Cryptanalysis of Chaos-based Cryptosystem from the Hardware Perspective

Yuling Luo¹, Dezheng Zhang¹, Junxiu Liu^{1*}, Yunqi Liu¹, Yi Cao², Xuemei Ding^{3,4}

¹*Guangxi Key Lab of Multi-Source Information Mining and Security,
Faculty of Electronic Engineering, Guangxi Normal University, Guilin, China, 541004*

²*Department of Business Transformation and Sustainable Enterprise,
Surrey Business School, University of Surrey, Surrey, UK, GU2 7XH*

³*College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, 350108*

⁴*School of Computing, Engineering and Intelligent Systems,
Ulster University, Londonderry, UK, BT48 7JL*

**Corresponding authors: liujunxiu@gxnu.edu.cn*

Received (to be inserted by publisher)

Chaos has been used in cryptography for years and many chaotic cryptographic systems have been proposed. Their securities are often evaluated by conducting conventional statistical tests, however few studies have referred to the security issue of the chaotic hardware cryptographic systems. This paper evaluates the security of the chaotic cryptographic system from a hardware perspective by using the side channel analysis attack. First, a chaotic block cryptosystem is designed and implemented based on an Atmel microcontroller. Then the conventional statistical security tests, including SP 800-22 test, characters frequency test, avalanche test etc., are used to verify its security performance. In the meantime, the correlation power analysis attack is carried out for the security evaluation. Experimental results demonstrate that even though the chaotic cryptographic system can pass the conventional statistical tests, it still has the probability to be attacked from a hardware perspective using the leaked side channel information including execution time and power consumption etc. This paper proposes another way to analyze the security of the chaotic cryptosystem, which can aid designing mechanisms to enhance the security of the hardware cryptosystems in the future.

Keywords: Side Channel Analysis; Correlation Power Analysis; Chaotic Block Cipher; Round Keys

1. Introduction

Information security has been gaining dramatic importance in the increasingly severe information security environment. Progresses have been made in the fields of cryptography and cryptanalysis to enhance the information security. As one emerging field, the chaotic cryptosystems have attracted many research attentions [Kocarev & Lian, 2011; Liu *et al.*, 2017]. The chaotic system is a deterministic, non-linear system, which is sensitive to the initial conditions and parameters [Liu *et al.*, 2016; Pareschi *et al.*, 2009; Luo *et al.*, 2015]. The inherently correlation between chaos and cryptography [Kocarev & Lian, 2011] inspires researchers to combine cryptography with chaos. Many chaotic cryptographic algorithms have been proposed [Luo *et al.*, 2016; Liu *et al.*, 2016; Chen *et al.*, 2004; Eyebe Fouda *et al.*, 2014; Elgendy *et al.*, 2016].

The security of those cryptosystems mainly depends on the complexity of encryption algorithms. In the meantime, the cryptanalysis of the chaotic cryptosystem has never stopped. Conventional mathematical-based methods have been proposed for the chaotic cryptosystem cryptanalysis, where most of them are based on the known/chosen plaintext attacks [Li *et al.*, 2009a,b, 2008; Li & Lo, 2011; Zhang *et al.*, 2017a]. Adversaries utilize the plaintexts, ciphertexts and the characteristics of the algorithms to conduct the attacks.

Further from the hardware system perspective, when the cryptosystems are in operation, some unexpected side channel information probably leak from the cryptosystems, e.g. the execution time, power consumption and electromagnetic radiation. The side channel analysis (SCA) attack utilizes the side channel information to attack these cryptographic systems. The research of SCA attack and its countermeasure is a crucial branch of cryptography. It has been developing over twenty years. The SCA attack was first introduced in the approach of [Kocher, 1996], where the timing analysis attack was used to break a Rivest-Shamir-Adleman (RSA) cryptosystem. Timing analysis utilizes the relevance between the execution time and the key to attack the systems. Then the simple power analysis (SPA) attack and the differential power analysis (DPA) attack were proposed as two typical power analysis (PA) attack methods [Kocher *et al.*, 1999]. The PA attack performs key analysis utilizing the power consumption information correlated with the operations and the intermediate data. As another attack method, template attack was proposed in the approach of [Chari *et al.*, 2002]. To conduct this attack, a distribution model of the side channel information should be set up first, i.e. an identical experimental platform is needed. By using this method, the cryptosystem can be broken with less information than DPA attack. Then correlation power analysis (CPA) attack was proposed in the approach of [Brier *et al.*, 2004]. It utilizes the correlation between the power consumption and the data being proceeded in the cryptosystem. Compared to the DPA, the CPA is more robust and efficient [Brier *et al.*, 2004]. As a branch of SCA, electromagnetic analysis attack depends on the correlations between the operations of cryptosystems and the electromagnetic field surrounding the devices [Gandolfi *et al.*, 2001]. Its advantage is that the adversary does not need direct contacts with the cryptosystems to conduct the attack.

The security of a chaos-based random number generator (RNG) was tested using the side-channel information in the approach of [Pareschi *et al.*, 2009]. It verified one common argument, i.e. whether the chaotic system can be predicted due to its inherent deterministic nature. The RNG was implemented based on $0.35\mu\text{m}$ complementary metal oxide semiconductor (CMOS) technology and the PA attack was used to conduct the attack. The result showed that the internal state of the chaotic hardware system cannot be retrieved through the PA attack, and the security performance was verified. However, only the security of the hardware RNG was evaluated. The chaotic cryptosystems also include other components (e.g. flash memory [Cao *et al.*, 2016]) which may lead to the risk of leaking side channel information. It is also very common that the cryptosystems are designed based on the micro controllers (i.e. using the embedded software [Zhang *et al.*, 2017b]) where the power analysis attack can be performed. Therefore, the security of chaotic cryptosystems should be considered and analyzed entirely. This work analyzes whether the chaotic system is vulnerable to the SCA attack. To the best knowledge of the authors, this is the first research work to analyze and evaluate the security performance of the hardware chaotic cryptosystems using the side channel information. The cryptosystem is designed based on chaotic systems and its performance is evaluated using the conventional methods such as SP 800-22 test, entropy test and a set of dependence tests. Then the CPA attack is employed to attack the proposed chaotic cryptosystems and the security performance is analyzed.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of SCA attack and chaotic systems. Section 3 illustrates the theory of CPA attack and the chaotic block cipher used in this paper in detail. Section 4 analyzes the statistical security of the proposed cryptographic algorithm, introduces the experimental platform, and gives the power analysis results. Section 5 concludes this paper and discusses the future work.

2. Overview of Power Analysis and Chaos

Most cryptosystems are implemented on electronic devices, which may leak side channel information (e.g. electromagnetic, power and time consumption etc.). Side channel information is correlated with the operations being performed during encryption or decryption processes [Kocher, 1996]. For example, a processor may take a longer execution time and more power to perform a multiplication than an addition operation. Side channel information also correlates with the data being processed. For instance, more power may be consumed when transforming the hexadecimal byte from $0x00$ to the $0xFF$ than transforming from $0x00$ to $0x01$, due to that changing more bits from previous to the next states means more registers need to be switched (this leads to consuming more energy). Even if the power consumption differences caused by processing different data are quite small, the critical information can be exposed to the adversary by given enough power traces, using the DPA or CPA attack [Kocher, 1996].

2.1. Power analysis attack

Most of the modern digital systems are implemented based on CMOS technology, whose power consumption can be calculated by

$$P_{total} = P_{stat} + P_{dyn}, \quad (1)$$

where P_{total} , P_{stat} and P_{dyn} are the total, static and dynamic power consumption, respectively [Le *et al.*, 2007]. The P_{stat} can be extremely low [Rabaey *et al.*, 2003] which is caused by the leakage current or the current that keeps the system running. The P_{dyn} is due to the instantaneous short-circuit current and the charging or discharging of load capacitance when the circuit states changed. The P_{total} is mainly contributed by the P_{dyn} [Le *et al.*, 2007]. Therefore, the critical information (e.g. plaintext or ciphertext) corresponding to these transitions can be analyzed using DPA or CPA attack. In order to reduce the computational complexities of the attacks, DPA and CPA attack methods divide the cipher key into several parts, where each part is attacked separately. For example, for a brute force search of a 128-bit cipher key, the computational complexity is equal to 2^{128} . If dividing the cipher key into 16 parts and attacking each part separately, the computational complexity is $16 \times 2^8 = 2^{12}$ which is greatly reduced.

The DPA and CPA attacks utilize the correlations between the intermediate data and the power consumption for the cryptanalysis. To conduct a DPA attack, a number of randomly generated plaintexts are used as the input for the cryptosystem. During the processes of encryption, their corresponding power traces are collected. A selection function that uses plaintexts and one possible key as parameters partitions the power traces into two subsets. If the possible key is correct, the two subsets present different characters, e.g. in one subset, all the specific bits of intermediate data equal to 1, and in the other subset, they equal to 0. Then differential operation makes the differences between these two subsets prominent. If the possible key is incorrect, the two power subsets are very random. There is no apparent difference between them. Thus, the differential operation does not generate high values. The DPA also faces some problems, such as the ghost peak, lack of robustness and others [Brier *et al.*, 2004]. The CPA attack can overcome these drawbacks. It calculates the correlations between the actual power consumption and hypothetical power consumption [Li *et al.*, 2008]. The hypothetical power consumption is generated from a power consumption model, which takes the plaintexts and a possible key as input. The key which corresponds to the largest correlation coefficient is the most likely correct key. More details about the CPA are provided in Section 3.2.

2.2. Chaotic systems

Chaos is a widely studied phenomenon in the fields of science and engineering, such as in the subject of climate [Stehlík *et al.*, 2016], the dynamical system control [Messadi & Mellit, 2017], chemical aspects [Bodale & Oancea, 2015], etc. Chaotic systems are highly complex nonlinear dynamic systems. Although their models are deterministic, they can generate pseudo-random sequences. The chaotic systems have three main characteristics [Kocarev & Lian, 2011]: a). the sensitivity to initial values; b). the pseudo-randomness of its orbit; and c). the feasible implementations in both hardware and software. These features are related to

the characters of the conventional cipher algorithms [Elgendy *et al.*, 2016]. Thus, chaotic systems have been used as an alternative solution for the cryptosystems [Chen *et al.*, 2004; Wang *et al.*, 2016]. In next section, a cryptosystem based on a chaotic system is designed and the security performance is analyzed using various conventional evaluation methods. The security of the proposed chaotic cryptosystem is cryptanalyzed by the CPA attack from the hardware perspective.

3. Encryption Algorithm and Cryptanalysis Method

This section proposes a chaotic block cipher algorithm, which is used as the experimental subject for the CPA attack. In this algorithm, substitution-permutation network is used. The round keys are generated by a chaotic map. Then the CPA attack scheme aiming at breaking the proposed encryption algorithm is given in detail, including the selections of power model and the attack point etc.

3.1. The chaotic block cipher

According to Shannon information theory, confusion and diffusion are two fundamental properties of the encryption algorithms [Shannon, 1949]. Confusion makes the statistical relationships between the plaintext, ciphertext and keys as complex as possible. Diffusion makes each bit in the plaintext affect multiple bits in the ciphertext. Theoretically, changing any plaintext bit leads to the changes of half number of ciphertext bits. In this algorithm, these two properties are satisfied by conducting several alternating rounds of substitution and permutation operations.

The work flow of the chaotic block cipher is shown in the Fig. 1, where both a block of plaintext and a block of ciphertext include 128 bits. Round keys are generated based on a tent map, and the 128-bit master key is used to generate the chaotic parameters. The output of the chaotic system is mapped to the integer between 0 and 255 to get the round keys. After adding the round keys, a confusion process is carried out using an S-box lookup table. Then a $GF(2^8)$ addition, multiplication and a Cat Map operation are carried out to meet the diffusion property, where $GF(2^8)$ denotes the Galois field with order 2^8 . These steps are iterated for M times to ensure the security.

Fig. 1. Work flow of the proposed encryption algorithm

The aforementioned steps of the proposed encryption algorithm are given below in details.

Step 1. Generating the round keys. This chaotic block cipher contains M iterations, for each iteration, a 16-byte round key is used. Those round keys are generated based on a tent map, which is defined by

$$f(x, \alpha_i) = \begin{cases} \frac{x}{\alpha_i}, & 0 \leq x < \alpha_i \\ \frac{1-x}{1-\alpha_i}, & \alpha_i \leq x < 1 \end{cases} \quad (2)$$

where α_i is chaotic parameter. Each byte in the master key is transformed to corresponding α_i by

$$\alpha_i = 0.51 + K_i/1000, \quad (3)$$

where the i^{th} byte of the master key is denoted as K_i and $i \in [1, 16]$. To ensure the randomness of the round keys, for each parameter α_i , the tent map is iterated for 20 times. For the first chaotic parameter α_1 , iterate the tent map $f(x, \alpha_1)$ for 20 times to get the iteration result $x_{para(1)}$, where the initial value x_0 is a decimal number between 0 and 1 (in this paper it is randomly set as 0.5987). For each rest parameter α_i , iterate $f(x, \alpha)$ for 20 times, using α_i and the previous iteration result $x_{para(i-1)}$ as parameter and initial value respectively. It is described by

$$x_{para(i)} = f^{20}(x_{para(i-1)}, \alpha_i). \quad (4)$$

Finally, using the final iteration result $x_{para(16)}$ and α_{16} as the initial iteration value and parameter respectively, iterate the tent map for $100 + 16M$ times to generate a decimal number sequence (between 0 and 1) where each element is denoted as x_i and $i \in [1, 100 + 16M]$. To guarantee the randomness, the first 100 elements are discarded. The real number x_i is rounded to get the i^{th} byte of round key (i.e. k_i), which is achieved by

$$k_i = \text{floor}(x_i \times 255), \quad (5)$$

where $\text{floor}(x)$ is the function that returns the maximal integer not greater than x .

Step 2. Adding the round keys. In the first round of encryption, the i^{th} byte of the intermediate data is calculated by

$$s_i = k_i \oplus c_i, \quad (6)$$

where c_i is the i^{th} byte of the plain text. For the rest rounds of encryption, the result is the product of exclusive-or operation between the round keys and the temporary result from the previous round.

Step 3. The permutation procedure. It is carried out by substituting the data through an S-box. There are many ways to generate an S-box, such as using chaotic system [Wang & Wang, 2014; Çavuşlu *et al.*, 2017], artificial construction, and mathematical construction [Trappe & Washington, 2006]. The S-box used in the AES algorithm is used directly in this work, and it is generated through the mathematical method [Trappe & Washington, 2006].

Step 4. The diffusion operation. A round of diffusion operation that includes two stages is implemented in this work. In the first stage, the first byte of intermediate data remains unchanged and the following elements are calculated by

$$s_{i+1} = s_{i+1} \oplus s_i, \quad (7)$$

where $1 \leq i \leq 15$. In the second stage, the last byte of intermediate value remains unchanged, and the former bytes are calculated by

$$s_i = \begin{cases} s_{i+1} \times s_i, & s_{i+1} \neq 0 \\ s_i, & s_{i+1} = 0 \end{cases}. \quad (8)$$

Step 5. 2-D cat map. To improve the performance of confusion and diffusion, a 2-D cat map is implemented. The 2-D cat map is used to disorder the data stored in a square matrix. If the input cannot be reshaped into a square matrix, some padding schemes can be used, such as ANSI X.923, ISO10126, PKCS7, etc. In this paper, the intermediate values generated from above steps are organized in a 4×4 matrix, and then the Arnold cat map is applied by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \text{mod } N, \quad (9)$$

where N stands for the size of the matrix, x_n stands for the original column index of the matrix, and y_n stands for the original row index. After this operation, the intermediate data located at (x_n, y_n) is transformed to the new location (x_{n+1}, y_{n+1}) .

Step 6. Repeating the steps 2-5 for M times.

The chaotic block cipher system completes the encryption through the aforementioned steps 1-6. Its security performance is analyzed using the conventional statistical methods such as NIST SP 800-22 test, completeness test, avalanche test and strict avalanche test in Section 4. In next subsection, the details about the CPA is given and the procedures about applying the CPA attack to the chaotic block cipher system is presented.

3.2. The CPA attack

The CPA has advantages over the DPA in many aspects, such as the efficiency, robustness, and the required quantity of power information [Brier *et al.*, 2004]. In this paper, the CPA attack is used to get the crucial information utilizing the correlation between the intermediate keys and the power consumption. To attack the chaotic block cipher, the power model that converts the numerical information of intermediate data into the power consumption information should be selected and the point where the intermediate data be attacked should be selected.

3.2.1. The selection of power model

Two commonly used power consumption models are Hamming weight and Hamming distance models [Brier *et al.*, 2004]. The Hamming weight of a byte is the number of bits with high value, and the Hamming distance between two bytes is denoted by the number of different bits between them. The Hamming weight of v_0 is denoted as $HW(v_0)$ and the Hamming distance between two values v_0 and v_1 is denoted as $HD(v_0, v_1)$. The relationships between the two models are

- (a) The Hamming distance between two values v_0 and v_1 can be expressed by the Hamming weight of the XOR result of these two values, which is described by

$$HD(v_0, v_1) = HW(v_0 \oplus v_1). \quad (10)$$

- (b) If all the bits of v_0 are equal to zero, Eq. (10) is changed to

$$HD(v_0, v_1) = HW(0 \oplus v_1) = HW(v_1). \quad (11)$$

- (c) Similarly, if all the bits of v_0 are equal to 1, Eq. (10) is changed to

$$HD(v_0, v_1) = HW(\tilde{v}_1) = n - HW(v_1), \quad (12)$$

where n denotes the total bit number of v_1 .

The power consumption of the CMOS circuit is related to the number of bits transforming from the previous state to the current state. The Hamming distance model can well characterize the states changes happened on the bus or in register [Mangard *et al.*, 2007]. However, applying this model requires the knowledge of current and previous data transferred on the data bus, which is difficult for adversaries to meet this requirement [Mangard *et al.*, 2007]. However, many current micro controllers use pre-charge buses where all bits on the bus are set to high before the state changes. In this situation, the Hamming distance model turns to Eq. (12). The power consumption is negatively correlated with the Hamming weight of the data being proceeded. In this paper, the Hamming weight model is used, because pre-charge bus is used in the used cryptosystem.

3.2.2. The selection of the attack point

In the first round of the encryption algorithm, the first three steps are shown in Fig. 2. When a cryptographic system is in operation, some intermediate data are generated (such as the e'_i and the e_i). They are related to the plaintext and the key. When a cryptographic system processes them, the system leaks the corresponding amount of power consumption. Based on the working mechanism of the cryptographic algorithm, these intermediate data can be calculated using the plaintext and the key. Then the possible values of these intermediates can be converted to hypothetical power consumption using the Hamming weight model. Therefore, the hypothetical power generated from the plaintext and the correct keys can get the largest

Fig. 2. The first three steps of the algorithm

correlation coefficient with the power consumed by the cryptosystem. Compare to use e'_i to generate hypothetical power, to use e_i is better. Because a slight difference in the input of S-box leads to completely different output, which makes performing a CPA attack more efficiently [Mangard *et al.*, 2007]. In this work, the output of the S-box is used to calculate the hypothetical power consumption (i.e. e_i is selected as the attack point). Combining with the encryption algorithm, the hypothetical power consumption H is calculated by

$$H = HW(Sbox(x)), \quad (13)$$

where HW denotes calculating the Hamming weight operation, x is the input of the substitution operation, and $Sbox$ is the substitution operation using the S-box lookup table.

3.2.3. The procedure of CPA attack

To run a CPA attack, plaintexts are randomly generated. These plaintexts are used as the input of the cryptosystem and their corresponding power traces are collected. In order to reduce the computational complexity, the 128-bits key is divided into 16 bytes. Each byte of the key is attacked separately. Then the plaintexts and the possible keys are used to generate the hypothetical power consumption. Finally, the correlation between the actual and hypothetical power consumption is calculated. The possible key corresponding to the largest correlation coefficient is the most likely correct key. The details of attacking the i^{th} byte of round key (i.e. k_i) are given below.

- Generate the plaintexts. D sets of plaintexts are generated randomly and stored in matrix C . The size of C is $D \times I$, where I is the number of bytes in each set of plaintexts. The symbol of $c_{d,i}$ denotes the i^{th} byte in the d^{th} plaintexts.
- Collect the power consumption data. A total of D sets of plaintexts are used as input of the cryptosystem. During the encryption process, a total of D sets of J -sample points power traces are stored in matrix T . The size of T is $D \times J$, where J is the number of sampling points in each power trace, $t_{d,j}$ denotes the sampling point j in trace d .
- Calculate the hypothetical power consumption. The procedure of generating the d^{th} row of hypothetical power matrix H^i is shown in Fig. 3, where $c_{d,i}$ is the i^{th} byte of the d^{th} block of plaintext, s is one possible value of k_i . Each possible value of k_i is XORed with the $c_{d,i}$. Then a substitution operation is performed on the result of previous steps. The hypothetical power consumption is generated in the next step by calculating the Hamming weight of the previous result. Similar steps are repeated for

the rest of blocks to get the hypothetical power consumption matrix. The element in the hypothetical power consumption matrix is calculated by

$$h_{d,s}^i = HW(Sbox(c_{d,i} \oplus s)), \quad (14)$$

where the $(d, s)^{th}$ element $h_{d,s}^i$ of H^i denotes the hypothetical power consumption corresponding to the d^{th} group of plaintexts and the possible value $s \in [0, 255]$ of the i^{th} key byte.

Fig. 3. Calculating the hypothetical power correlated with the d^{th} set of plaintexts

- (d) Calculate the correlation between the actual and hypothetical power consumptions. The procedure of calculating the correlation matrix P^i is shown in Fig. 4. The matrix P^i is generated by calculating the correlation coefficient between each column of the matrix H^i and each column of the matrix T . The

Fig. 4. The correlation coefficient calculation. $\rho_{s,j}^i$ is the correlation coefficient between the s^{th} column of H^i (i.e. the hypothetical power generated from the possible value s) and the j^{th} column of T (i.e. the actual power consumption at sampling point j). P denotes the operation to calculate Pearson correlation coefficient.

Pearson correlation coefficient between the s^{th} column in H^i and the j^{th} column in T is calculated by

$$\rho_{s,j}^i = \frac{cov(H_{:,s}^i, T_{:,j})}{\sigma_{H_{:,s}^i} \sigma_{T_{:,j}}} = \frac{\sum_{d=1}^D [(h_{d,s}^i - \bar{h}_s^i)(t_{d,j} - \bar{t}_j)]}{\sqrt{\sum_{d=1}^D (h_{d,s}^i - \bar{h}_s^i)^2 \sum_{d=1}^D (t_{d,j} - \bar{t}_j)^2}}, \quad (15)$$

where $H_{:,s}^i$ denotes the s^{th} column vector in matrix H^i , $T_{:,j}$ denotes the j^{th} column vector in matrix T , $cov(H_{:,s}^i, T_{:,j})$ denotes the covariance between the column vector $H_{:,s}^i$ and column vector $T_{:,j}$, $\sigma_{H_{:,s}^i}$ denotes the standard deviation of column vector $H_{:,s}^i$, and $\bar{h}_s^i$ denotes the mean of $H_{:,s}^i$. The $\rho_{s,j}^i$ denotes the correlation between the hypothetical power calculated from the possible value s and the actual power at the sampling point j . To understand the generation of matrix H^i clearly, an example is shown in Fig. 5. It shows the procedure of generating the s^{th} row of P^i , which is the correlation between the hypothetical power generated from possible value s and the actual power consumption at each sampling point. The correlation coefficient between the s^{th} column of H^i and each column of T is calculated respectively, and the result is stored in the s^{th} row of P^i . The complete matrix P^i can be obtained by repeating similar operations. The row index of the largest $\rho_{s,j}^i$ in P^i is the best guess of k_i . Repeat these operations to get the best guess of each byte of round key, then all the round keys can be obtained.

Fig. 5. The procedure to calculate the s^{th} row of P^i

4. Experimental Results

This section firstly analyzes the security of the proposed cryptosystem through the statistical test. Then the experimental platform is briefly introduced. In the last subsection, the CPA attack is carried out to evaluate the security performance of the proposed system.

4.1. Statistical security tests

In this subsection, the security of the proposed cryptosystem is evaluated using the widely used statistical tests [Liu *et al.*, 2012; Tong *et al.*, 2015], including the character frequency test, information entropy test, dependence test, and SP 800-22 test.

4.1.1. Character frequency test

Character frequency is a crucial feature to analyze the security of a cryptographic algorithm. For example, some substitution cryptographic algorithms can be attacked using the frequency of characters [Tong *et al.*, 2015]. For a good encryption algorithm, the ASCII value distribution of the ciphertext should be uniform. In this paper, the character frequencies of 1,600,000 bytes of plaintext and their corresponding ciphertext are calculated. The results are shown in Fig. 6 and Fig. 7. As shown in Fig. 6, the distribution of the characters in plaintext is very nonuniform. From Fig. 7, it can be seen that the proportion of each ASCII value is around 0.004, which is closed to the probability of uniform distribution $1/256$. Therefore, it is difficult to break this encryption algorithm using probability statistics attack [Tong *et al.*, 2015].

4.1.2. Information entropy test

The information entropy measures the uncertainty of random events. The more uncertain the random event, the larger the information entropy. The entropy of a random variable X is defined by

$$H(X) = \sum_x p(x) \log_2 \frac{1}{p(x)}, \quad (16)$$

where $H(X)$ denotes the entropy of random event X , $p(x)$ stands for the possibility of X getting the result x . In this work, the value of a byte (8 bits) is regarded as a random event. The entropy of these random

Fig. 6. The distribution of the ASCII values in the plaintext

Fig. 7. The distribution of the ASCII values in the ciphertext

event denotes the minimum bits number to indicate these random events [Trappe & Washington, 2006]. If the values of the bytes are uniformly distributed, the entropy of these bytes should be equal to eight [Tong *et al.*, 2015], i.e. $p(x) = 1/256, x \in [0, 255]$. The Eq. (16) turns into

$$H(X) = \sum_{x=0}^{255} \frac{1}{256} \log_2 \frac{1}{1/256} = 8. \quad (17)$$

In this work after the analysis of 1,600,000 bytes ciphertext, the entropy is equal to 7.998942. It is very close to eight, which means the ciphertext are well confused [Tong *et al.*, 2015].

4.1.3. Dependence test

The dependence criteria can reflect the performance of the diffusion, and it is measured by the degree of completeness d_c , avalanche effects d_a and strict avalanche criterion d_{sa} [Preneel *et al.*, 2000]. For an encryption algorithm, if any output bit is affected by all the input bits, this algorithm is complete. The algorithm satisfies the avalanche effect, if about half number of ciphertext bits change when any bit of plaintext changes. The algorithm meets the strict avalanche effect if each bit of ciphertext has a probability of 50% to change whenever any bit of the plaintext changes. To calculate those three parameters, a dependence matrix A and a distance matrix B should be calculated first. A function with n bits input and m bits output is denoted as $f : (GF(2))^n \rightarrow (GF(2))^m$. The vector $x^{(i)}$ denotes the vector obtained by complementing the i^{th} bit of vector $x = (x_1, \dots, x_n) \in (GF(2))^n$. The $(i, j)^{th}$ element in dependence matrix A is denoted as a_{ij} . It denotes the number of inputs for which complementing the i^{th} input bit leads to the j^{th} output bit changing, i.e.

$$a_{i,j} = \#\{x \in X | (f(x^{(i)}))_j \neq (f(x))_j\}, \quad (18)$$

where the function of $\#\{\}$ calculates the number of elements in the set. The $(i, j)^{th}$ element in distance matrix B is denoted as b_{ij} . It denotes the number of inputs for which complementing the i^{th} input bit leads to j bits changing, i.e.

$$b_{i,j} = \#\{x \in X | HW(f(x^{(i)}) \oplus f(x)) = j\}. \quad (19)$$

After computing the dependence matrix and the distance matrix, those three parameters of d_c , d_a and d_{sa} can be calculated. The completeness is calculated by

$$d_c = 1 - \frac{\#\{(i, j) | a_{i,j} = 0\}}{mn}. \quad (20)$$

The avalanche is calculated by

$$d_a = 1 - \frac{\sum_{i=1}^n |\frac{1}{\#X} \sum_{j=1}^m 2jb_{i,j} - m|}{mn}. \quad (21)$$

The strict avalanche is calculated by

$$d_{sa} = 1 - \frac{\sum_{i=1}^n \sum_{j=1}^m |\frac{2a_{i,j}}{\#X} - 1|}{mn}. \quad (22)$$

An algorithm has a good dependence if the $d_c = 1$, $d_a \approx 1$, $d_{sa} \approx 1$. After analyzing 10,000,000 bits, the parameters of the proposed algorithm are $d_c = 1$, $d_a = 0.99977$ and $d_{sa} = 0.997468$, thus the performance of diffusion is qualified.

4.1.4. SP800-22 test

The SP 800-22 test is used to test the randomness of the sequence, and it includes 15 sub-tests [Tong *et al.*, 2015]. A good encryption algorithm should pass this test which indicates that it can resist statistical attack [Deng *et al.*, 2015]. After analyzing the 10,000,000 bits of ciphertext where the plaintext is generated randomly, the results are shown in Table 1. It can be seen that the proposed cryptosystem passed all the 15 sub-tests which demonstrate the randomness of the output sequence.

Table 1. SP800-22 test results

Statistical test	p -value	Conclusion
Frequency test	0.447884	pass
Block Frequency test (m=128)	0.113159	pass
Cumulative sums (forward)	0.503958	pass
Cumulative sums (reverse)	0.692403	pass
Rank test	0.309995	pass
Longest runs of ones test	0.964850	pass
Runs test	0.342874	pass
FFT test	0.571481	pass
Non-overlapping Templates test (m=9, B=000000001)	0.898738	pass
Overlapping Template test (m=9)	0.858003	pass
Universal test	0.720205	pass
Approximate entropy (m=10)	0.629128	pass
Random Excursions (x=+1)	0.815800	pass
Random Excursions Variant (x=-1)	0.472917	pass
Linear Complexity (M=500)	0.057064	pass
Serial (m=16)	0.656760	pass

4.2. Security analysis under CPA attacks

4.2.1. Experimental platform

The experimental platform for the power analysis attack usually consists of four parts: a power supplier, cryptosystem hardware device, power consumption captor, and power consumption analyzer. In this work, the proposed cryptosystem is implemented using an Atmel XMEGA128D4 micro controller, see Fig. 8(c). The power consumption of the cryptosystem is collected by a Xilinx Spartan 6 device-based FPGA board via the sub-miniature version A (SMA) and the universal synchronous/asynchronous receiver/transmitter (USART) connections to the cryptosystem, see Fig. 8(b). The collected power consumption data is transmitted to the computer through a universal serial bus (USB) and analyzed by a software. The system clock of the hardware cryptosystem and the sampling frequency are controlled by the FPGA device, where the latter is set to be four times of the former for the power sampling. The computer software can initialize the FPGA device, control the communications between the hardware systems, and analyze the power consumption data while the cryptosystem is under the CPA attacks.

Fig. 8. The cryptosystem and the CPA platform. (a) CPA software, (b) CPA hardware system and (c) Cryptosystem.

4.2.2. Experimental data analysis

To carry out the CPA attack, one hundred 128-bit plaintexts are generated randomly and used as the input of cryptosystem. One hundred groups of corresponding power consumption data are collected. Each group data contains 3,000 sample points. One group of power consumption data is shown in Fig. 9, which is obtained when the cryptosystem encrypts one plaintext. For the total 100 plaintexts, the power consumption data is stored in a matrix T , where the $(d, j)^{th}$ element $t_{d,j}$ denotes the sampling point j while encrypting the d^{th} plaintext.

In order to attack the i^{th} byte k_i of the first group of round keys, the corresponding hypothetical power consumption matrix H^i is calculated. The s^{th} column vector $H_{:,s}^i$ denotes the hypothetical power consumption corresponding to the possible value s of the key-byte k_i . Then the correlation coefficients between every column vector of H^i and every column vector of T are calculated. For the byte k_i , the guessed value which corresponds to the largest correlation coefficient is mostly like the actual key, and this

Fig. 9. Power consumption curve in time domain

Table 2. The best key guesses and their point index

Byte index	Actual key	Largest correlation coefficient			Second largest correlation coefficient	
		Guessed value	Correlation	Point Index	Guessed value	Correlation coefficient
0	0x25	0x25	0.847	403	0x44	0.435
1	0x47	0x47	0.913	448	0x00	0.411
2	0x88	0x88	0.865	516	0x1D	0.459
3	0xE3	0xE3	0.863	584	0x9C	0.474
4	0x35	0x35	0.815	652	0x74	0.463
5	0x65	0x65	0.900	720	0x30	0.438
6	0xC1	0xC1	0.816	788	0xED	0.466
7	0x76	0x76	0.868	856	0x34	0.481
8	0xE1	0xE1	0.867	924	0x97	0.418
9	0x39	0x39	0.847	992	0x3F	0.451
10	0x6E	0x6E	0.867	1060	0x2C	0.441
11	0xD1	0xD1	0.799	1128	0xA0	0.467
12	0x58	0x58	0.863	1196	0x42	0.436
13	0xA8	0xA8	0.869	1264	0xB0	0.450
14	0xA5	0xA5	0.813	1332	0xEB	0.481
15	0xAC	0xAC	0.648	1383	0xCF	0.473

value is called as the best key guess of the byte k_i . The CPA attack result is shown in Table 2, where its actual value and the best key guess are shown for every k_i and $i \in [0, 15]$. The best key guesses are equal to the actual keys, and the points where the best key guesses obtained are shown in Table 2. The possible values corresponding to the second largest correlation coefficients are also given which are much lower than the best key guesses. Table 2 shows that the correlation coefficients between the power consumption and the correct key guesses are around 0.8, and the correlation coefficients between the power consumption and the incorrect key guesses are less than 0.5. It is shown that using the CPA method and the Hamming weight model, the correct key can be obtained, and the differences between the correct key and the incorrect key are obvious.

4.2.3. Efficiency analysis of CPA attack

In order to evaluate the efficient of the proposed power consumption model, i.e. $h_{d,s}^i = HW(Sbox(c_{d,i} \oplus s))$, the relations between actual and hypothetical power consumption are analyzed. As the procedure of attacking each byte of the key is similar, the following analysis focus on attacking the first key byte k_0 . The actual power consumption data in the 403th column of trace matrix T and the hypothetical power consumption corresponding to the correct key $0x25$ are firstly normalized between 0 and 1. Then they are shown in Fig. 10. It can be seen that the actual power consumption and hypothetical power consumption calculated through the proposed model are negatively related, i.e. the data proceed by the cryptographic system can be attacked using leaked power consumption information.

The correlation between the actual power consumption and the hypothetical power consumption calculated from the correct key guess is shown in Fig. 11 (a). If one bit of the correct guess is changed, and the changed value is used as the input of the model, the correlation between the actual power consumption and the hypothetical power consumption is shown in Fig. 11 (b). The relatively high correlation coefficient for the correct guess and low correlation coefficient for the incorrect guess show that this model can well characterize the power consumption of the micro controller with pre-charge buses [Mangard *et al.*, 2007]. Therefore, the cryptosystem can be attacked using the proposed method.

4.2.4. Performance improvement

Each piece of power consumption data contains 3,000 sampling points, while only some of them are valuable for the CPA attack. Thus, dividing the power consumption data into several stages, and carrying out the CPA attack in one particular stage can reduce the computational complexity. The power consumption curve of encrypting one piece of plaintext is shown in Fig. 9. From the general outline, each stage can be roughly separated. From point 0 to point 330, there are 16 similar peaks, and the time consumption is relatively low. Therefore, this stage is consisted by 16 identical simple operations. In this stage, the plain text are

Fig. 10. Power consumption varies with plaintext

Fig. 11. Correlations of hypothetical and actual power consumption

in the operations of XOR with the intermediate cipher keys. From point 380 to point 1,400, there are 16 similar peaks, but the time consumption is greater than the previous stage. In this stage, the substitution operation is carried out. According to the previous analysis in Section 3.2, carrying the CPA attack in this stage performs very well. Through CPA attacks, the points where the best correlation coefficient of each k_i obtained are marked with red points. The locations of these red points verify that the various stages of the encryption process are divided correctly, by analyzing the shape of the power consumption curve. Conducting CPA attack in this stage can significantly reduce the computational complexity by 60%.

4.2.5. Discussion

A chaotic block cryptographic system was attacked using the CPA attack in this paper. It is only an example and it should be noted that other chaotic block ciphers can also be attacked using the SCA attack, if the cryptographic system leaks the side channel information correlated with the data processed or the operation performed. In most of chaotic block ciphers, round keys are generated from the master key using chaotic maps at the beginning of the encryption, and they participate in each round of cryptographic operations. During those operations, there are some intermediate values that correlate with the plaintext and the round keys. When the cryptographic hardware system process those intermediate values, the corresponding side channel information probably leaks. Therefore, round keys can be attacked by analyzing the correlation between the hypothetical and actual side channel information. The hypothetical information is calculated from a model which maps the hypothetical intermediate values to hypothetical side channel information. The model used in this paper is a Hamming Weight model, which maps the hypothetical intermediate values $Sbox(c_{d,i} \oplus s)$ to hypothetical power consumption $h_{d,s}^i = HW(Sbox(c_{d,i} \oplus s))$ as illustrated in Section 3.2.3. It can be also applied to other chaotic block cryptographic systems. The intermediate values being attacked are selected according the features of the ciphers and it is correlated with the plaintext and the round keys. Therefore theoretically, the CPA can attack most of chaotic block cipher hardware systems if the correct intermediate values and the power consumption model are selected.

5. Conclusion

In this paper, a chaotic block cryptographic system was designed and implemented based on an Atmel XMEGA128D4 micro controller. The proposed cryptographic algorithm passed the conventional statistical security tests including SP 800-22, avalanche test, character frequency test. The CPA was carried out to attack the proposed cryptographic system. Results showed that although the cryptographic algorithm passed the statistical tests, its critical information such as intermediate round keys can still be analyzed by the CPA attack using the leaked power consumption information when the hardware cryptosystem is in operation. This paper proposed another direction to evaluate the security performances of the chaotic

hardware cryptosystems. Future work will investigate the chaotic cryptosystems that can resist the CPA attack, and design mechanisms to increase the security of the chaotic hardware systems.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant 61661008, the Guangxi Natural Science Foundation under Grants 2017GXNSFAA198180 and 2016GXNSFCA380017, the funding of Overseas 100 Talents Program of Guangxi Higher Education, Guangxi Key Lab of Multi-source Information Mining & Security under Grant MIMS15-07, the Doctoral Research Foundation of Guangxi Normal University under Grant 2016BQ005, the Research Project of Guangxi University of China, and the Innovation Project of Guangxi Graduate Education under Grant YCSW2018096.

References

- Bodale, I. & Oancea, V. A. [2015] "Chaos control for willamowskirssler model of chemical reactions," *Chaos, Solitons & Fractals* **78**, 1 – 9.
- Brier, E., Clavier, C. & Olivier, F. [2004] "Correlation Power Analysis with a Leakage Model," *Cryptographic Hardware and Embedded Systems*, pp. 16–29.
- Cao, L., Luo, Y., Bi, J., Qiu, S., Lu, Z., Harkin, J. & McDaid, L. [2016] "An authentication strategy based on spatiotemporal chaos for software copyright protection," *Security and Communication Networks* **8**, 4073–4086.
- Çavuşolu, Ü., Kaçar, S., Pehlivan, I. & Zengin, A. [2017] "Secure image encryption algorithm design using a novel chaos based S-Box," *Chaos, Solitons & Fractals* **95**, 92–101.
- Chari, S., Rao, J. R. & Rohatgi, P. [2002] "Template Attacks," *Cryptographic Hardware and Embedded Systems*, pp. 13–28.
- Chen, G., Mao, Y. & Chui, C. K. [2004] "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals* **21**, 749–761.
- Deng, Y., Hu, H., Xiong, W., Xiong, N. N. & Liu, L. [2015] "Analysis and Design of Digital Chaotic Systems with Desirable Performance via Feedback Control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **45**, 1187–1200.
- Elgendy, F., Sarhan, A. M., Eltobely, T. E., El-Zoghdy, S. F., El-sayed, H. S. & Faragallah, O. S. [2016] "Chaos-based model for encryption and decryption of digital images," *Multimedia Tools and Applications* **75**, 11529–11553.
- Eyebe Fouda, J. S. A., Effa, J. Y. & Ali, M. [2014] "Highly secured chaotic block cipher for fast image encryption," *Applied Soft Computing Journal* **25**, 435–444.
- Gandolfi, K., Mourtel, C. & Olivier, F. [2001] "Electromagnetic Analysis: Concrete Results," *Cryptographic Hardware and Embedded Systems*, pp. 251–261.
- Kocarev, L. & Lian, S. [2011] *Chaos-Based Cryptography*, Vol. 354 (Springer Berlin Heidelberg, Berlin, Heidelberg).
- Kocher, P., Jaffe, J. & Jun, B. [1999] "Differential Power Analysis," *CRYPTO* (Springer, Santa Barbara, CA, USA), pp. 388–397.
- Kocher, P. C. [1996] "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *International Cryptology Conference on Advances in Cryptology* (Berlin, Heidelberg), pp. 104–113.
- Le, T.-H., Clediere, J., Serviere, C. & Lacoume, J.-L. [2007] "Noise Reduction in Side Channel Attack Using Fourth-Order Cumulant," *IEEE Transactions on Information Forensics and Security* **2**, 710–720.
- Li, C., Li, S., Asim, M., Nunez, J., Alvarez, G. & Chen, G. [2009a] "On the security defects of an image encryption scheme," *Image and Vision Computing* **27**, 1371–1381.
- Li, C., Li, S., Chen, G. & Halang, W. A. [2009b] "Cryptanalysis of an image encryption scheme based on a compound chaotic sequence," *Image and Vision Computing* **27**, 1035–1039.
- Li, C. & Lo, K.-T. [2011] "Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks," *Signal Processing* **91**, 949–954.

- Li, H., Wu, K., Peng, B., Zhang, Y., Zheng, X. & Yu, F. [2008] "Enhanced Correlation Power Analysis Attack on Smart Card," *International Conference for Young Computer Scientists (IEEE)*, pp. 2143–2148.
- Liu, W., Sun, K. & Zhu, C. [2016] "A fast image encryption algorithm based on chaotic map," *Optics and Lasers in Engineering* **84**, 26–36.
- Liu, Y., Luo, Y., Song, S., Cao, L., Liu, J. & Harkin, J. [2017] "Counteracting Dynamical Degradation of Digital Chaotic Chebyshev Map via Perturbation," *International Journal of Bifurcation and Chaos* **27**, 1750033.
- Liu, Y., Tian, S., Hu, W. & Xing, C. [2012] "Design and statistical analysis of a new chaotic block cipher for Wireless Sensor Networks," *Communications in Nonlinear Science and Numerical Simulation* **17**, 3267–3278.
- Luo, Y., Cao, L., Qiu, S., Lin, H., Harkin, J. & Liu, J. [2016] "A chaotic map-control-based and the plain image-related cryptosystem," *Nonlinear Dynamics* **83**, 2293–2310.
- Luo, Y., Du, M. & Liu, J. [2015] "A symmetrical image encryption scheme in wavelet and time domain," *Communications in Nonlinear Science and Numerical Simulation* **20**, 447–460.
- Mangard, S., Oswald, E. & Popp, T. [2007] *Power Analysis Attacks* (Springer US, Boston, MA).
- Messadi, M. & Mellit, A. [2017] "Control of chaos in an induction motor system with LMI predictive control and experimental circuit validation," *Chaos, Solitons and Fractals* **97**, 51–58.
- Pareschi, F., Scotti, G., Giancane, L., Rovatti, R., Setti, G. & Trifiletti, A. [2009] "Power analysis of a chaos-based Random Number Generator for cryptographic security," *IEEE International Symposium on Circuits and Systems*, 1, pp. 2858–2861.
- Preneel, B., Bosselaers, A., Rijmen, V., Rompay, B. V., Granboulan, L., Stern, J., Murphy, S., Dichtl, M., Biham, E. & Dunkelman, O. [2000] "Comments by the NESSIE Project on the AES Finalists," *Tetsu-to-Hagane* **40**, 700–709.
- Rabaey, J. M., Chandrakasan, A. & Nikoli, B. [2003] *Digital Integrated Circuits-A Design perspective* (Pearson), ISBN 0130909963.
- Shannon, C. E. [1949] "Communication Theory of Secrecy Systems," *Bell System Technical Journal* **28**, 656–715.
- Stehlík, M., Dušek, J. & Kiseák, J. [2016] "Missing chaos in global climate change data interpreting?" *Ecological Complexity* **25**, 53–59.
- Tong, X.-J., Wang, Z., Liu, Y., Zhang, M. & Xu, L. [2015] "A novel compound chaotic block cipher for wireless sensor networks," *Communications in Nonlinear Science and Numerical Simulation* **22**, 120–133.
- Trappe, W. & Washington, L. C. [2006] *Introduction to cryptography with coding theory* (Pearson Education India).
- Wang, Q., Yu, S., Li, C., Lu, J., Fang, X., Guyeux, C. & Bahi, J. M. [2016] "Theoretical Design and FPGA-Based Implementation of Higher-Dimensional Digital Chaotic Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers* **63**, 401–412.
- Wang, X. & Wang, Q. [2014] "A novel image encryption algorithm based on dynamic S-boxes constructed by chaos," *Nonlinear Dynamics* **75**, 567–576.
- Zhang, L. Y., Liu, Y., Pareschi, F., Zhang, Y., Wong, K.-W., Rovatti, R. & Setti, G. [2017a] "On the Security of a Class of Diffusion Mechanisms for Image Encryption," *IEEE Transactions on Cybernetics*, 1–13.
- Zhang, X., Yu, S., Chen, P., Lü, J., He, J. & Lin, Z. [2017b] "Design and ARM-embedded implementation of a chaotic secure communication scheme based on H.264 selective encryption," *Nonlinear Dynamics* **89**, 1949–1965.