







Research of CPA Attack Methods Based on Ant Colony Algorithm

Xiaoyi Duan¹ , You Li¹ , Jianmin Tong¹ , Xiuying Li¹, Siman He² ,
and Peishu Zhang¹

¹ Beijing Electronic Science and Technology Institute, Beijing, China

² Hunan National Secrecy Science and Technology Evaluation Center, Hunan, China

Abstract. The Power analysis attack is an effective method of attacking encryption devices for leakage of side-channel information. CPA (Correlation Power Analysis) is a common method. The traditional method of Power Analysis Attack, which is only one-byte key, is analyzed in one attack and repeats multiple operations to obtain the whole secret key. In this way, a successful attack needs more power curves. In this paper, a new attack method is proposed to select the optimal secret key group through the Ant Colony Algorithm and attack all the bytes of the secret key simultaneously. It can greatly eliminate the influence of the channel noise and improve the efficiency of the attack. To prove the effectiveness of this new method, the AES algorithm as an example is implemented on the MEGA16 microcontroller. The power consumption curve of the AES algorithm with a fixed secret key and random plaintext is collected, and the power consumption is analyzed separately by the original method and the new method. As a result, the success rate of the original method is only 10.981% when using 4000 power curves; however, the new one is up to 100%, which is increased by 89.019%. When the power curves do not exceed 3000, the success rate of the original method is zero. However, the success rate of the new method can reach 34.375% even if only 1500 power curves are used. The new method is more effective than the original one. Being affected by parameters, the attack time of the new method is not consistent but much less than the original method.

Keywords: Power analysis attack · CPA (Correlation Power Analysis) · AES algorithm · Ant Colony Algorithm

1 Introduction

Power analysis attack is a method of attacking encryption devices for leakage of side-channel information such as time consumption, power consumption, or electromagnetic radiation during these devices works [1]. This new type of attack is much more effective than the mathematical method of cryptanalysis, thus posing a serious threat to cryptographic devices. In recent years, with the popularization of varieties of cryptographic chips and embedded devices, power analysis attack brings more damage to system security. Thus power analysis attack and their corresponding countermeasure have become

popular research fields in the world at present. CPA (Correlation Power Analysis) is less affected by the noise, and attackers using CPA do not need to know the detail about the attacked device. It has become one of the main methods of Power Analysis Attack for domestic and foreign scholars.

At present, the research of energy analysis attack focuses on the relationship between the energy change of a byte in the chip and the hypothesis model. In 2009, Massimo et al. [2] designed a general multi-bit power consumption model for a precharged circuit based on the characteristics of symmetric algorithm and the structure of the processor. This model has high accuracy for the precharged circuit. In 2013, Oswald research found that after linear transformation of power consumption curve, it can accurately quantify the impact of a linear filter on power analysis attack so as to effectively select the optimal linear filter to improve attack efficiency [3]. Satoh R et al. proposed a new power analysis attack method that can be used to improve the efficiency of the resistance evaluation of cryptographic LSI. The proposed method performs power analysis not in the conventional time-domain but the frequency domain [4]. In 2014, Kim et al. found that due to the existence of noise information in the power consumption curve, the attack efficiency will be reduced. They proposed a principal component analysis method based on the original data to raise the idea of correlation coefficient analysis. This method first sought the principal component of the original data and then selected the power consumption curve with good quality according to the principal component to attack the power consumption analysis. The efficiency of this method is higher than that of ordinary methods. The attack has been greatly improved [5]. In paper [6] explores the use of machine learning techniques to perform a power analysis attack and to deal with high dimensional feature vectors. In this paper [7] investigate the vulnerability of SIMON and LED lightweight block cyphers against Differential Power Analysis (DPA) attack. In 2015, Pozo Applied singular spectrum analysis (SSA) to power analysis attacks to improve the signal-to-noise ratio of signals and attack efficiency [8], aiming at the problem that low sampling rate will affect the signal analysis. L Guo et al. proposed a differential power analysis attack on dynamic password token based on the SM3 algorithm [9]. This paper [10] present a review of the power analysis attack and its techniques. Also, a brief detail on some of the power analysis attacks on smart card and FPGA has been presented. In 2016, L Guo et al. proposed a chosen-plaintext differential power analysis attack on HMAC-SM3 [11]. Masoumiet et al. proposed a practical smart card implementation of an advanced encryption standard (AES-128) algorithm combined with a simple yet effective masking scheme to protect it against first-order power analysis attacks in both time and frequency domain [12]. The paper [13] proposes a method for performing power analysis attacks against SIMECK. In 2017, Eleonora et al. Proposed an end-to-end attack method based on a convolutional neural network for the problem of power curve dislocation, which can effectively achieve the attack without realigning the power curve in advance [14]. Chakraborty A et al. proposed a generic Correlation Power Analysis (CPA) attack strategy against STT-MRAM based cryptographic designs using a new power model [15]. In 2018, Wiemers A et al. based on a theoretical analysis on quantifying the remaining entropy, deriving a practical search algorithm. Which even in a setting with high noise or few available traces can either successfully recover the full AES key or reduce its entropy significantly [16]. In 2019, Kim et al. [17] introduced a

method to analyze side channels using convolutional neural networks. The paper [18] used a Convolutional Neural Network (CNN) to attack the algorithm implementation on the single-chip computer with mask and interference defence. In 2020, Cai X et al. proposed an energy trace compression method for differential power analysis attack [19]. Xiaoyi Duan et al. proposed data enhancement to solve Hamming weight imbalance of Sbox output values in machine learning [20].

The traditional method of Power Analysis Attack, which is only one-byte key, is analyzed in one attack and repeats multiple operations to obtain the whole secret key. In this way, a successful attack needs more power curves. The new CPA attack method proposed in this paper is an innovative application of the Ant Colony Algorithm. This approach takes full advantage of the leaking of the whole 8 S-boxes, significantly reducing the number of power curves and increasing the success rate of attack. As a result, the success rate of the original method is only 10.981% when using 4000 power curves; however, the new one is up to 100%, which is increased by 89.019%. When the power curves do not exceed 3000, the success rate of the original method is zero. However, the success rate of the new method can reach 34.375% even if only 1500 power curves are used. When the power curves are limited in number or greatly impacted by noise, the original attack method may be difficult to succeed. The new method, however, can efficiently use the limited power curves to analyze and increase the possibility of obtaining the correct secret key.

The structure of this paper is as follows: In Sect. 2, relevant research at home and abroad is introduced in brief. In Sect. 3, some concepts are discussed, such as AES, CPA method, Ant Colony Algorithm and the new attack method proposed by us. In Sect. 4, an attack experiment is implemented separately by the original method and the new method. Also, the results of the experiment are analyzed. In Sect. 5, the new attack method and its efficiency are summarized.

2 Background Knowledge

2.1 Introduction of AES Algorithm

With the rapid development of computing power, the DES algorithm is not enough to ensure information security and is gradually replaced by the AES algorithm. AES algorithm is a symmetric encryption algorithm used by the American National Standards Institute and was officially adopted in business in 2001.

As shown in Table 1, the block size of the AES algorithm is 128 bits, and the key length can be 128, 192 or 256, which determines the number of encryption rounds. In this paper, the 128-bit AES algorithm is introduced briefly, and it is attacked to verify the effectiveness of the new attack method proposed by us.

AES algorithm is a block cypher algorithm based on iterative computation, in which the data stream is encrypted or decrypted with 128 bits as a block. AES uses a substitution/permutation network, called SP structure, to carry out round operation iteratively. Before data encryption, the input data was divided into blocks. Each block has four words (32 bits). Each word contains 4 bytes, and 8 bits consist of a bit.

In general, plain text is described in terms of square matrices in bytes, called state matrices. In each round of the algorithm, the contents of the state matrix continually

Table 1. AES key length and the number of Encryption rounds

	Key length (32 bit)	Block size (32 bit)	Encryption rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

transmit changes, and the result is output as ciphertext. Similarly, a 128-bit key is also a matrix represented in bytes, called the key matrix. The key matrix is expanded into a sequence of 44 words through the key scheduling program- $\omega[0], \omega[1], \dots, \omega[43]$, the first four elements of which are $\omega[0], \omega[1], \omega[2], \omega[3]$ as the original key to plus the initial key; The last 40 words are divided into ten groups, each of 4 words (128 bit) respectively for ten rounds of round-keys plus computing.

AES does not use the Feistel network. The whole block is processed in each round instead of a half. So AES decryption process is not consistent with its encryption process.

During encryption, plaintext and original key are encrypted once before the first iteration of the AES algorithm. In the last round of iteration, no column mixed. The decryption operation is the inverse operation of the encryption operation. Therefore, the round key addition operation is performed once before the first round of decryption, and

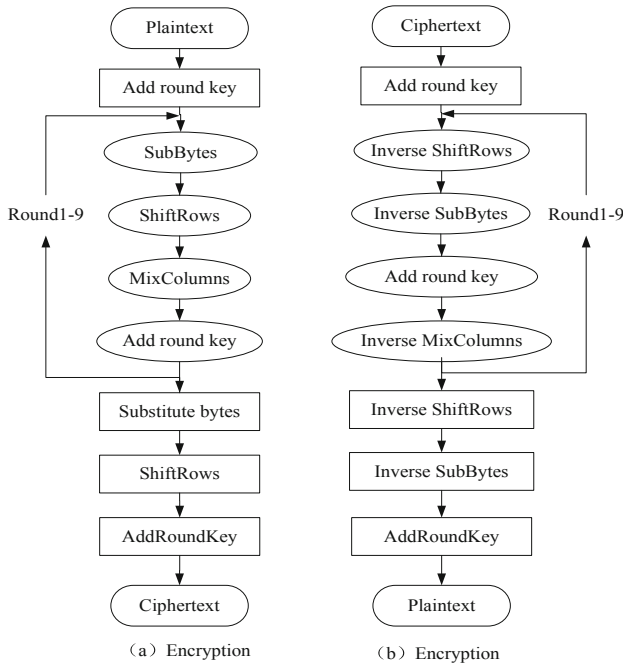


Fig. 1. Flow chart of the AES-128 algorithm

the reverse column mixing is not performed in the last round. The algorithm flow chart is shown in Fig. 1.

Before encryption and decryption, AES divides the plaintext and the key into several bytes and rearranges them to form a matrix. The basic processing steps include SubBytes, ShiftRows, MixColumns, and AddRoundKey. To increase the effectiveness of AES, the calculations are done in finite fields in SubBytes and MixColumns. AddRoundKey uses the operation of bitwise XOR. ShiftRows uses the operation of cyclic shift. Though SubBytes must perform the multiplicative inverse over finite fields, the operation can be simplified by using a look-up table. MixColumns reflects the validity of AES. It makes the bytes of plaintext and key fully mixed, which increases the difficulty of attacking. Combining with the operating characteristics of the 9-round transform of AES, our energy attack mainly focused on the first round, then extracted the voltage of the AES cryptographic chip and recorded 10,000 valid data.

2.2 Introduction of CPA

Unlike Simple Power Analysis (SPA), Differential Power Analysis (DPA) attack does not analyze the power consumption of the device along the time axis. However, it analyzes the linear dependence of the power consumption and the processed data at fixed times. DPA attack can analyze the key information from the small differential signal of the power curve. However, it needs to collect a large amount of information and collect multiple sets of the power curve and record the plaintexts and ciphertexts of each power curve. Usually, it needs Some SPA analysis experience and longer time to operation, and the high requirements of the platform equipment. Using the differential mathematical statistics method, PDA poses a severe challenge to the security of the cryptographic chip. It has become the focus of many researchers at home and abroad.

Correlation Power Analysis (CPA) is an extension of Kocher's classical differential power attacks, and it was proposed by Brier et al. in 2004. Selecting an unknown but constant reference state, CPA establishes a Hamming model and analyses the coefficient of correlation between power consumption sample and Hamming-weight of the processed data. The main idea is that when the attacker knows the plaintext, he can change the plaintext and collect the corresponding power curve. Specific steps are as follows:

- 1) The oscilloscope is used to collect the power consumption data of the encryption chip being performing encryption or decryption (In this paper, the output voltage value of S-box in the first round of AES is collected) to obtain a power consumption matrix T of $M \times N$.

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,M} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,M} \end{bmatrix} \quad (1)$$

Wherein, M rows of the matrix T represent M different sampling values of power curve, and N columns represent N power curves. Each power curve has the same key but different plaintext.

- 2) The attacker guesses the secret key of the encryption chip and uses plaintext and guess the key to calculate the hypothetical intermediate value matrix $X_{N \times 256}$ according to the formula (2).

$$\begin{cases} X = [X_{i,j}]_{N \times 256} \\ X_{i,j} = SBOX(m_i, k_j) \end{cases} \quad (2)$$

$i = 1, 2, \dots, N; j = 1, 2, \dots, 256$

Wherein, $k = (k_1, k_2, \dots, k_{256})$ contains the entire guess key of one byte. $m = (m_1, m_2, \dots, m_N)$ is the plaintext of the N power curves and $SBOX(m_i, k_j)$ is the operation function of SubBytes in AES.

- 3) Using the Hamming Weight model, the hypothetical intermediate value matrix is mapped to a hypothetical power matrix H .

Therein, HW is a function to calculate the Hamming Weight (the number of logic 1).

- 4) Use Eq. (3) to calculate the correlation coefficient between the hypothetical power matrix H and the measured power matrix T obtained from the collected power curves.

$$\begin{cases} \rho = [\rho_{i,j}]_{256 \times M} \\ \rho(H_i, T_j) = \frac{E(H \times T) - E(H) \times E(T)}{\sqrt{Var(H) \times Var(T)}} \end{cases} \quad (3)$$

$i = 1, 2, \dots, 256; j = 1, 2, \dots, M$

Therein, $E(H)$ and $E(T)$ are the mathematical expectation of the column vectors H_i and T_j . $Var(H)$ and $Var(T)$ are the covariance of H_i and T_j . i is the number of all possible key of one byte. M is the number of sampling points per power curve.

- 5) Find the highest point of the correlation coefficient, and regard the corresponding guess key as the correct key of this byte.

In the CPA attack, calculating the correlation coefficient is a very important part to be distinguished from the other side-channel attack methods. A correlation coefficient is also known as the Person correlation coefficient called ρ . It has no unit, and its value range is $[-1, 1]$. A positive value represents a positive correlation. A negative value represents a negative correlation. The absolute value of 1 means completely related, and one of 0 means irrelevant. CPA analyzes the correlation coefficient between Hamming Weight and power consumption of the intermediate variable and determines the linear relationship between column hi ($i = 1, 2, \dots, K$) and column tj ($j = 1, 2, \dots, T$) by the correlation coefficient. The result is the estimated correlation coefficient in matrix R . The highest correlation coefficient is corresponding to the correct key. Otherwise, the intermediate variable and power consumption have not the expected relationship of direct

proportional due to the wrong key. The estimate for each ri, j is based on D elements of column hi and column tj .

The theoretical premise of the CPA is that the power consumption of the encryption chip with a precharge bus is proportional to the number of “1” been processed. However, there is usually no precharge bus in the encryption chip. According to the principle of CMOS (Complementary Metal-Oxide Semiconductor Transistor), the power consumption is proportional to the frequency of 0/1 conversions (Hamming Distance) instead of the number of “1” been processed (Hamming Weight). Therefore, existing CPA methods cannot obtain the correlation coefficient between the high-power consumption and the success key. In the case of large noise, the correlation coefficient between the correct key and the wrong one may be almost the same. Therefore, the correlation coefficient analysis methods still need to be improved.

2.3 Introduction of Ant Colony Algorithm

During the study of ant foraging, Italian scholar Dorigo et al. found that the ants can release chemical called pheromone in the path they passed through. The ants can walk along the path of higher concentration of pheromone, and every passing ant will leave pheromone on the road, which forms a positive feedback mechanism. After a period, the entire ant colony can reach the food source along the shortest path. Inspired by this behavior, Dorigo proposed an evolutionary algorithm called Ant Colony Algorithm. It is a new heuristic optimization algorithm and has distributed computation, information positive feedback and heuristic searchability. In essence, the ant colony algorithm is a new heuristic optimization method. Its flow chart is shown in Fig. 2.

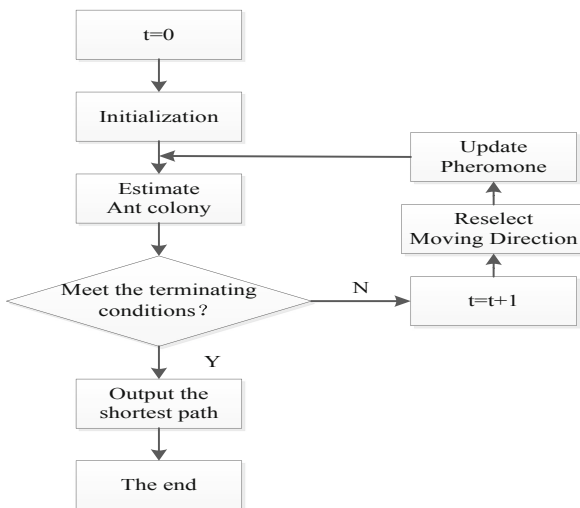


Fig. 2. Flow chart of Ant Colony Algorithm

Initially, the pheromone on each path is equal and it is set as $\tau_{ij}(0) = C$ (C is a constant). During the movement, ant $k(k = 1, 2, \dots, m)$ can decide the transfer

direction according to the pheromone in the road. The rule of state transfer using by ant colony is called the random proportional rule. When located at a node i , an ant k uses the pheromone trail to compute the probability of choosing j as the next node. At time t , the probability $P_{ij}^k(t)$ is:

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t)\eta_{is}^\beta(t)}, & j \in allowed_k \\ 0, & otherwise \end{cases} \quad (4)$$

Therein, $allowed_k = \{0, 1, \dots, n - 1\}$ means the next node that the ant k can choose. According to formula (4), the transfer probability $P_{ij}^k(t)$ is directly proportional to $\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t)$. η_{ij} is the visibility factor. α and β are the two parameters respectively reflect the relative weights of accumulated information and heuristic information in the ants' path selection.

After the ant completes its tour, the pheromone amount on each path will be adjusted according to the formula (5) and formula (6).

$$\tau_{ij}(t + 1) = \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t, t + 1) \quad (5)$$

$$\Delta \tau_{ij}(t, t + 1) = \sum_{k=1}^m \Delta \tau_{ij}^k(t, t + 1) \quad (6)$$

Therein, $\Delta \tau_{ij}^k(t, t + 1)$ is the amount of pheromone left in the path (i, j) by the ant k now of $(t, t + 1)$. Its value depends on the performance of the ants. The shorter the path, the more pheromone is released. $\Delta \tau_{ij}(t, t + 1)$ is the pheromone increment of path (i, j) in this cycle. $(1 - \rho)$ is the pheromone decay parameter. Usually, $\rho < 1$ is set to avoid the infinite accumulation of the pheromone amount on the path.

Ant colony algorithms can be applied to many optimization problems. Ant's walking path represents the feasible solution of the problem to be optimized, and all the paths of the ant colony form the solution space. In the optimal path, the ants release more pheromones, and in the inferior path, the ants release less. With time, the concentration of pheromone accumulated on the optimal path and the number of ants choosing this path is also increasing. In the end, the whole ant colony will concentrate on the optimal path, which is the optimal solution under the positive feedback. Energy Analysis Attack can also be considered as an optimization problem. A possible key can be regarded as a feasible solution, and the collection of all possible keys (2^{128} in total) constitutes the solution space of the optimization problem. The key with a higher correlation coefficient corresponds to a better path, and the one with a lower correlation coefficient corresponds to a poorer path. The final optimal solution is the correct key.

3 Introduction to the New Method

In order to generate the guessed key by Ant Colony Algorithm, Firstly, a map must be established between the guessed key and the ants' path. As shown in Fig. 3, each bit

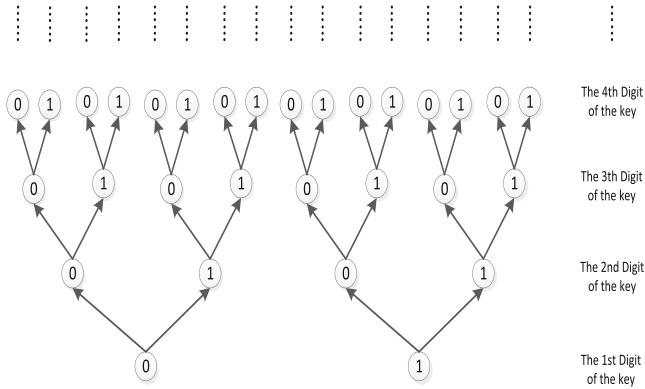


Fig. 3. The map between the guessed key and the ants' path

of the guessed key can be regarded as a node of the path, and each complete ant path corresponds to a guessed key.

After mapping between the guess key and the ants' path, it is necessary to initialize the ant colony algorithm. Specific parameters settings are shown in Experimental Data and Analysis.

After initialization, the evaluation way of ants' route needs to be modified. The ant colony algorithm was originally used to find the shortest path between two points, and the evaluation indicator of the pros and cons of the path is the length of it. If the ant colony algorithm is used in the CPA attack, it is necessary to replace the indicator by the probability of the guessed key being the real key. This can be judged by the correlation coefficient. The higher the correlation coefficient is, the better the path is, and the lower the correlation coefficient is, the worse the path is.

In each round of iteration of the ant colony algorithm, an optimal path will be found, that is, the guessed key with the largest correlation coefficient. It can generate the ant path of the next round iteration according to the best path in this round. The specific rules are as follows:

- 1) Using the correlation coefficient of this optimal path and the other paths of this round, the relative error value can be found respectively according to the formula (7). Therein, Δ is the absolute error and L is the true value (the correlation coefficient of the optimal path in this round).

$$\delta = \frac{\Delta}{L} \cdot 100\% \tag{7}$$

- 2) The relative error value can determine the search range of the ant colony in the next round of iteration. If the relative error value δ is less than 20%, then the iteration path of the next round will be generated in the vicinity of this path with a certain probability. If the relative error value δ is greater than or equal to 20%, then the iteration path of next round will be generated in the global scope with a certain probability. The probability of choosing which road at each branch depends on the

concentration of pheromones on that road. In other words, the concentration of pheromone will affect each value (0 or 1) of the guess key value.

- 3) Finally, the termination condition of the ant colony algorithm needs to be set. Since the solution space of the real key is too large, we cannot and do not need to successively verify all the solutions in the entire solution space. Therefore, we regard the number of iterations as the termination condition of the algorithm. When the pre-set iteration round is completed, the shortest path is output, and its corresponding guess key is the correct key calculated by the algorithm. The ant colony algorithm may tend to converge and achieve the correct key before the pre-set iteration round is accomplished. If the pre-set iteration round is too small, the ant colony algorithm may not converge, and the correct key achieved is not the true key. If the pre-set iteration round is too large, the ant colony algorithm tends to converge, but the correct key found in the optimal local solution rather than the optimal global solution. The calculated correct key is not the true key. As for the choice of iteration rounds, we will discuss in detail in Experimental Data and Analysis.

4 Experimental Data and Analysis

4.1 Experimental Environment

To obtain reliable experimental data, a power analysis platform is established, which consists of three parts, encryption equipment, data acquisition and data analysis. Atmel ATMEGA16A microcontroller is selected as an encryption device, and its clock frequency is set to 4 MHz. For the data acquisition part, the Tektronix DPO7104 oscilloscope was selected in this experiment, and its sampling frequency is set to 50 MHz. In order to ensure the validity of the experimental data, a total of eight sets of data are collected for repeated experiments in this experiment. Each group of data collects 10,000 fixed-key and random plaintext power consumption curves, and each curve has 25,000 power consumption sampling points. The fixed key of each group of data is different. We choose MATLAB to implement CPA, in which the guessed key is generated by the Ant Colony Algorithm, and the attack target is the output bytes of S-box in the first round of AES-128.

4.2 Analysis of Experimental Data

4.2.1 Experimental Performance

There are many indicators to measure the efficiency of attack, such as the success rate of attack, the number of power curves, the amount of computation, computation time and so on. Among them, the amount of computation is inconvenient for statistics and comparison, and the success rate of attack is strongly correlated with the number of power curves. Therefore, we choose the success rate of attack as the main indicator of attack efficiency. Also, computation time is related to many factors, and we will analyze and compare separately.

First of all, we keep the related parameters of the Ant Colony Algorithm unchanged and use different numbers of power curves for CPA attacks. The success rate of attack with the original method and the new method is separately recorded, as shown in Table 2 and Fig. 4.

Table 2. CPA attack success rates for different numbers of power curves

The number of power curves	Success rate	
	Original method (%)	New method (%)
1500	0	34.375
2000	0	57.031
2500	0	69.531
3000	0	86.719
3500	3.775	96.094
4000	10.981	100
4500	18.032	100
5000	89.213	100
5500	98.072	100
6000	100	100

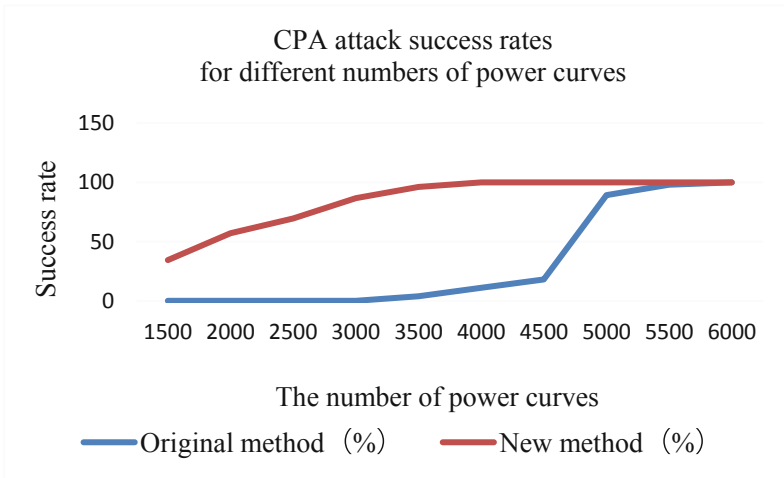


Fig. 4. CPA attack success rates for different numbers of power curves

Because the collected power curve is greatly disturbed by noise, and no denoising is performed, the noise interference in the analysis result is relatively large. It can be seen from Table 2 that the increase in the numbers of power curves has a positive effect on

the improvement of attack success rate. CPA attacks using the original method have a success rate of 0 when the number of power curves is less than 3000. With the increase of the numbers of power curves, the success rate of CPA increases. When the number of power curves reaches 6000, the attack success rate of the original method reaches 100%. However, the success rate of the new method can reach 34.375% even if only 1500 power curves are used. When the number of power curves is up to 4000, the attack success rate of the original method can reach 100%. In order to ensure a successful attack, the power curves of the new method are about 33.33% less than those of the original method. We conclude that the new method can greatly reduce the number of power curves required for a successful attack. Also, when the number of power curves is less than 5000, the attack success rate of the new method is far greater than that of the original method. When the number reaches 4500, the success rate of the new method is still 81.968% higher than that of the original method. It can be inferred that the new method can achieve higher success rates with fewer power curves.

It is important to reduce the number of curves needed to successfully attack. For example, in some restricted situations, only a certain number of power curves can be collected. The original method maybe with a low success rate or even completely unsuccessful. However, the new method can efficiently use the limited power curves and analyze to increase the probability of obtaining the correct key. Another possibility is that the collected power curves may be of poor quality and greatly affected by noise, due to environmental constraints. Only enough power curves collected can reduce the effect of noise. As the number of power curves drastically increases, the entire analysis process of power attack can last if dozens of days and the data storage space occupied can become very large. If the data utilization can be improved, and the number of power curves can be substantially reduced, a considerable amount of analysis and processing time and data storage space will be saved.

When using the original method, it takes about one minute to verify each byte of a guess key, and completing the computation of all the 16 bytes may cost 16 min to 4096 min (68 h and 16 min) in theory. While using the new method of CPA attack, the same power curves of 4000 can accomplish the attack within 4 h. If you ignore the data loading time, it needs less than 2 h (The real-time depends on the specific parameters of the Ant Colony Algorithm, which will be discussed later) for computation itself, saving 97% of the computation time. As mentioned earlier, the number of power curves is positively related to the success rate, and more power curves are usually used to ensure a higher attack success rate. For the original method, the number of power curves has nothing to do with the computation time. Even if the number of power curves is increased, the computation time will not be increased. For the new method, with the increase of the numbers of power curves, the attack success rate gradually increases, but the time needed for the attack also increases. It can be seen from the experimental results that even though the time needed for the attack increases, the increase is not big, and the computation time is still far less than the time required by the original method; furthermore the attack success rate is always far higher than that of the original method. Therefore, the price is acceptable. Also, when the number of power curves reaches 4000, the success rate of attack can increase to 100%. So we can use no more than 4,000 curves to attack, which can maintain the success rate of 100% and avoid unnecessary time-consuming.

4.2.2 Analysis of Experimental Results

The traditional energy analysis attack method CPA only analyzes the problem of one byte key in one attack. This attack method only uses part of the collected energy curve, that is, the energy change caused by only one byte key, and does not make full use of the energy characteristics of the collected 16 bytes. This paper uses the characteristics of ant colony algorithm to propose a CPA attack method based on ant colony algorithm. This method can make full use of all 16 S-boxes of AES algorithm to leak information and maximize the utilization rate of leaked information. Through ant colony algorithm, the optimal key group in energy analysis attack can be selected and all key bytes can be attacked at the same time, and it can eliminate the influence of channel noise to a great extent. To sum up, as the experimental results show, the new method is far superior to the original method in attack success rate, the number of power consumption curves and operation time, which fully verifies the effectiveness of the new method.

4.2.3 Influencing Factors

Before using the CPA attack based on Ant Colony Algorithm, the parameters of the Ant Colony Algorithm need to be set, including the number of ants *ant*, the number of ants' movements *times*, pheromone volatile coefficient *rou*, transfer probability constant *P0* and search range. In the experiment, using the characteristics of distributed computing of the Ant Colony Algorithm, we split the key of 128 bits into 16 bytes, each of which is 8 bits. The computation time is reduced because of parallel computing, so the search area is 0–255. Also, the pheromone volatility coefficient *rou* is set to 1 and the transition probability constant *P0* is set to 0.1. The number of ants *ant* and the number of ants' movements *times* also affect the experimental results.

As mentioned above, one of the main application directions of the new method is to improve the attack success rate as much as possible under the premise of the limited number of power curves. Therefore, we limit the number of power curves to 3000 in discussing the success rate of different parameters in the Ant Colony Algorithm. Then we keep the other parameters unchanged and record the success rate of the CPA attack and the average computation time with different numbers of ants *ant* and ants' movements *times*. The experimental results are shown in Table 3, Table 4, Fig. 5 and Fig. 6.

There is a positive correlation between the number of ants and the success rate of CPA attacks. When the number of ants reaches 25, the increase of attack success rate becomes more slowly than before. When the number is up to 150, the success rate reaches a maximum of 84.375%.

Table 3. CPA attack success rate with different numbers of ants

Ant	Success rate(%)	Average computation time(s)
5	34.375	14.3312
10	57.031	19.0709
15	70.313	23.6315
20	72.656	28.8801
25	79.688	34.0064
30	80.469	38.3656
35	81.250	42.0212
40	81.250	45.8054
45	82.813	49.8914
50	83.594	60.7187
100	83.594	120.7354
150	84.375	205.3296

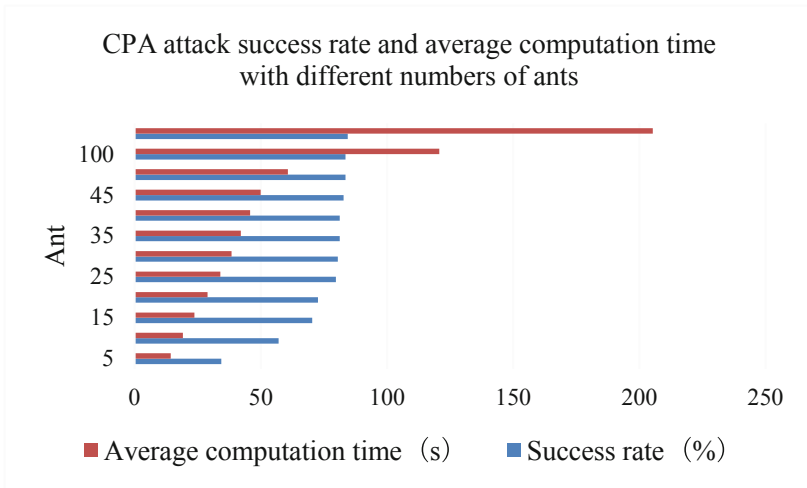


Fig. 5. CPA attack success rate with different numbers of ants

Simultaneously, there is a positive correlation between the number of ants and the average computation time. When the number of ants increases by 5, the average computation time averagely increases by 6.5862 s. Since the increase in the number of ants does not increase the cost of time much, it is a good choice to appropriately increase the number of ants, which can enhance the success rate of attack.

Table 4. CPA attack success rate with a different number of times

Times	Success rate (%)	Average computation time (s)
10	82.813	201.7058
20	74.219	233.8375
30	83.594	258.8970
40	83.594	289.9357
50	86.719	327.9992
60	83.594	354.6122
70	83.594	380.5585

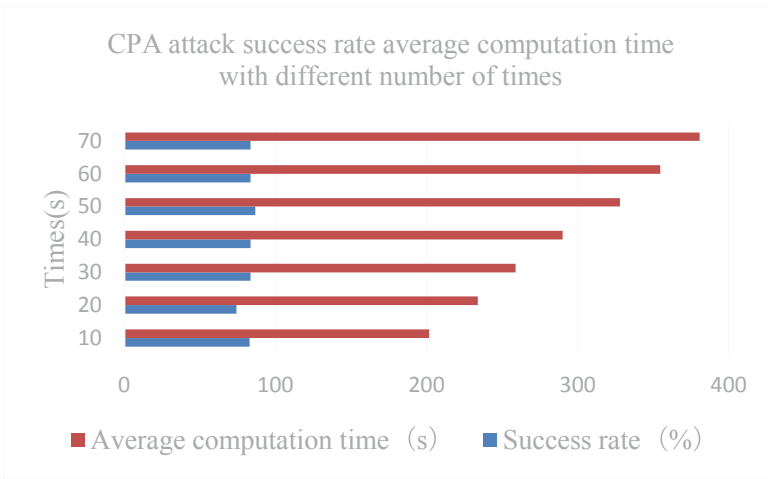


Fig. 6. CPA attack success rate with a different number of times

As mentioned before, the relationship between the number of ants’ movement (iteration rounds) and the success rate of attack is not linear. When the number of ants’ movements is 50, the success rate of attack is the highest, reaching 86.719%. Using this as a benchmark, the success rate of attack will decrease no matter how ants’ movement is increased or decreased. In this experiment, we can conclude that the optimal number of ants’ movements of the Ant Colony Algorithm is 50.

To ensure that the experimental results are not affected by the number of ants, it is set to 200, leading to a longer average computation time. Also, the average computation time is positively related to the number of ants moving. While the number of ant movement increase by 10, the average computation time averagely increase by 25.5504 s. Since the increase in the number of ants’ movement does not necessarily increase the success rate of attack, we do not recommend setting the ants number too large. It is more appropriate to repeat the experiments until the ant’s number is adjusted to an optimal value.

5 Conclusion

In this paper, a new CPA attack method based on Ant Colony Algorithm is proposed. Based on the traditional CPA attack, this method combines the advantages of the Ant Colony Algorithm, including distributed computation, information positive feedback and heuristic searchability. It can fast search the optimal solution (the correct key) in the whole situation to improve the efficiency of the CPA attack. We believe the new method takes full advantage of all the eight S-boxes leakage and maximizes the utilization of leakage information. Thus, it significantly reduces the number of power curves needed for the attack and increases the success rate of the attack with limited power curves. To verify this, we compared the attack results of the new method and the original method by the attack experiment to the AES algorithm. The experimental results show that the new method far surpasses the original one in terms of the success rate of attack, the number of power curves required for attack, and computation time etc., which fully validates the validity of the new method. During the experiment, we found that some parameters of the ant colony algorithm, such as the number of ants and the number of ants' movements, will have an impact on the attack efficiency. We briefly discussed this and gave some suggestions on choosing the best parameters. The new CPA attack method proposed in this paper is universal, so it has a certain value for power analysis technology.

Acknowledgments. This research was supported by the High-tech discipline construction funds of China (No. 20210032Z0401, No. 20210033Z0402) and the open project of Key Laboratory of cryptography and information security in Guangxi, China (No. GCIS201912).

References

1. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
2. Alioto, M., Poli, M., Rocchi, S.: Differential power analysis attacks to precharged buses: a general analysis for symmetric-key cryptographic algorithms. *Dependab. Secure Comput. IEEE Trans.* **7**(3), 226–239 (2009)
3. Oswald, D., Paar, C.: Improving side-channel analysis with optimal linear transforms. In: Mangard, S. (ed.) CARDIS 2012. LNCS, vol. 7771, pp. 219–233. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37288-9_15
4. Satoh, R., Matsushima, D., Shiozaki, M., et al.: Subkey driven hybrid power analysis attack in frequency domain against cryptographic LSIs and its evaluation. *IEEJ Trans. Electron. Inf. Syst.* **133**(7), 1322–1330 (2013)
5. Kim, Y., Ko, H.: Using principal component analysis for practical biasing of power traces to improve power analysis attacks. In: Lee, H.-S., Han, D.-G. (eds.) ICISC 2013. LNCS, vol. 8565, pp. 109–120. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12160-4_7
6. Lerman, L., Bontempi, G., Markowitch, O.: Power analysis attack: an approach based on machine learning. *Int. J. Appl. Cryptogr.* **3**(2), 97–115 (2014)
7. Shanmugam, D., Selvam, R., Annadurai, S.: Differential power analysis attack on SIMON and LED block ciphers. In: Chakraborty, R.S., Matyas, V., Schaumont, P. (eds.) SPACE 2014. LNCS, vol. 8804, pp. 110–125. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12060-7_8

8. Merino Del Pozo, S., Standaert, F.-X.: Blind source separation from single measurements using singular spectrum analysis. In: Güneysu, T., Handschuh, H. (eds.) CHES 2015. LNCS, vol. 9293, pp. 42–59. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48324-4_3
9. Guo, L., Li, Q., Wang, L., et al.: A differential power analysis attack on dynamic password token based on SM3 algorithm. International Conference on Information Science & Electronic Technology (2015)
10. Mahanta, H.J., Azad, A.K., Khan, A.K.: Power analysis attack: a vulnerability to smart card security. In: International Conference on Signal Processing & Communication Engineering Systems. IEEE (2015)
11. Guo, L., Wang, L., Liu, D., et al.: A chosen - plaintext differential power analysis attack on HMAC - SM3. In: 2015 11th International Conference on Computational Intelligence and Security (CIS). IEEE (2016)
12. Masoumi, M., Habibi, P., Dehghan, A., Jadidi, M., Yousefi, L.: Efficient implementation of power analysis attack resistant advanced encryption standard algorithm on side-channel attack standard evaluation board. Int. J. Internet Technol. Secur. Trans. **6**(3), 203 (2016). <https://doi.org/10.1504/IJITST.2016.080392>
13. Yoshikawa, M., Nozaki, Y., Asahi, K.: Multiple rounds aware power analysis attack for a lightweight cipher SIMECK. In: IEEE Second International Conference on Big Data Computing Service & Applications. IEEE (2016)
14. Cagli, E., Dumas, C., Prouff, E.: Convolutional neural networks with data augmentation against Jitter-based countermeasures. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 45–68. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_3
15. Chakraborty, A., Mondal, A., Srivastava, A.: Correlation power analysis attack against STT-MRAM based cyptosystems. In: IEEE International Symposium on Hardware Oriented Security & Trust. IEEE (2017)
16. Wiemers, A., Klein, D.: Entropy reduction for the correlation-enhanced power analysis collision attack. In: Proceedings of the 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, 3–5 Sep 2018 (2018)
17. Kim, J., Picek, S., Heuser, A., Bhasin, S., Hanjalic, A.: Make some noise unleashing the power of convolutional neural networks for profiled side-channel analysis. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2019**(3), 148–179 (2019). 430
18. Benadjila, R., Prouff, E., Strullu, R., Cagli, E., Dumas, C.: Deep learning for side-channel analysis and introduction to ASCAD database. J. Cryptogr. Eng. **10**(2), 163–188 (2019)
19. Cai, X., Li, R., Kuang, S., Tan, J.: An energy trace compression method for differential power analysis attack. IEEE Access **8**, 89084–89092 (2020)
20. Duan, X., Chen, D., Fan, X., Li, X., Ding, D., Li, Y.: Research and implementation on power analysis attacks for unbalanced data. Secur. Commun. Netw. **2020**, 1–10 (2020)