

Distributed power analysis attack on SM4 encryption chip

Haoran Gong^{1*} and Tailiang Ju²

^{1*}UESTC,, No. 2006 Xiyuan Avenue, ChengDu, 611730, SiChuan, China.

²UESTC, No. 2006 Xiyuan Avenue, ChengDu, 611730, SiChuan, China.

*Corresponding author(s). E-mail(s): *****;

Contributing authors: *****;

Abstract

Encryption chips are specialized integrated circuits that incorporate encryption algorithms for data encryption and decryption, ensuring data confidentiality and security. In China, the domestic SM4 algorithm is commonly utilized, as opposed to the international AES encryption algorithm. These widely implemented encryption standards have been proven to be difficult to crack through crypt analysis methods. Currently, power consumption side-channel attacks are the most prevalent method. They involve capturing power consumption data during the encryption process and subsequently recovering the encryption key from this data. The two leading methods are Differential Power Analysis (DPA) and machine learning techniques. DPA does not necessitate prior knowledge but relies heavily on the number of power consumption curves. With only 50 power consumption data points, the accuracy is a mere 80%. Machine learning methods require prior knowledge, achieving an accuracy rate above 95% with only 30 power traces, albeit with training times typically exceeding 15 minutes. In this paper, a distributed energy analysis attack approach was presented based on Correlation Power Analysis (CPA). The power consumption data was divided into 16 subsets, with each subset corresponding to 8 bytes of the key. By training each subset separately, the 8-byte key's corresponding power consumption data is reduced to only 100 dimensions, resulting in a 76% decrease in cracking time and a 3% improvement in cracking accuracy rate. This article also trains a more complex 256 classification model to directly crack the final key, achieving a success rate of 28% in cracking 128-bit passwords with only 1 energy trace.

Keywords: SM4 Algorithm; encryption chip; power analysis attack; distributed; machine learning

1 Introduction

Encryption chips play a significant role in today's society, in areas such as financial payments, smart cards, mobile devices, and more. For example, many electronic devices like bank cards, smartphones, and electronic passports use secure chips to store and protect sensitive information. However, despite employing various encryption algorithms to safeguard data security, they are not always secure in practice. The compromise of encryption chips can lead to various harms, including financial losses, personal privacy breaches, security vulnerabilities, and more.

Side-channel attacks and power analysis attacks have become hot topics in the field of encryption chip attacks. Side-channel attacks are a method of attack that can decrypt keys by monitoring the electromagnetic radiation, power consumption, or other physical characteristics generated by encryption chips. However, side-channel attacks are not always feasible because they require significant computation and analysis, along with physical access to the hardware of the encryption chip. With technological advancements, power analysis attacks have also become a potent weapon for decrypting encryption chips. This attack method has seen rapid development over the past few years, and many research findings related to power analysis attacks have been published in international journals and conferences.

In 1999, KOCHER proposed the power side-channel attack method [1], which provided an alternative encryption attack method aside from mathematical analysis. This method revealed the relationship between encryption hardware and encrypted data by using physical information generated during hardware data encryption, such as power consumption, electromagnetic radiation, and time-based data, to decrypt encryption algorithms [2]. Subsequently, power analysis attacks have been widely applied and developed. For example, Chari et al. and Mangard et al. proposed attack methods based on power and electromagnetic radiation analysis, referred to as 'Differential Power Analysis' and 'Differential Electromagnetic Analysis,' respectively. Additionally, with the rapid development of machine learning, especially deep learning [3], there is an increasing number of researchers applying machine learning to side-channel attacks, and its effectiveness far surpasses traditional analytical methods. BACKS [4] et al. applied machine learning to sound side-channel attacks on printers, HOSPODAR [5] et al. classified intermediate values in template attacks using least squares support vector machines, LERMAN [6] et al. used algorithms like random forests, support vector machines, self-organizing maps for side-channel analysis, HEUSER [7] et al. employed multi-class support vector machines for attacking multi-value (Hamming weight models), BARTKEWITZ [8] et al. further improved the aforementioned work, proposing new multi-classification strategies based on categories, and MARTINASEK [9] et al. introduced a neural network-based AES side-channel attack method and classified AES keys.

Unlike the commonly used international Advanced Encryption Standard (AES), the SM4 encryption algorithm is an emerging algorithm and is one of the commercial cryptographic algorithms recommended by the China National Cryptography Administration. The design goal of the SM4 algorithm is to provide high-strength data encryption protection while ensuring sufficient security, efficiency, and flexibility.

The use of the SM4 algorithm is widespread, especially in fields such as mobile payments, the Internet of Things (IoT), and cloud computing. As a cryptographic algorithm developed independently by China, the SM4 algorithm has gained broad international recognition and application. In addition to its extensive use in areas such as finance and government, the SM4 algorithm has also been approved by the International Organization for Standardization (ISO/IEC) as an international standard, receiving widespread international adoption.

Compared to the AES algorithm, the SM4 algorithm is more resistant to attacks such as power analysis. The interaction between the key and plaintext in SM4 is more concealed. Additionally, the key decryption in SM4 is not independent. If the higher-order bits of the key are decrypted incorrectly, it will inevitably lead to errors in the lower-order bits of the key. This characteristic makes the success rate of breaking SM4 consistently lower than that of AES. Moreover, the training models for machine learning in the context of SM4 are also more complex.

The aim of this work is to minimize the decryption time and reduce the number of power traces used, while ensuring a successful decryption rate.

2 SM4 Encryption Algorithm and Hardware Implementation

2.1 SM4 Encryption Algorithm

The SM4 algorithm employs a 128-bit key and has a block key length of 128 bits. It supports five different modes of operation, including ECB, CBC, CFB, and CTR. The algorithm's processes include round key addition, S-Box substitution, linear transformation, key expansion, and inverse operations.

The first step is key expansion, which extends the key MK into 32 round keys rk_i :

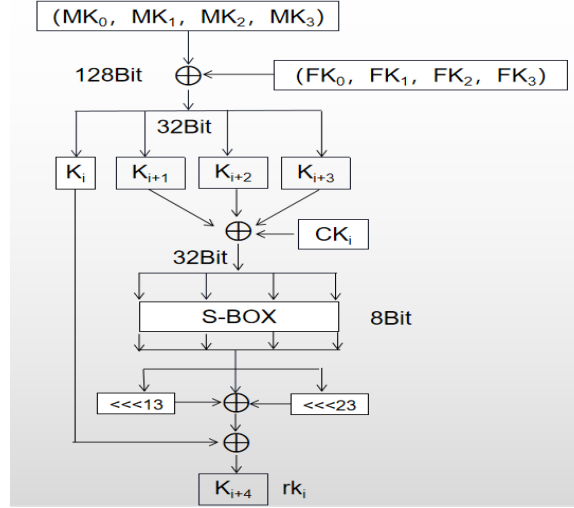


Fig. 1 SM4 Key Expansion

The second part is plaintext encryption:

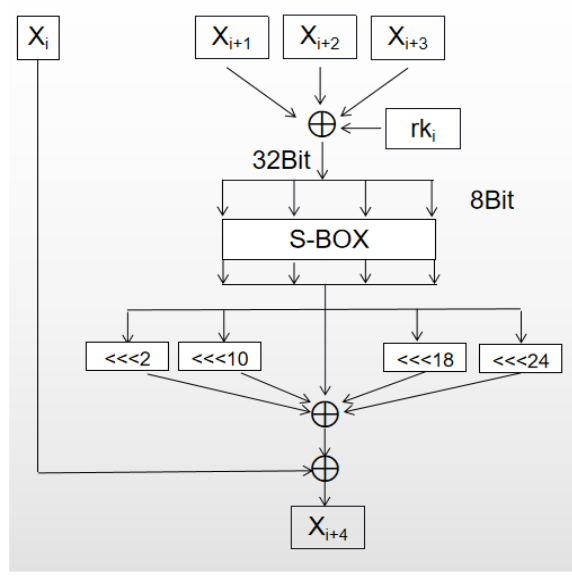


Fig. 2 SM4 Encryption Process

The entire encryption process can be described using a simple formula:

Where the key MK , plaintext PT , and ciphertext CT are all 128 bits, and the key MK_i , X_i , and rk_i are all 32 bits.

$$MK = [MK_0, MK_1, MK_2, MK_3] \quad (1)$$

$$PT = [X_0, X_1, X_2, X_3] \quad (2)$$

$$F_1(MK) = [rk_0, rk_1, rk_2, \dots, rk_{31}] \quad (3)$$

$$\begin{aligned} F_2(X_0, X_1, X_2, X_3, rk_0) &= X_4 \\ F_2(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) &= X_{i+4} \\ &\dots\dots\dots \\ F_2(X_{31}, X_{32}, X_{33}, X_{34}, rk_{31}) &= X_{35} \end{aligned} \quad (4)$$

$$CT = [X_{35}, X_{34}, X_{33}, X_{31}] \quad (5)$$

2.2 Hardware Implementation of SM4 Encryption

The chip chosen for this project is the Atmel Xmega-128D4, and the target board used is the ChipWhisperer CW308 UFO board.

ChipWhisperer is a company specialized in providing side-channel attack tools and training. CW308 is one of their produced target boards used for side-channel attack experiments.

The CW308 target board features a 50 MHz XMEGA microcontroller and provides various peripheral interfaces, such as a high-speed ADC (12-bit, 105 MSPS), programmable clock (100 MHz - 500 MHz), and a USB interface for connecting to the host computer. This board can be used in conjunction with the ChipWhisperer Lite or Pro USB analyzer for the analysis and attack of side-channels in embedded systems.

The hardware implementation steps for the SM4 encryption algorithm are as follows:

- (1) Microchip Studio is selected as the development environment, and the C++ code for SM4 encryption is initially written.
- (2) The project's main program, serial communication program, and driver programs are completed and compiled.
- (3) The generated HEX file is burned into the Xmega128D4 chip.

3 Collection of SM4 Encryption Power Traces

Choosing the ChipWhisperer CW1200 Power Consumption Acquisition Platform

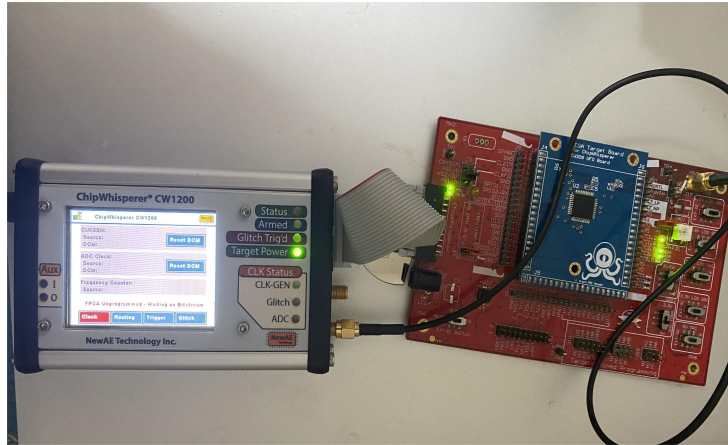


Fig. 3 Power Consumption Acquisition Platform

In this study, the number of sampling points is set at 24,400, with an analog-to-digital converter offset of 1,250, and triggering on the rising edge. Two different power consumption curves are collected:

- (1) 500 power consumption traces with a fixed key and random plaintext, used for decryption.

(2) 10,000 power consumption traces with random key and random plaintext, used for training in machine learning methods.

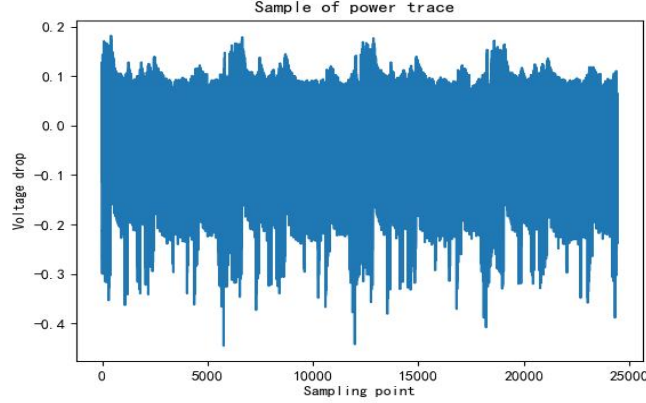


Fig. 4 Examples of Power Traces (Power Consumption Curves)

4 Power Analysis Attack

4.1 Correlation Power Analysis Attack (CPA)

4.1.1 Attack Principle

The effectiveness of side-channel attack depends on the selection of attack points within the cryptographic algorithm and the corresponding energy model.

In the case of the SM4 encryption algorithm, during each round iteration, the input for each round is XORed with the round key for that round, followed by passing through an S-box. The S-box is a non-linear transformation that generates significant power consumption during the transformation, as compared to linear transformations. Therefore, choosing this point as an intermediate value for power analysis attacks makes it easier to break the key. This value, denoted as V_{atk}^i , can be represented as follows:

$$V_{atk}^i = HW(Sbox(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)) \quad (6)$$

* $HW()$ converts numbers into Hamming weight.

4.1.2 Cracking of the Round Key

$$rk_0 = [rk_0[0], rk_0[1], rk_0[2], rk_0[3]] \quad (7)$$

Starting with the cracking of $rk_0[0]$, proceed to crack every 8 bits:

(1) With n identical keys and different plaintexts, compute the correlation coefficient between the power trace samples and the intermediate value Hamming weight, as expressed by the following formula:

P_{jk} represents the value of the k -th sample point of the j -th power trace, and P_k represents the value vector of all power traces at time k :

$$P_k = [P_{0k}, P_{1k}, P_{2k}, \dots, P_{nk}] \quad (8)$$

(2) For each power trace, calculate V for $m \in (0, 256)$:

$$PT^j = [X_0^j, X_1^j, X_2^j, X_3^j] \quad (9)$$

PT^j represents the plaintext of the j -th power trace.

$$X_l^j = [X_l^j[0], X_l^j[1], X_l^j[2], X_l^j[3]] \quad (10)$$

$$V_{rk_0[0]=m}^j = HW(Sbox(X_1^j[0] \oplus X_2^j[0] \oplus X_3^j[0] \oplus m)) \quad (11)$$

$$V_{rk_0[0]=m} = ([V^0, V^1, \dots, V^n] | rk_0[0] = m) \quad (12)$$

(2) Calculate the correlation coefficient $Cor(k, m)$.

$$Cor(k, m) = Cor(P_k, V_{rk_0[0]=m}) \quad (13)$$

Where P_k and $V_{rk_0[0]=m}$ are both n -dimensional vectors, and their correlation coefficient calculation formula is:

$$Cor(P, V) = \frac{\sum_{i=1}^n (P_i - \bar{P})(V_i - \bar{V})}{\sqrt{\sum_{i=1}^n (P_i - \bar{P})^2} \sqrt{\sum_{i=1}^n (V_i - \bar{V})^2}} \quad (14)$$

Iterate through sample points $k \in (0, 24400)$ and $m \in (0, 256)$, searching for the maximum point. At this point:

$$MAX(Cor(P_k, V_{rk_0[0]=m})) = Cor(P_{pos}, V_{rk_0[0]=key}) \quad (15)$$

Take key as the cracked value for $rk_0[0]$, it appears at the pos -th sample point in the power trace. Use this method to sequentially crack $rk_0[1]$, $rk_0[2]$, $rk_0[3]$, and obtain the complete rk_0 .

(3) Iterate 3 times to crack rk_1 , rk_2 , and rk_3 .

$$F_2(X_0, X_1, X_2, X_3, rk_0) = X_4 \quad (16)$$

$$V_{atk}^1 = Sbox(X_2 \oplus X_3 \oplus X_4 \oplus rk_1) \quad (17)$$

Equation 17 is used to crack rk_1 . In total, 4 rounds of cracking are performed to obtain rk_0 , rk_1 , rk_2 , and rk_3 , from which the original key is reconstructed.

4.1.3 recovery of key

Given rk_0 , rk_1 , rk_2 , rk_3 , with CK_i as fixed parameters, and L' representing a constant linear transformation within the encryption, the formula yields:

$$K_i = rk_i \oplus L'(Sbox(rk_{i-3} \oplus rk_{i-2} \oplus rk_{i-1} \oplus CK_i)) \quad (18)$$

Subsequently, the SM4 key MK is reconstructed from K_0 , K_1 , K_2 , K_3 , and FK_i (fixed parameters), following this computation method:

$$MK_i = K_i \oplus FK_i \quad (19)$$

$$MK = [MK_0, MK_1, MK_2, MK_3] \quad (20)$$

At this point, the initial key is fully recovered.

```
Connected to pydev debugger (build 213.7172.26)
the cracked round key 0 is: [141, 85, 208, 90]
the cracked round key 1 is: [202, 144, 7, 169]
the cracked round key 2 is: [195, 215, 93, 7]
the cracked round key 3 is: [91, 82, 165, 149]
the right key is: [59, 126, 21, 22, 40, 174, 210, 166]
[171, 247, 21, 136, 9, 207, 79, 60]
the cracked key is: [59, 126, 21, 22, 40, 174, 210, 166]
[171, 247, 21, 136, 9, 207, 79, 60]
Cracking succeeded
Process finished with exit code 0
```

Fig. 5 Successful CPA Attack

4.1.4 The Weaknesses of CPA Attack

(1) CPA attacks depend on the number of power traces to be cracked. In the experiments, the number of power traces was gradually reduced, and multiple inputs were used. The success rate of CPA attacks on the SM4 encryption chip is as follows:

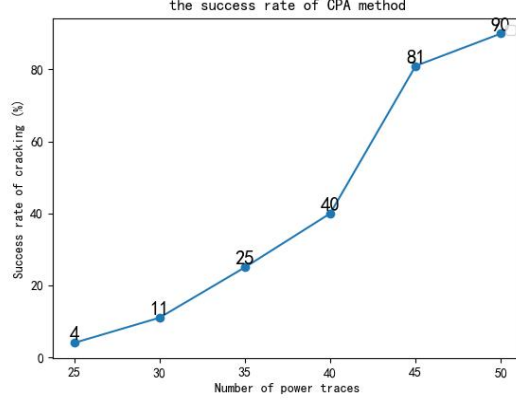


Fig. 6 Cracking success rate of CPA method

(2) The CPA method relies on clock alignment, and it cannot crack the key when constant asynchrony is introduced.

(3) The CPA method cannot crack the key when random masking is applied.

4.2 Machine learning-based power analysis attack

4.2.1 Attack Principle

For each power trace, there is a corresponding attack intermediate value. If multiple power traces are obtained with random plaintext and random keys, and their corresponding intermediate values are calculated, with power traces represented as $\vec{X}$ and the Hamming weight of the corresponding intermediate value as Y , we can use machine learning methods to train a model:

$$f(\vec{X}) = Y \quad (21)$$

Then, the trained model is applied to the target power traces for key recovery [10].

4.2.2 Attack steps

(1) PCA (Principal Component Analysis) Dimensionality Reduction

The original power traces consist of 24,400 sampling points. Regardless of the training mode, a dataset with 24,400 dimensions would require an impractically long training time. The core idea of PCA is as follows: the principal components of a

matrix are the eigenvectors of its covariance matrix, sorted by their corresponding eigenvalues. PCA reduces a set of potentially correlated high-dimensional variables into a set of lower-dimensional, linearly uncorrelated variables known as principal components. These lower-dimensional data components aim to retain as much of the original data's variance as possible. Without delving into specific details, the PCA algorithm can be applied to achieve data dimensionality reduction through the use of PCA API calls.

(2) Machine Learning-Based Key Recovery

Using machine learning methods, a corresponding Y is obtained for each power trace.

$$\begin{aligned} f(\vec{X}_i) &= Y_i \\ \vec{Y} &= [Y_0, Y_1, \dots, Y_n] \end{aligned} \quad (22)$$

Taking the attack on $rk_0[0]$ as an example, iterate through $rk_0[0] \in (0, 256)$, corresponding to the i -th power trace, with the intermediate value:

$$\begin{aligned} (V_i | rk_0[0] = m) &= Y_i^m \\ Y^{\vec{m}} &= [Y_0^m, Y_1^m, \dots, Y_n^m] \end{aligned} \quad (23)$$

Compare each bit of all $Y^{\vec{m}}$ with $\vec{Y}$ one by one, and determine the value of m that makes the most identical bits as the cracked key.

4.2.3 Attack Performance

We trained with 10,000 power traces and used 30 power traces for key cracking. We employed three different methods: SVM, LSTM, and CNN. Multiple experiments were conducted, and the results are recorded as follows:

Table 1 Cracking Speed and Success Rate of Various Machine Learning Methods

Dims(PCA)	2000	1600	1200	800	400
SVM(S)	968	831	782	704	646
LSTM(S)	1261	1140	1036	920	840
CNN(S)	1128	1045	898	832	785
SVM(%)	95	90	72	66	51
LSTM(%)	97	95	78	70	65
CNN(%)	97	92	75	73	57

¹(s) represents the cracking time of the method

²(%) represents the success rate of the cracking

From Table 1, it can be observed that as the dimensionality increases, the training time becomes longer, and the success rate improves. Conversely, with fewer dimensions, training time is reduced, but the success rate decreases.

Analyzing the underlying reasons, excessive dimensionality reduction results in data loss. While it may improve speed, it leads to a decrease in success rate. PCA dimensionality reduction employs the same approach for cracking each round key, which makes it unable to capture the specific sampling points corresponding to each round key, preventing precise matching.

5 Distributed Power Analysis Attack

5.1 Attack Principle

Distributed power attack is based on the correlation coefficient between each 8-bit round key and the power trace, which samples the power trace to generate 16 sub-power traces. This reduces the dimensionality of the data, thereby improving the cracking efficiency. At the same time, each sub-power trace is more targeted and less susceptible to interference, which can increase the success rate.

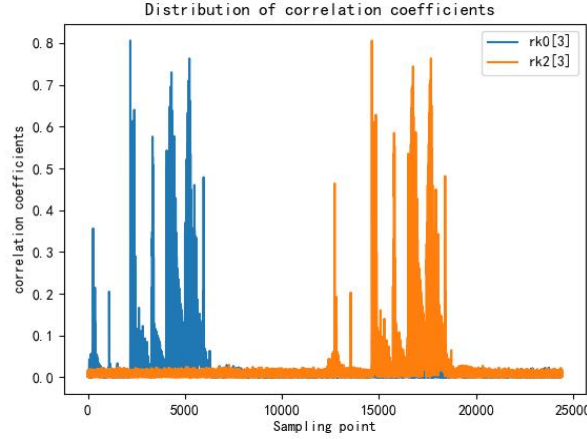


Fig. 7 The correlation coefficient of rk0[3] and rk2[3]

From Figure 7, it can be observed that different round keys manifest at different positions on the power traces. By considering the magnitude of the correlation coefficients, it is possible to extract the sampling points from the power traces for each round key.

5.2 Attack Performance

In this study, we selected the top 100 points with the highest correlation coefficients for extraction, forming 16 sub-traces. We then trained them separately using machine learning-based methods and conducted multiple experiments, with the results as follows:

Table 2 Success Rates of Various Machine Learning Methods

Power trace	30	25	20	15	10
SVM(%)	96	93	87	79	72
LSTM(%)	98	95	90	85	77
CNN(%)	96	92	84	71	65

Table 3 Processing Times of Various Machine Learning Methods

SVM	LSTM	CNN
190S	370S	235s

From Tables 2 and Tables 3, it can be observed that the three machine learning methods have similar accuracy. When using 10 power traces, the success rate is approximately around 70%. In terms of training time, SVM is the fastest, while LSTM takes the longest.

5.3 Attacking of masked SM4 encryption chip

In engineering, it is common to incorporate masking techniques into the encryption process to counteract side-channel attacks, such as power analysis attacks. Compared to standard encryption methods, masking involves operations like Galois multiply or XOR with intermediate values within the encryption process.

The location at which the sub-plaintext and the round key first interact during the SM4 encryption process:

$$V = X_1 \oplus X_2 \oplus X_3 \oplus rk_0 \quad (24)$$

If the mask is added before this point, it is treated as the key, and the method mentioned earlier is used to find the intermediate value, and the mask is attacked. If the mask is added after this point, the round keys are cracked first, and the entire mask and key are cracked through multiple iterations using this method.

In this paper, an attempt was made to perform XOR operations with a fixed mask at each S-box output, and using this method, all the keys and masks were successfully cracked.

6 Cracking key from one power trace

6.1 Attack Principle

Building on the previous work, we shifted from classifying based on Hamming weight to classifying directly based on the key, turning a 9-class problem into a 256-class problem. Due to this finer classification, there was a noticeable decrease in classification accuracy, with the accuracy for each 8-bit key dropping from 98

(1)As observed from Figure 7, every 8-byte key is related to approximately 6000 sample points in the power trace. We chose the dimensionality of the features to be 6000.

(2)To preserve the temporal characteristics, we selected a continuous window of 6000 sample points with the highest correlation coefficient.

(3)We employed a 1D CNN+LSTM model for training.

These optimizations were implemented to improve the classification accuracy.

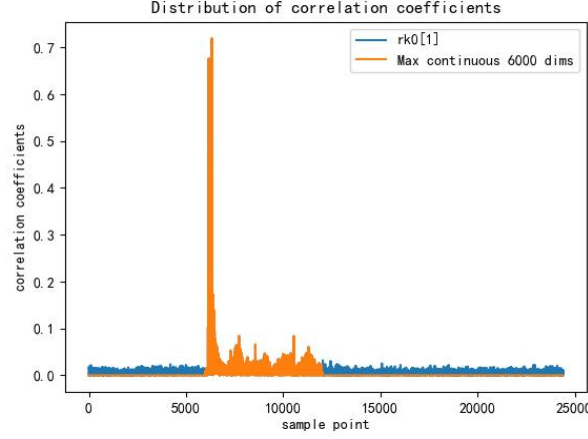


Fig. 8 Max continuous 6000 dims

6.2 Model Establishment

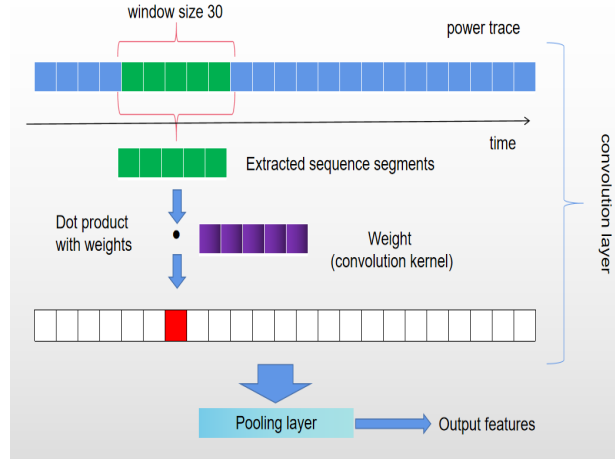


Fig. 9 One-dim CNN

After multiple rounds of convolution and pooling, the data dimension is reduced from 6000 to 1000, and then a LSTM network is used for 256-class classification. The layer sizes are as follows: 1000→800→600→400→256.

6.3 Attack Performance

Due to the larger size of the model, training times often reach up to 5 hours (without GPU). The classification success rate for 8-bit keys averaged at 92%. In the case of only 1 power trace available, the probability of successfully recovering all 16 bits reached 28%. With just 3 power traces available, the success rate for key recovery reached 45%.

```
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 63ms/step
1/1 [=====] - 0s 48ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 63ms/step
1/1 [=====] - 0s 64ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 47ms/step
1/1 [=====] - 0s 47ms/step
cracked 4 round key : [[204, 85, 208, 90], [221, 155, 133, 73], [69, 96, 128, 9], [67, 98, 214, 123]]
The cracked key is [122, 126, 21, 22, 40, 174, 210, 166, 171, 247, 21, 136, 9, 207, 79, 60]
The right key is: [122, 126, 21, 22, 40, 174, 210, 166, 171, 247, 21, 136, 9, 207, 79, 60]
Cracking success
Process finished with exit code 0
```

Fig. 10 Cracking key from one power trace

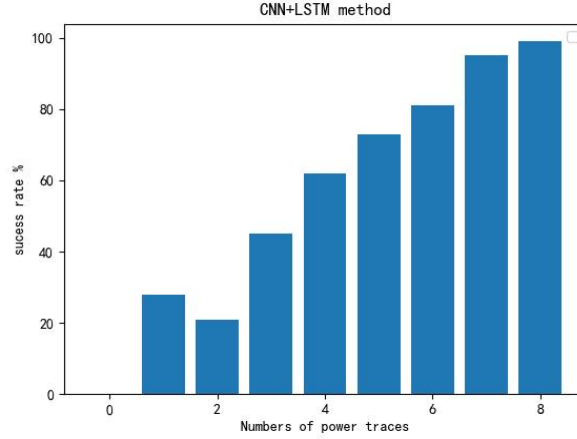


Fig. 11 The success rate

From the figure, it can be observed that since it's based on individual selections where each trace produces an independent result, having 2 traces doesn't offer any improvement compared to just 1 trace. It randomly selects one result among them. When there are 8 traces available, the success rate for decryption reaches 99%.

7 Conclusion

This paper has accomplished three main parts: the hardware implementation of the SM4 encryption algorithm, the collection of power traces, and the decryption of keys using power analysis attacks. In terms of key decryption, both common CPA methods and machine learning-based approaches have been employed. Additionally, a distributed decryption method has been utilized to enhance decryption speed and accuracy. With only 10 power traces, the success rate of decryption has been increased to over 70%, and the decryption time has been reduced by 76%. By using a 256-classification model, the success rate of decrypting a 128-bit key has reached 28% with only one power trace. Furthermore, this study has successfully broken the SM4 encryption chip with simple masking.

In real-world scenarios, encryption chips may perform other operations simultaneously during encryption, introducing noise that makes it difficult to capture power leakage, thereby increasing the difficulty of decryption. Additionally, the experiments in this paper heavily rely on clock alignment. Therefore, future research should focus more on noise filtering techniques to make power analysis attacks effective even in more complex situations.

appendix:

SM4 power traces download address:

<https://drive.google.com/file/d/1dbUEeosEiT0eZ-31x-SC0aDxxT2UbQ-p/view?usp=sharing>

References

- [1] Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19, pp. 388–397 (1999). Springer
- [2] Quisquater, J.-J., Samyde, D.: Electromagnetic analysis (ema): Measures and counter-measures for smart cards. In: Smart Card Programming and Security: International Conference on Research in Smart Cards, E-smart 2001 Cannes, France, September 19–21, 2001 Proceedings, pp. 200–210 (2001). Springer
- [3] Learning, D.: Ian goodfellow, yoshua bengio, aaron courville. The reference book for deep learning models **1** (2016)

- [4] Backes, M., Dürmuth, M., Gerling, S., Pinkal, M., Sporleder, C., *et al.*: Acoustic {Side-Channel} attacks on printers. In: 19th USENIX Security Symposium (USENIX Security 10) (2010)
- [5] Hospodar, G., Gierlichs, B., De Mulder, E., Verbauwhede, I., Vandewalle, J.: Machine learning in side-channel analysis: a first study. *Journal of Cryptographic Engineering* **1**(4), 293–302 (2011)
- [6] Lerman, L., Bontempi, G., Markowitch, O.: Side channel attack: an approach based on machine learning. *Center for Advanced Security Research Darmstadt* **29** (2011)
- [7] Heuser, A., Zohner, M.: Intelligent machine homicide: Breaking cryptographic devices using support vector machines. In: *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pp. 249–264 (2012). Springer
- [8] Bartkewitz, T., Lemke-Rust, K.: Efficient template attacks based on probabilistic multi-class support vector machines. In: *Smart Card Research and Advanced Applications: 11th International Conference, CARDIS 2012, Graz, Austria, November 28-30, 2012, Revised Selected Papers 11*, pp. 263–276 (2013). Springer
- [9] Martinasek, Z., Zeman, V.: Innovative method of the power analysis. *Radioengineering* **22**(2), 586–594 (2013)
- [10] Lerman, L., Bontempi, G., Markowitch, O.: Power analysis attack: an approach based on machine learning. *International Journal of Applied Cryptography* **3**(2), 97–115 (2014)