, 2011, 84: 1660–1669