Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 3, No. 2, 2023

Divide-and-Conquer Template Attack on Scloud

Yang Zhou*

School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610200, China
* **Corresponding author Email:** 496056734@qq.com

Abstract: Since shor proposed the quantum algorithm, the traditional public key cryptosystem is facing the risk of being breached. To address this issue, Post-Quantum Cryptography (PQC), also known as Anti-Quantum Cryptography, has been developed. Among the various PQC techniques, lattice-based cryptography has emerged as a significant area of research due to its many advantages. This paper studies the leakage of post quantum cryptography cloud on physical devices. Discovered that the matrix calculation operation of the Scloud algorithm has an impact on key security. Based on the characteristics of the curve, we propose a divide and conquer template attack method and introduce convolutional neural network technology for template construction. On the Cortex-M3 board, we have demonstrated that the proposed attack can effectively recover the key.

Keywords: Post quantum cryptography; Side-channel attack; Template attack; Convolutional neural networks.

1. Introduction

With the continuous development of quantum computing technology, the field of traditional cryptography faces unprecedented security threats. In particular, the proposal of Shor's algorithm has proven that quantum computers can decrypt widely-used cryptographic systems such as RSA [1] and elliptic curve cryptography (ECC) within polynomial time. In response to this challenge, researchers have been seeking encryption methods that still possess security in the quantum computing environment for the past few decades, known as post-quantum cryptographic algorithms. Among the various post-quantum cryptographic methods, lattice-based cryptography is considered one of the most promising research directions. There are several schemes for lattice-based cryptographic algorithms, with the main categories being based on Learning with Errors (LWE) [2] and its derivative forms (Ring-LWE [3] and Module-LWE [4]). Their keys are usually composed of several sets of smaller integer vectors, but with significant differences in the methods of operating these vectors for calculations. Ring-LWE and Module-LWE typically rely on Number Theoretic Transform (NTT) for polynomial multiplication calculations, while LWE uses matrix multiplication for calculations.

Scloud is a lattice-based cryptographic algorithm based on the LWE problem, featuring public-key encryption schemes and key encapsulation mechanisms. Scloud relies on the equation B=AS+E, where A, B, S, and E are various matrices in $\mathbb{Z}_q$ with q being a power of 2. The dimensions of these matrices, the modulus q, and the distributions for generating errors E and secrets S are parameters of this scheme.

Although lattice cryptography has theoretically proven its security in the design phase, it may still be susceptible to side-channel attacks when implemented on physical devices. A recent investigation into side-channel attacks in the NIST PQC project has found that post-quantum algorithms have side-channel attack (SCA) [5] vulnerabilities in their physical implementations. Side-channel attacks were proposed by Paul Kocher and have since seen many encryption schemes being broken by SCA. SCA uses physically-measured side-channel information to recover secret information (e.g., encryption keys). Side-channel information includes power consumption, electromagnetic radiation, emitted sounds, and execution time during the operation of encryption devices. Therefore, SCA poses a major threat to the implementation of encryption schemes, especially for embedded device applications.

In this paper, we propose a divide-and-conquer template attack based on Convolutional Neural Networks (CNNs) [6] for the Scloud algorithm. Our goal is to attack three points of interest, corresponding to loading a secret value into a register, the actual $\mathbb{Z}_q$ multiplication, and updating the accumulator with the result. All positions of the S matrix are attacked independently, allowing for easy template creation. We introduce convolutional neural networks to construct the templates, eliminating the need for precise selection of points of interest on the power consumption curve.

2. Preliminaries

2.1. Template Attack

Template Attack [7] is an attack method based on side-channel attacks, which collects a large amount of leaked information during the cryptographic implementation process and uses statistical analysis techniques for attacks. The core of a Template Attack is to establish a relationship between specific information leaked during the encryption process and sensitive information. The classic Template Attack involves building noise templates. This is because, in signal processing, the observed samples are usually modeled as a combination of the intrinsic signal generated by the operation and noise, which can be either intrinsically generated or environmental noise. Although for repeated calls of the operation, the signal part is the same, the noise is best modeled as samples randomly drawn from a noise probability distribution, which depends on the operation and other environmental conditions. For an attacker attempting to find the correct hypothesis given a single sample S, the optimal method is to use the maximum likelihood method, i.e., the best guess is to choose the operation that maximizes the probability of the observed noise in S. Calculating this probability requires the attacker to accurately model the intrinsic signal and noise probability distribution for each operation.

Template Attacks usually consist of two phases: the template building phase and the attack phase. In the template building phase, the attacker first needs to obtain the execution

information of a series of different keys on the same hardware device, such as power consumption, electromagnetic leakage, etc. Through in-depth statistical analysis of this information, the attacker can create characteristic templates for that hardware implementation. These templates contain the relationship between specific operations (such as encryption, decryption, etc.) and the observed side-channel information. In the attack phase, the attacker needs to collect side-channel information generated by the victim's hardware device when executing the target key operation. The attacker then compares this information with the previously built templates to determine which template matches the information generated by the victim's device. In this way, the attacker can gradually recover part or all of the target key information.

2.2. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are neural networks commonly used in fields such as image recognition, computer vision, and natural language processing. They are mainly composed of convolutional layers, pooling layers, and fully connected layers. CNNs were initially designed for image classification. Their main principles are local receptive fields and weight sharing. Local receptive fields refer to the fact that the convolutional layer, when processing input data, only focuses on the local region of the input data and ignores the influence of other regions. The advantage of this approach is that it can reduce the dimensionality of the input data, making the convolutional neural network more efficient. Weight sharing means that in the convolutional layer, the weight parameters in the convolutional kernel are the same for different input data, which can significantly reduce the number of parameters that need to be trained, thus lowering the risk of overfitting.

The main advantage of CNNs is that they can extract high-level features from raw data and perform classification on these features. In the context of template attacks, the role of Convolutional Neural Networks (CNN) is primarily as a classifier. In template attacks, CNNs are commonly used to map the input side-channel leakage data to specific secret key values. The attacker trains the CNN to learn how to extract specific patterns from the side-channel leakage data, which are associated with the values of the secret key. The attacker can use these patterns to identify the secret key values of the algorithm being executed on the target device.

2.3. Scloud

Scloud is a lattice-based cryptographic scheme based on the LWE problem, which includes public-key encryption and key encapsulation mechanisms. Figure 1 describes the encryption process of Scloud. The Initiator first generates public parameters A from a random seed seedA, samples

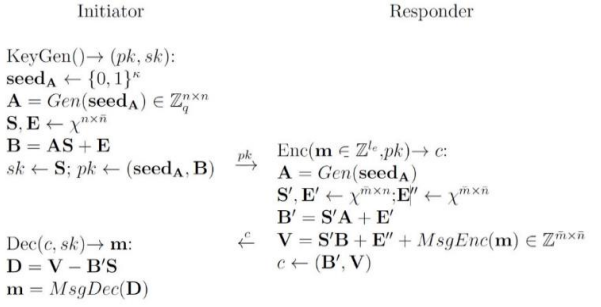

Figure 1. Scloud process

secret error terms (**S** and **E**) from a specific distribution, and calculates the message (**B**, b), which is sent to the receiver along with the seed. Given **B** and **A**, neither a classical computer nor a quantum computer should be able to compute the small error terms S and E, due to the LWE problem. The Responder then achieves information encryption by generating the ciphertext (B', V) using his temporary error samples and sends it to the Initiator.

The core operation of Scloud is computing B←AS + E.

Without loss of generality, we will focus only on a single column of the secret matrix S, denoted as s, in the following content. Therefore, our target operation is b←As + e, trying to recover the small value s from the known A and b based on the leakage of the matrix multiplication As. For a given A and b, the correctness of a guessed s can be checked by examining whether b−As is within the supported range. This is sufficient, as an incorrect s would make the result pseudorandom. Algorithm 1 describes the matrix multiplication of Scloud, where for each iteration of the outer loop, the accumulator sum is initialized to zero and updated n times through an equal number of $\mathbb{Z}_q$ multiplications. This implies that for each secret entry s[i], the attacker can exploit portions of n power traces, i.e., each time it is used in line 5, prompting the use of horizontal attacks.

Algorithm 1 Matrix multiplication in Scloud

1: $\mathbf{b} \leftarrow \mathbf{e}$
2: **for** $i = 1, \dots, n$ **do**
3: $\quad sum \leftarrow 0$
4: $\quad$ **for** $j = 1, \dots, n$ **do**
5: $\quad\quad sum \leftarrow sum + (\mathbf{A}[i,j] \cdot \mathbf{S}[j] \ mod \ q)$
6: $\quad \mathbf{b}[i] \leftarrow (\mathbf{b}[i] + sum) \ mod \ q$
7: **return b**

2.4. Experimental Setup

We implemented the Scloud algorithm on a CortexM3 platform, using the Inspector EM probe for data acquisition and a Lecroy WaveRunner 640Zi oscilloscope with a sampling frequency of 2.5GHz. The collected data was not processed in any way except for alignment. The convolutional neural network environment is based on GPU TensorFlow (TensorFlow gpu 2.8.0). The operating system is Linux, and the GPU is a combination of two NVIDIA GeForce RTX 3080 and four NVDIA GeForce RTX 2080Ti.

3. Divide-and-Conquer Template Attack

Each entry of the subkey s is independently and randomly sampled from the same distribution, so we can attack each position. Therefore, we consider a divide-and-conquer template attack. A significant advantage of this approach is that the total number of templates is relatively small, allowing us to preprocess the analysis data. We use CNN to extract features directly from the entire operation curve.

For each input s, the distinguisher can calculate a distinguishing score vector based on the obtained information, making it easier to determine the correct target value. The probability of correctly identifying the target value by selecting the element with the highest score (corresponding to the maximum a posteriori probability estimate) is referred to as the first-order success rate. Based on the first-order success rate of the subkeys, we use the first-order subkey group success rate to measure the experimental performance, i.e.,

the probability of successfully selecting the highest scoring value as the correct key for each subkey in the key group. Due to the use b – As for verification, a sub key group contains n sub keys and the first-order subkey group success rate has only two possible values, 0 and 1.

We independently attack each position of s, but the results are similar for all positions. Figures 2 and 3 show the training accuracy and loss function.

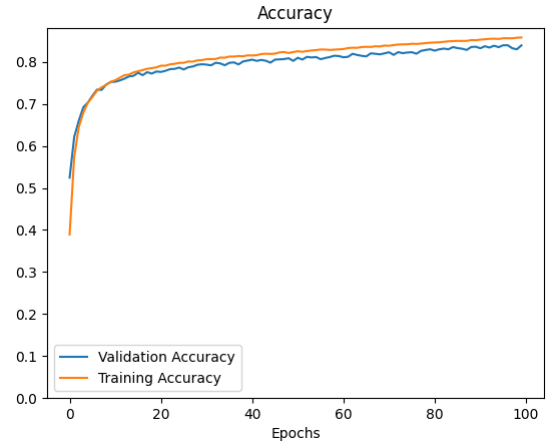

Figure 2. Accuracy rate

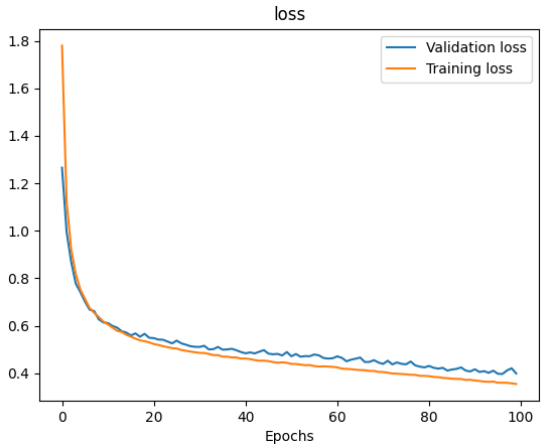

Figure 3. loss function

From the above two figures, it can be observed that the loss values of the training and validation sets gradually decrease during the training process, indicating that the model is continuously optimized during the learning process. Similarly, the accuracy and validation accuracy are steadily improving, further proving that the model's performance gradually improves during the training process. In the later stage of training, the improvement speed of accuracy and loss values slows down, indicating that the model is approaching convergence.

For a single matrix calculation, each position participates in the calculation n times, i.e., there are n curves. In this experiment, the algorithm parameter n = 640, which means there are 640 curves for attacking one position and a subkey group contains 640 subkeys. Under this condition, the first-order subkey group success rate is shown in Figure 4.

From Figure 4, it can be seen that the divide and conquer template attack has shown good results in the practical application of attacking key groups, and can recover most of the key groups. The main reason why this attack strategy is effective is because the row size of the key matrix is large, which determines the number of times a single sub key is repeatedly operated on. The higher the value, the lower the security of the key.

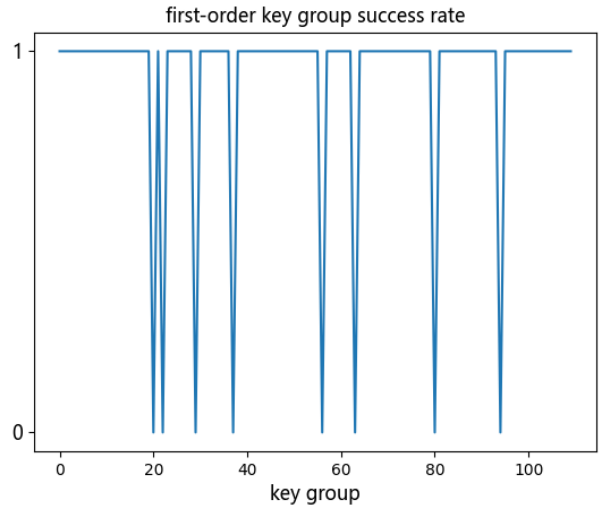

Figure 4. first-order key group success rate

4. Conclusion

Encryption algorithms play a crucial role in large-scale communication protocols. Since these protocols operate with ephemeral secrets, the side-channel analysis of their encryption algorithms may potentially be overlooked. In this paper, we demonstrate that the divide and conquer template attack is a feasible method that exhibits good performance in attacking key groups, which means there is a high probability of recovering complete keys.

References

[1] Rivest, R.L., Shamir, A. and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 1978, pp. 120-126.

[2] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM), 56(6), 2009, pp. 1-40.

[3] Peikert C, Regev O, Stephens-Davidowitz N. Pseudorandomness of ring-LWE for any ring and modulus. InProceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing 2017 Jun 19 (pp. 461-473).

[4] Chunhuan Z, Zhongxiang Z, Xiaoyun W, Guangwu X. Distinguishing LWE Instances Using Fourier Transform: A Refined Framework and its Applications. Cryptology ePrint Archive. 2019.

[5] Erdfelder E, Faul F, Buchner A. GPOWER: A general power analysis program. Behavior research methods, instruments, & computers. 1996, pp. 1-11.

[6] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. ICET 2017, pp. 1-6.

[7] Chari, S., Rao, j. R., & Rohatgi, P. Template attacks. In Cryptographic Hardware and Embedded Systems-CHES. 2002.