International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 7788-7800

https://doi.org/10.70517/ijhsa464670

A Dual-Dimensional Analysis Method for Information Security Design and Anti-Attack Capability of Finite Field Cryptographic Algorithms

Mingzhi Qi[1,*] and Rui Chen[2]
[1] College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, Shandong, 266044, China
[2] Hangzhou Zhihuigu Technology Co., LTD., Hangzhou, Zhejiang, 310000, China
Corresponding authors: (e-mail: qimingzhi2024@163.com).

Abstract As an important issue in modern society, information security is addressed in this paper by designing a cryptographic algorithm based on finite fields in symmetric cryptographic algorithms. The SM4 algorithm is selected for design, and byte substitution is performed using finite field methods to propose a masking defense scheme for the SM4 algorithm. After analyzing the masking protection and related power consumption of the cryptographic algorithm designed in this paper, its resistance to attacks is compared with that of the standard SM4 algorithm and the ordinary SM4 multiplication masking algorithm. After adding the mask, the guessed values of each byte in each round of the key show no significant peaks, making it impossible for CPA attacks to obtain the true information of the master key, thereby achieving protection of the master key. Under the same frequency, the mask area of this algorithm is the largest (40,000 gates), but it has the highest information security.

Index Terms symmetric cryptography, finite field, SM4 algorithm, mask defense, anti-attack capability

I. Introduction

Cybersecurity refers to the processes and measures involved in protecting network data from unauthorized access, disclosure, tampering, or destruction [1]. In the digital age, individuals, businesses, and government agencies increasingly rely on cyberspace for information exchange and business operations. Therefore, ensuring cybersecurity is crucial for safeguarding privacy rights, trade secrets, state secrets, and social stability [2]-[5]. The consequences of cyberattacks can be catastrophic, including economic losses, damage to reputation, and even threats to national security [6], [7]. However, with the widespread adoption of the internet and the rapid development of information technology, the scope of cyberattacks has expanded, and the methods used have also increased [8], [9]. Examples include hacking, virus infections, and data theft, which pose serious threats to network information transmission, presenting ongoing challenges to cybersecurity defense [10]-[12].

To address cybersecurity threats, various security technologies and strategies have been developed, including firewalls, intrusion detection systems, security protocols, virtual private networks, multi-factor authentication, and encryption technologies [13]-[15]. Among these, encryption technology is particularly critical, as it ensures that only users with the correct key can access information by encoding data [16]. Although existing technologies have improved network security to some extent, they also have issues such as inappropriate encryption algorithm selection, incomplete security protocol implementation, vulnerabilities in authentication and authorization mechanisms, and insufficient security auditing and logging [17]-[20]. Therefore, researching and developing new information security encryption algorithms, particularly novel encryption technologies combining finite field operations, is of great significance for enhancing network security defense capabilities [21], [22].

Encryption algorithms are primarily divided into two categories: symmetric encryption algorithms and asymmetric encryption algorithms. In symmetric encryption technology, the same key is used for both encryption and decryption, meaning that the sender and receiver must pre-share a secret key, and ensuring the security of this key is of utmost importance. Reference [23] introduces searchable symmetric encryption technology deployed in the cloud and proposes combining index tables and trees to improve update and storage efficiency, thereby achieving richer functionality while reducing the risk of data leakage. Reference [24] proposes a miniature symmetric encryption algorithm (NTEA) based on the TEA (Tiny Encryption Algorithm), which has low memory overhead and is easy to implement in hardware and software. NTEA dynamically introduces additional key scrambling points in each encryption round, effectively mitigating the avalanche effect in IoT data transmission and improving decryption efficiency. Reference [25] introduces pre-modulo operations into the traditional scrambling-diffusion structure to design an efficient symmetric image encryption algorithm. This enhances the algorithm's dependence on plaintext images, ensuring the security of image data during transmission over public networks.

Asymmetric encryption, also known as public-key encryption, uses a pair of mathematically related keys: a public key and a private key. The public key can be openly shared with anyone for encrypting information, while only the corresponding private key can decrypt such information. Reference [26] indicates that asymmetric encryption technology based on matrix decomposition plays a significant role in wireless body area networks (WBANs), effectively addressing robustness issues in WBANs while demonstrating good performance in terms of key space and encrypted image pixel distribution. Reference [27] proposes a digital signature scheme suitable for asymmetric encryption technology, which can resist private key recovery and forgery attacks while significantly reducing the execution time of the signing operation process, thereby efficiently generating and verifying signatures. Reference [28] employs optical images of biometric keys to encrypt asymmetric cryptosystems, while combining a random phase mask key from the Phase Truncated Fourier Transform (PTFT) scheme to improve key distribution, thereby enhancing the security of encryption and decryption operations.

Since symmetric encryption algorithms face significant security challenges, and asymmetric encryption algorithms have limitations in terms of encryption and decryption efficiency, integrating the advantages of both encryption algorithms is expected to better protect data security. Literature [29] explores the characteristics of encryption algorithms in different systems and the effectiveness of hybrid encryption, suggesting that using asymmetric encryption algorithms to protect keys and symmetric algorithms to encrypt messages can further enhance the security of data transmission. Literature [30] emphasizes that hybrid cryptography can effectively improve the security and performance of remote cloud servers by integrating multiple cryptographic algorithms, fully considering algorithm combination design, implementation methods, and limitations, aiming to provide a sufficiently secure foundational mechanism for cloud computing.

This paper starts from information security requirements and selects the SM4 algorithm for cryptographic algorithm design based on finite fields in symmetric cryptographic algorithms. A finite field method is used for byte substitution, and a low-area-consumption inverse calculation method is proposed to simplify the computational complexity. Taking side-channel attacks as an example, a countermeasure for cryptographic algorithm masking schemes is devised, and ultimately, an SM4 algorithm masking defense scheme is proposed. Experimental simulations using CPA attacks are conducted to calculate four rounds of keys and verify the effectiveness of the proposed cryptographic algorithm's defense mechanism. A power consumption analysis is performed on the SM4 algorithm to assess its resistance to side-channel attacks. Finally, the proposed SM4 masking defense scheme is compared with the standard SM4 algorithm and the standard SM4 multiplication masking algorithm to validate the security of the proposed scheme.

II. Cryptographic algorithm design based on finite fields

II. A. Theoretical Foundations of Symmetric Cryptographic Algorithms

II. A. 1) Cryptography Basics

(1) Introduction to Cryptography

Cryptography ensures the confidentiality, integrity, availability, and resistance of information. Cryptography encompasses two major areas of development: cryptography and cryptanalysis. Cryptography involves the study of encryption algorithms and the writing of cryptographic codes to ensure information confidentiality, while cryptanalysis involves the study of how to obtain plaintext from ciphertext. The two areas are both antagonistic and interdependent, and together they drive the advancement of cryptography.

Based on the type of key used during encryption and decryption, cryptographic algorithms are primarily divided into two categories: symmetric cryptographic algorithms and asymmetric cryptographic algorithms. In symmetric cryptographic algorithms, the encryption key and decryption key are typically the same or can be derived from each other through a simple transformation. In contrast, asymmetric cryptographic algorithms use different keys for encryption and decryption, and these keys cannot be derived from each other. The encryption key is publicly disclosed, while the decryption key is kept confidential. The following sections will provide a detailed introduction to symmetric cryptographic algorithms.

(2) Symmetric Cryptographic Algorithms

Symmetric cryptographic algorithms primarily include block ciphers and stream ciphers. The following sections will provide a detailed introduction to the SM4 cryptographic algorithm [31], [32].

1) Block Cipher Algorithms

Block cipher algorithms divide plaintext into equal-length blocks, each of which is transformed individually. Each block is encrypted using the key to convert the plaintext into an equal-length ciphertext block. There are two common structures for block cipher algorithms: the Feistel structure and the SPN structure. The Feistel structure is a typical iterative structure that uses a product-based approach for encryption, enabling data to be thoroughly scrambled and forming a high-security encryption system. Common Feistel-based cipher algorithms include DES, RC5, GOST, and LOKI. The SPN structure is a substitution-permutation network, primarily comprising three modules: the substitution layer—used to achieve confusion, the permutation layer—used to achieve diffusion, and the key addition layer—used to achieve data and key mixing. Common cryptographic algorithms using the SPN structure include DGZN, HDED, etc. The SM4 algorithm uses the Feistel structure.

2) Stream ciphers

Stream ciphers use a “one-time pad” approach for encryption and decryption, generating a random key stream using a key stream generator to encrypt plaintext into ciphertext. During decryption, the same key stream is used to decrypt the ciphertext. Some typical stream ciphers include: the Zu Chongzhi Algorithm Suite (ZUC Algorithm), SOSEMANUK, Grain, and RC4.

(3) Block Symmetric Cryptography Algorithm Operating Modes

The Electronic Code Book (ECB) mode is the most commonly used mode for block ciphers. In ECB mode, the plaintext is divided into several groups based on its size before encryption. Each group is then encrypted and decrypted separately using the same key K. The computations for each group are performed independently without interfering with one another. The encrypted output is the ciphertext of the groups. The decryption process is analogous to the encryption process. The Cipher Block Chaining (CBC) mode can be defined as:

Encryption process: $C_i = E_K(P_i \oplus C_{i-1})$ where $C_0 = IV, i = 0,1,\cdots,n$.

Decryption process: $P_i = D_K(C_i) \oplus C_{i-1}$, where $C_0 = IV, i = 0,1,\cdots,n$.

Block Cipher Mode (BCM), also known as CBC mode.

The CFB mode is similar to the CBC mode in that it also introduces feedback. The slight difference is that in the CFB mode, the encryption module is entered first, followed by the XOR operation, while in the CBC mode, the order is reversed. Specifically, in the CFB mode, the input to the first group of encryption modules is the initial vector IV. The encrypted result of the initial vector IV is XORed with the first group of plaintext to obtain the first group of ciphertext. The decryption structure is the same. The CFB mode can be defined as follows:

Encryption process: $C_i = E_K(C_{i-1}) \oplus P_i$, where $C_0 = IV, i = 0,1,\cdots,n$.

Decryption process: $P_i = E_K(C_{i-1}) \oplus C_i$, where $C_0 = IV, i = 0,1,\cdots,n$.

Output Feedback Mode (OFB mode). The OFB mode can convert cipher blocks into synchronized cipher streams. During the first encryption, the initial vector IV is encrypted and transformed into a cipher stream, which is then XORed with the plaintext to obtain the ciphertext. In subsequent encryptions, the output of the previous group is again XORed with the current plaintext by the encryption module. Due to the symmetry of the XOR operation, the decryption and encryption operations are the same. The OFB mode can be defined as:

Encryption process: $C_i = P_i \oplus O_i$, where $O_0 = IV, O_i = E_K(O_{i-1}), i = 0,1,\cdots,n$.

Decryption process: $P_i = C_i \oplus O_i$, where $O_0 = IV, O_i = E_K(O_{i-1}), i = 0,1,\cdots,n$.

Counter mode, also known as CTR mode, is also referred to as integer counting mode. CTR mode is similar to OFB mode in that both use a key stream to perform encryption and decryption. The difference lies in the fact that the key stream in the CTR mode is obtained through an incrementing encryption counter. The first half of the encryption counter is a random number, the same as the initial vector (IV), while the latter half is the counter, typically starting from 1. Similarly, after passing through the encryption module, the ciphertext is obtained by XORing with the plaintext block, and the decryption operation is identical to the encryption operation.

II. A. 2) Finite fields

(1) Definition of a finite field

A field is a set on which we can perform various operations without leaving the set. A finite field contains only a finite number of elements, where is the order of the finite field, is a prime number, and is a positive integer. Finite fields are also known as Galois fields and have wide applications in modern coding, computer theory, combinatorial mathematics, and other fields.

In the SM4 symmetric cipher algorithm, the finite field used is, and each element in the finite field represents each byte in the SM4 cipher algorithm.

(2) Operations on finite fields

The SM4 cryptographic algorithm primarily uses addition and multiplication operations over the finite field $GF(2^8)$, represented by the symbols “$\oplus$” and “·”, respectively.

Addition involves summing the coefficients of two polynomials over $GF(2^8)$; if each element in $GF(2^8)$ is treated as a byte, then addition is equivalent to a bitwise XOR operation. For example:

Let $f(x)=x^6+x^5+x^2+x+1, g(x)=x^4+x^2+1$, then:

$$f(x)\oplus g(x)=x^6+x^5+x^4+x \tag{1}$$

$$f(x)\oplus g(x)=x^6+x^5+x^4+x \tag{2}$$

$$67+15=72 \tag{3}$$

Multiplication is similar to general polynomial multiplication, with the only difference being that the coefficients are multiplied in $GF(2^8)$ and the result of the multiplication must be modulo-operated on the irreducible polynomial $m(x)$ in $GF(2^8)$, where $m(x)=x^8+x^4+x^3+x+1$. For example, if $f(x)=x^6+x^5+x^2+x+1, g(x)=x^4+x^2+1$, then:

$$f(x)\oplus g(x)=x^7+x^6+x^3+x \tag{4}$$

$$01100111\cdot 0010101=11001010 \tag{5}$$

$$67\cdot 15=ca \tag{6}$$

The calculation process is as follows:

$$\begin{aligned} f(x)\times g(x) &= (x^6+x^5+x^2+x+1)\times(x^4+x^2+1) \\ &= x^{10}+x^9+x^8+x^7+2x^6+2x^5+2x^4+x^3+2x^2+x \end{aligned} \tag{7}$$

$$\begin{aligned} f(x)\cdot g(x) &= f(x)\times g(x) \quad \bmod \quad m(x) \\ &= x^7+x^6+x^3+x+1 \end{aligned} \tag{8}$$

In particular, 2 multiplied by a polynomial can be calculated using the following formula: when $b_7=1$, perform an XOR operation; otherwise, do not perform the operation.

Let $m(x)=b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+1$, then:

$$2\cdot m(x)=\begin{cases} b_6b_5b_4b_3b_2b_1b_0 0 & b_7=0 \\ (b_6b_5b_4b_3b_2b_1b_0 0)\oplus(00011011) & b_7=1 \end{cases} \tag{9}$$

Based on this, larger multiplication operations can be performed.

For example, to multiply 3 by the polynomial $m(x)$, (00000011) can be split into two parts, multiplied by (00000010) and (00000001) respectively, and then the two products can be XORed:

$$\begin{aligned} 00000011)\cdot(b_7b_6b_5b_4b_3b_2b_1b_0) &= [(00000010)\oplus(0000001)]\oplus(b_7b_6b_5b_4b_3b_2b_1b_0) \\ &= [(00000010)\oplus b_7b_6b_5b_4b_3b_2b_1b_0]\oplus(b_7b_6b_5b_4b_3b_2b_1b_0) \end{aligned} \tag{10}$$

Multiplying by the polynomial $m(x)$ can be broken down into two operations of multiplying by (00000010):

$$(00000100)\cdot(b_7b_6b_5b_4b_3b_2b_1b_0)=(00000010)\cdot(00000010)\cdot(b_7b_6b_5b_4b_3b_2b_1b_0) \tag{11}$$

(3) Polynomials over finite fields

Elements of the finite field $GF(2^8)$ can also be represented by polynomials:

$$b(x)=b_{n-1}x^{n-1}+b_{n-2}x^{n-2}+\cdots+b_2x^2+b_1x+b_0 \tag{12}$$

where p_i is an element of the finite field $GF(2^8)$. Polynomial operations also include addition and multiplication. Let two polynomials in the finite field $GF(2^8)$ be:

$$m(x)=m_3x^3+m_2x^2+m_1x+m_0, \quad n(x)=n_3x^3+n_2x^2+n_1x+n_0 \tag{13}$$

Then, the polynomial addition operation is:

$$m(x)\oplus n(x)=(m_3\oplus n_3)x^3+(m_2\oplus n_2)x^2+(m_1\oplus n_1)x+(m_0\oplus n_0) \tag{14}$$

The multiplication of polynomials is represented by $\otimes$, and the operation is relatively complex. Let:

$$d(x) = m(x) \otimes n(x) = d_3 x^3 + d_2 x^2 + d_1 x + d_0 \tag{15}$$

Multiplication can be represented by a matrix, namely:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} m_0 & m_3 & m_2 & m_1 \\ m_1 & m_0 & m_3 & m_2 \\ m_2 & m_1 & m_0 & m_3 \\ m_3 & m_2 & m_1 & m_0 \end{bmatrix} \begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix} \tag{16}$$

II. B.SM4 Algorithm

The SM4 algorithm has a block size of 128 bits and a key length of 128 bits. Both the encryption algorithm and the key expansion algorithm use a 32-round iteration structure. The decryption algorithm is the same as the encryption algorithm, except that the decryption round keys are used in reverse order of the encryption round keys.

II. B. 1) SM4 encryption algorithm

For convenience, we define Z_2^m to represent an m-bit vector. The symbol $\oplus$ denotes a 32-bit exclusive OR operation, and the symbol $<<< i$ denotes a 32-bit cyclic left shift by i bits.

Let the 128-bit plaintext input be X:

$$X = (X_0, X_1, X_2, X_3), X_i \in Z_2^{32}, i = 0,1,2,3 \tag{17}$$

After encryption, the 128-bit ciphertext output is obtained as Y:

$$Y = (Y_0, Y_1, Y_2, Y_3), Y_i \in Z_2^{32}, i = 0,1,2,3 \tag{18}$$

Round key $rk_i \in Z_2^{32}, i = 0,1,2,\cdots,31$. The encryption transformation can be expressed as follows:

$$\begin{gathered} X_{i+4} = X_i \oplus T(X_{i+1}, X_{i+2}, X_{i+3}, rk_i) \\ i = 0,1,2,\cdots,31 \end{gathered} \tag{19}$$

$$Y = (Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32}) \tag{20}$$

Let T be the composite permutation and R be the reverse permutation.

The composite permutation T consists of two parts, a nonlinear transformation τ and a linear transformation L, i.e., $T = L(\tau(.))$. The nonlinear transformation τ is the S box permutation, where the input is A and the output is B:

$$\begin{aligned} B &= (b_0, b_1, b_2, b_3) \\ &= \tau(A) \\ &= (sbox(a_0), sbox(a_1), sbox(a_2), sbox(a_3)) \end{aligned} \tag{21}$$

$$b_i \in Z_2^8, i = 0,1,2,3 \tag{22}$$

The input of the linear transformation L is the output of the nonlinear transformation τ, and the output is 32 bits:

$$C = L(B) = B \oplus B <<< 2 \oplus B <<< 10 \oplus B <<<< 18 \oplus B <<< 24 \tag{23}$$

After the 32nd round of operations, the final ciphertext is obtained by applying the reverse transformation R. The reverse transformation is defined as follows:

$$R(C_0, C_1, C_2, C_3) = (C_3, C_2, C_1, C_0) \tag{24}$$

$$C_i \in Z_2^{32}, i = 0,1,2,3 \tag{25}$$

II. B. 2) SM4 Key Expansion Algorithm

The encryption operation consists of 32 rounds, and the round keys are generated using a key expansion algorithm. Let the 128-bit input key be: $MK = (MK_0, MK_1, MK_2, MK_3)$ The round keys $rk_i \in Z_2^{32}, i = 0,1,2,\cdots,31$ are generated as follows:

$$(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1 MK_2 \oplus FK_2, MK_3 \oplus FK_3) \tag{26}$$

$$\begin{gathered} rk_i = K_{i+4} = K_i \oplus T^{'}(K_{i+1}, K_{i+2}, K_{i+3}, CK_i) \\ i = 0,1,2,\cdots,31 \end{gathered} \tag{27}$$

$T^{'}$ is basically the same as T, except that the linear transformation part has changed: $T^{'} = L^{'}(\tau(.))$:

$$L'(B) = B \oplus B <<< 13 \oplus B <<< 23 \tag{28}$$

Among them, CK_i and FK_i are given constants, CK, which can also be obtained by calculation:

$$\begin{gathered} CK_i = (ck_{i,0}, ck_{i,1}, ck_{i,2}, ck_{i,3}) \\ ck_{i,j} = (4i + j) \bmod 256, i = 0,1,2,\cdots,31 \end{gathered} \tag{29}$$

The overall block diagram of the SM4 algorithm encryption operation is shown in Figure 1.

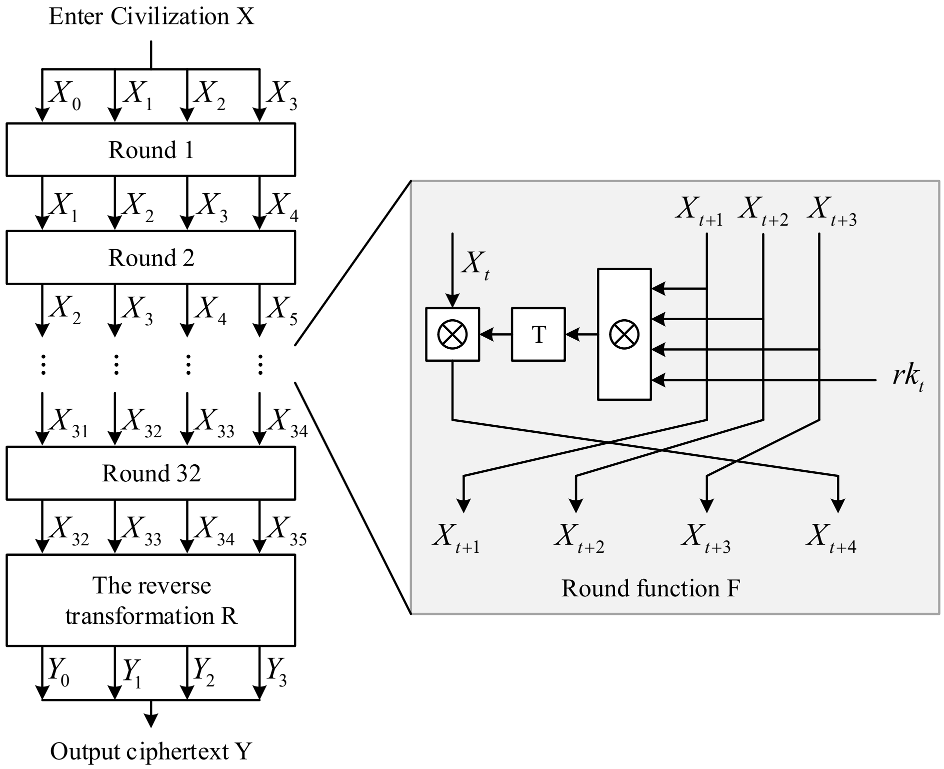


Figure 1: SM4 algorithm encryption process

II. C.Anti-attack methods

II. C. 1) Byte Replacement

Byte substitution, as the only nonlinear operation component, can also be referred to as the S-box. Both algorithm standards provide the lookup table data for the S-box [33]. Since the operation is performed on 8-bit data, there are only 256 possible values to guess during an attack, making this step a critical attack point for CPA attacks.

Considering the need to minimize area consumption and enable algorithm reuse, this paper adopts a finite field-based method to implement the byte replacement component. Since direct inversion in the $GF(2^8)$ field requires a significant amount of computation, this paper proposes a low-area inversion calculation method. This algorithm reduces the inversion operation from the $GF(2^8)$ field to the $GF(2^4)$ field, significantly reducing the computational complexity. Let a be an element in $GF(2^8)$; the formula for computing the inverse of element a is:

$$\delta a = a_h x + a_1 \tag{30}$$

$$d = (a_h^2 \times p_0) + (a_h \times a_1) + a_1^2 \tag{31}$$

$$d' = d^{-1} \tag{32}$$

$$a_h' = a_h \times d' \tag{33}$$

$$a_1' = (a_h + a_l) \times d' \tag{34}$$

$$a^{-1} = \delta^{-1}(a_h' x + a_l') \tag{35}$$

Among them, δ is the isomorphism mapping matrix, δ^{-1} is its inverse matrix, a_h, a_1, d and p_0 are elements of $GF(2^4)$, where a_h and a_1 are the upper four bits and lower four bits of the result of the isomorphism mapping of a, respectively, d is an important intermediate value in the calculation, and operations d and d^{-1} simplify the 8-bit inverse problem to a 4-bit inverse problem, p_0 is a constant on $GF(2^4)$.

II. C. 2) Power Consumption Side-Channel Attacks and Countermeasures

Side-channel attacks, as a commonly used attack technique, are one of the major threats to hardware security [34]. In 1999, Kocher et al. proposed simple power analysis (SPA) and differential power analysis (DPA) attacks. SPA was used to reveal the instruction sequences executed by cryptographic algorithms, while DPA was used to plot the differential power traces, thereby breaking the DES implementation. In 2002, Roman Novak et al. expanded on the SPA research, combining chosen plaintext attacks with SPA to successfully attack smart cards supporting RSA. In 2004, correlated power analysis (CPA) attacks were proposed. Unlike DPA attacks, CPA attacks calculate the corresponding key information by computing correlation coefficients, making this attack method more resistant to noise. These attack methods all rely on the principle that intermediate values of algorithms are correlated with power consumption data to obtain accurate keys. Therefore, the key to defense lies in weakening the relationship between intermediate values and power consumption data.

Currently, the primary protection methods at the algorithm level are divided into masking methods and hiding methods. Hiding methods weaken the relationship between power consumption and intermediate values by evenly distributing or randomizing power consumption, but hiding methods alter the overall power consumption characteristics of the algorithm. Masking methods randomize intermediate values, making it difficult for attackers to analyze the correct key. Masking methods are primarily divided into Boolean masking and arithmetic masking. Since S-boxes are a key focus of protection in algorithms, masking algorithms can be further categorized based on their implementation forms into lookup table-based masking algorithms, addition chain-based masking algorithms, and finite field-based masking algorithms. Finite field-based masking algorithms have advantages such as lower algorithmic complexity, lower hardware costs, and simpler implementation compared to the other two methods. Current masking defense schemes primarily focus on random masking and the number of computational modules. For first-order masking, only one random mask is used per operation, and the area of the computational modules directly impacts the overall resource consumption of the algorithm. Therefore, simplifying the computational process while increasing randomness is an important research direction for masking schemes. By reusing the critical multiplication modules, six multiplication modules are used to improve the inverse operation of the masking S-box, and this additive masking can effectively resist zero-value attacks.

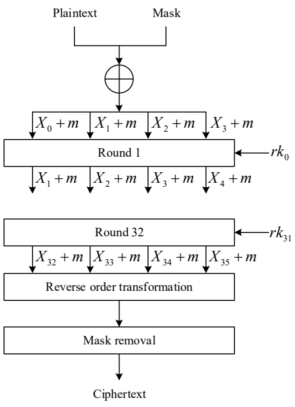


Figure 2: Mask defense scheme for SM4 algorithm

II. C. 3) SM4 Algorithm Mask Defense Scheme Design

The masking defense scheme of the SM4 algorithm is shown in Figure 2.

The wheel function structure with masked protection is shown in Figure 3. A 32-bit random number m is used as the mask, and the 128-bit plaintext is divided into four groups of 32-bit words: X_0, X_1, X_2, X_3, which are respectively XORed with the mask m. Next, the masked $X_1, X_2,$ and X_3 undergo mask-type S-box transformations and linear transformations, followed by XOR operations with the masked X_0. Finally, the masked X_4 is iteratively computed through compensation transformations. As shown in Figure 2, $X_1 \oplus m, X_2 \oplus m, X_3 \oplus m$ and the computed $X_4 \oplus m$ are used as inputs for the next iteration. This process is repeated for 32 rounds, and the ciphertext is obtained through reverse transformation and demasking operations.

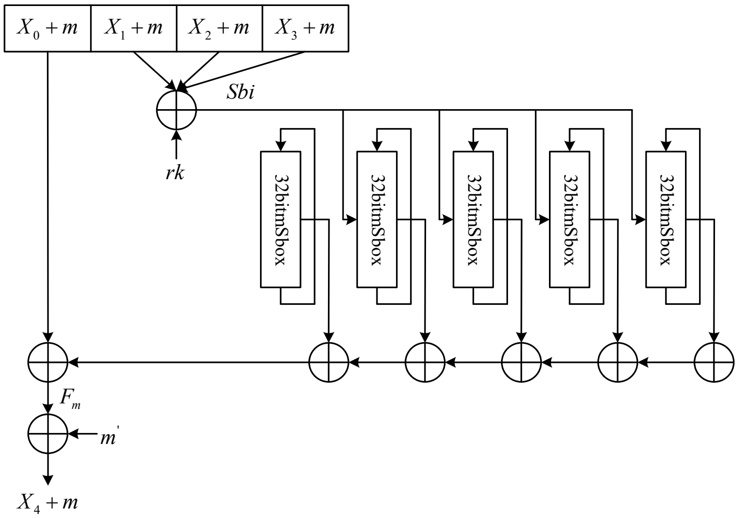


Figure 3 Round function structure with mask defense

The iterative calculation process in Figure 3 is as follows:

$$Sbi = X_1 \oplus m \oplus X_2 \oplus m \oplus X_3 \oplus m \oplus rk_0 \tag{36}$$

$$MuskSbox(Sbi) = Sbox(X_1 \oplus X_2 \oplus X_3 \oplus rk_0) \oplus Sbox(m) \tag{37}$$

$$\begin{aligned} F_m &= X_0 \oplus m \oplus L[MaskSbox(Sbi)] \\ &= X_0 \oplus m \oplus L[Sbox(X_1 \oplus X_2 \oplus X_3 \oplus rk_0)] \oplus L[Sbox(m)] \\ &= m \oplus X_4 \oplus L(Sbox(m)) \end{aligned} \tag{38}$$

$$m' = L[Sbox(m)] \tag{39}$$

$$F_{\mathrm{m}} + m' = m \oplus X_4 \tag{40}$$

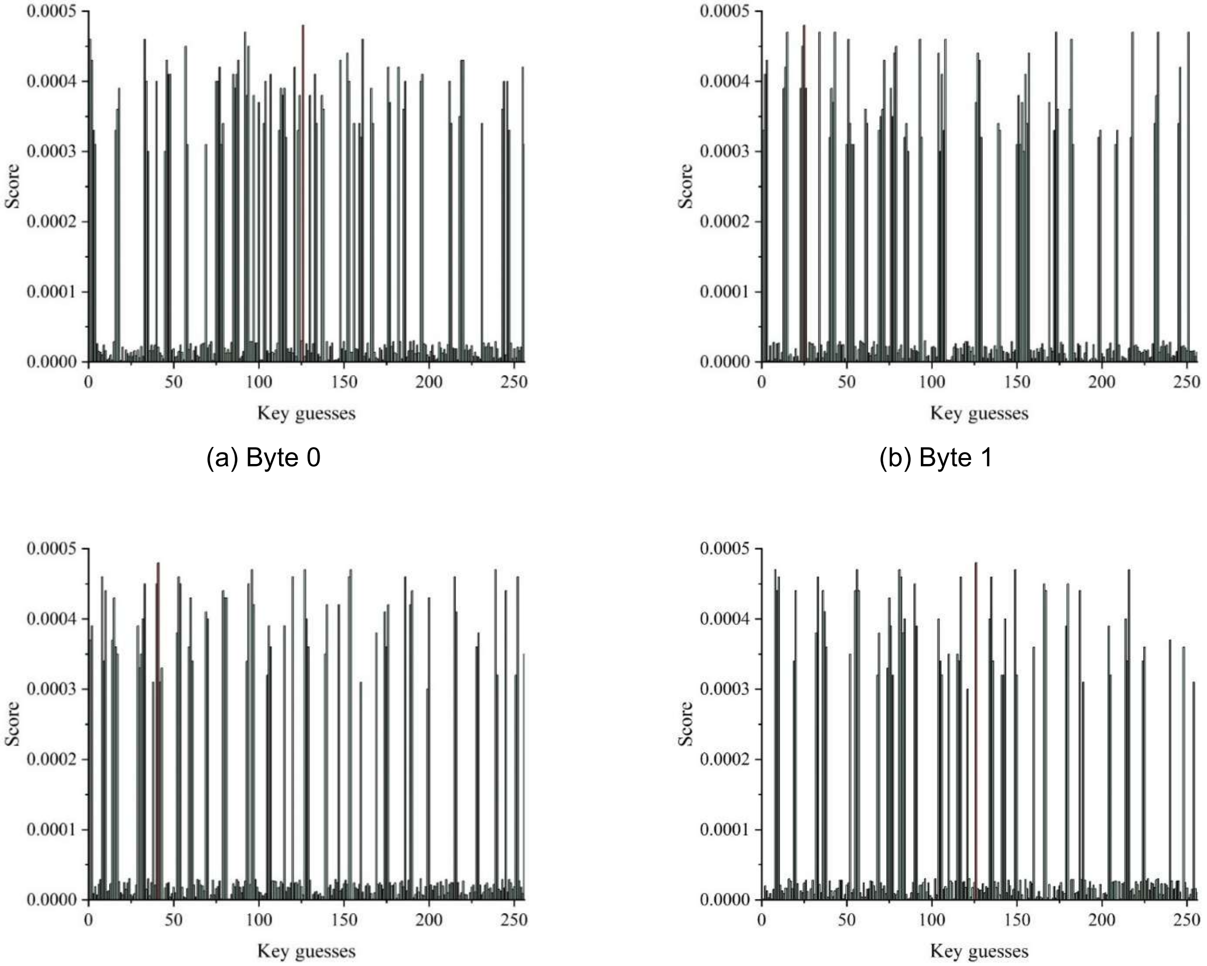
Among these, Sbi is the input of the mask-type S-box, rk_0 is the corresponding key for this round, MaskSbox is the mask-type byte substitution function, L is the linear transformation function, F_m is the output of the mask-compensated previous round function, and m' is the random number required for the compensation transformation.

III. Experimental Results and Analysis

III. A. SM4 Algorithm Mask Protection Analysis

This section verifies through experimental simulation whether this method can resist CPA attacks. Similarly, to prevent data transmission errors, we set the initial key of the algorithm in the verification function and the initial key of the algorithm uploaded to the development board to be consistent during the experiment. Using CPA analysis, the master key information of the SM4 cryptographic algorithm was successfully obtained. After implementing the masking countermeasure, the same CPA attack method was used in the experiment to verify whether the algorithm's actual key could be obtained. During the experiment, power consumption data collection must be verified to ensure that the oscilloscope can capture the data, thereby ensuring the experiment runs normally.

During the CPA power consumption analysis experiment, it was verified that the oscilloscope could normally collect the power consumption curve after the algorithm was masked. The first four rounds of keys were attacked, and each byte's guessed value corresponded to the sampling points of the relevant power consumption curve. Unlike the power consumption curve collected in the previous CPA attack analysis, the power consumption curve corresponding to each byte's guessed value was not a peak. The first round of keys obtained through the masked CPA attack is shown in Figure 4. As shown in Figure 4, after adding the mask, the first round of keys obtained through the attack is [126, 25, 41, 128], successfully obtaining the four bytes of the first round of keys' guessed values.



(a) Byte 0

(b) Byte 1

(c) Byte 2

(d) Byte 3

Figure 4: Add mask CPA attack the first round of keys

Similarly, by analyzing the power consumption curves obtained during the second to fourth rounds of the SM4 algorithm using a CPA power consumption attack, we can recover the estimated values for each byte in each round's key and identify corresponding values in the collected power consumption curves. Using the same method, we obtain the power consumption curves for the second to fourth rounds of key collection, where the second-round key is [49, 172, 126, 158], the third-round key is [193, 73, 54, 26], and the fourth-round key is [184, 133, 129, 167].

When the SM4 cryptographic algorithm is not performing any operations, we can use the CPA power analysis attack to obtain the first four rounds of keys and recover the initial master key based on the principles of the SM4 algorithm. The cryptographic algorithm employs masking countermeasures to defend against side-channel power consumption attacks, hiding intermediate data during the algorithm's execution process, thereby preventing the detection of its actual energy consumption. Using the same CPA analysis attack to obtain the four-round keys and restore the master key based on the algorithm's principles, we calculated that the restored SM4 cryptographic algorithm master key is completely different from the initial master key we set for the algorithm. In each group of experimental simulations, the guessed values for each byte in each round key did not exhibit obvious peak values, making it impossible to obtain the correct guessed values for each byte in each round key. This indicates that the CPA attack cannot obtain the true information of the algorithm's master key, proving that this method can resist CPA attacks.

III. B. Power consumption analysis related to the SM4 algorithm

To improve simulation speed, this section pre-computes the middle round keys and stores them in memory, using a round operation module to perform a CPA attack on the original SMS4 algorithm. First, under the SMIC 0.18μm CMOS process, the RTL code of the SMS4 algorithm round operation circuit module is synthesized to obtain the circuit netlist E1. Using a 128-bit random number as the plaintext input for the circuit netlist E1, 2,000 transistor-level simulations were performed, and the ciphertext and current trace curves were recorded. Since the power supply voltage is a constant value, the circuit's power consumption trace can be represented by the current trace and yields the same attack results. The experimental results are as follows.

Taking Byte 0 as an example, the Pearson correlation coefficient r is calculated. The correlation coefficient curve for the DPA attack on Byte 0 is shown in Figure 5.

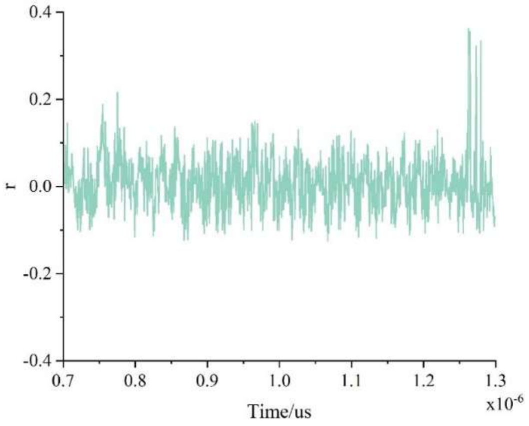
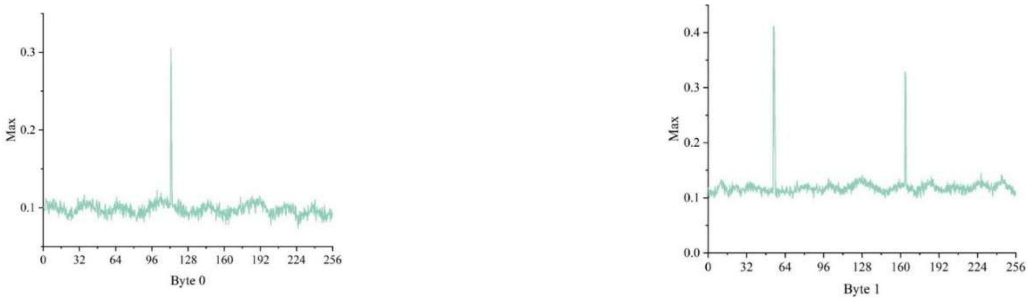


Figure 5: The correlation coefficient curve of DPA attack on Byte 0

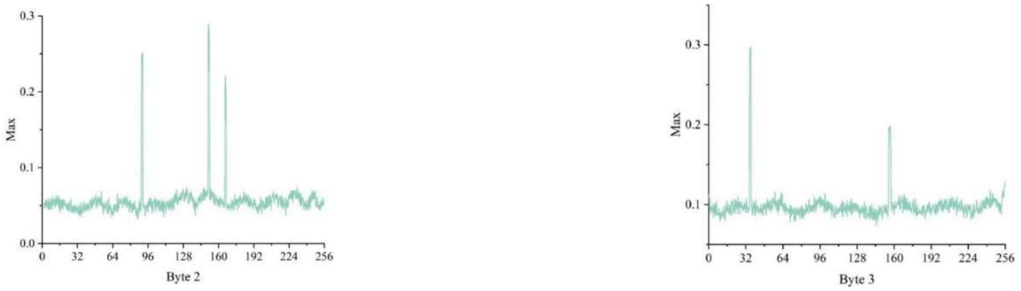
The peak absolute value distribution of the correlation coefficient curves for DPA attacks on Byte 0, Byte 1, Byte 2, and Byte 3 is shown in Figure 6. In Figure 6(a), there is a distinct peak on the y-axis, and the corresponding x-axis coordinate represents the correct key guess value. Thus, we obtained the correct subkey 0x82 through the DPA attack. Similarly, the same method can be used to directly attack the remaining three bytes. From Figures 6(b) to (d), we can respectively obtain the remaining subkeys 0x36, 0xB0, and 0x105, i.e., 0x8236B0105, which matches the given standard value, proving that the attack proposed in this paper targeting the round keys of the SM4 algorithm is successful.

Based on the obtained round keys, by utilizing the method for deriving the encryption key, the final four round keys can be further attacked to derive the encryption key, thereby completing the entire differential power analysis attack process.



(a) Byte 0

(b) Byte 1



(c) Byte 2

(d) Byte 3

Figure 6: The peak absolute distribution of the correlation coefficient curve of the DPA attack

III. C. Comparison of masking schemes

The author has performed VLSI design on the SM4 multiplication mask algorithm. Considering that multiple data paths in this scheme involve the same L transformation, the L transformation can be reused. A timing folding method is adopted to temporarily store the results of one computation path before performing the computation of another path. This extends the algorithm's computation time but reduces the hardware implementation cost.

Under the SMIC 0.18 μm CMOS process, RTL code design was completed for the standard SM4 algorithm, the standard SM4 multiplication mask algorithm, and the SM4 mask defense scheme circuit module proposed in this paper. Logic synthesis and layout routing were performed under the same conditions, resulting in three circuit netlists: E1, E2, and E3. The verification and analysis platform used was the Inspector 4.2 side-channel analysis platform developed by Dutch company Riscure. The FPGA implementations of the three schemes were analyzed. During the experiment, the circuit current traces during encryption were sampled. Since the power supply voltage is a constant value, the circuit's power consumption traces can be represented by current traces, and the two are equivalent in power consumption analysis. A total of 100,000 random plaintext simulation tests were conducted, recording the corresponding ciphertext and current traces. After sampling the current traces, differential power analysis tests were performed using the Inspector4.2 side-channel analysis platform. Since the S-box is the component with the highest power consumption in the SM4 algorithm circuit and is also the most vulnerable to attacks, DPA attacks were conducted on the S-box outputs of the first round of the three SM4 cryptographic algorithm circuits. Correlation analysis was performed on the first four bytes of the first round S-box outputs, with the results shown in Figures 7 to 9. The curves in the figures represent correlation curves, with each byte represented by a separate curve. If a noticeable peak appears in the correlation curve, it indicates significant information leakage in that portion of the operation, and an attacker could use this as an attack point to potentially infer the subkey for that round.

As shown in Figure 7, the S-box output of the standard SM4 algorithm exhibits significant correlation with prominent peaks. By collecting 1,000 sets of encryption and decryption curves for DPA analysis, the complete initial key can be derived. As shown in Figure 8, after adding a multiplication mask to the SM4 algorithm, the correlation in the S-box output section is significantly reduced, but there are still subtle peaks, indicating that DPA attacks are still possible. By collecting 10,000 sets of encryption and decryption curves for DPA analysis, the complete initial key can be derived. As shown in Figure 9, there are no obvious peaks in the S-box output section. Collecting 100,000 pairs of encryption and decryption curves for DPA analysis yields no meaningful information. The experimental results demonstrate that the SM4 masking scheme proposed in this paper can effectively resist side-channel attacks.

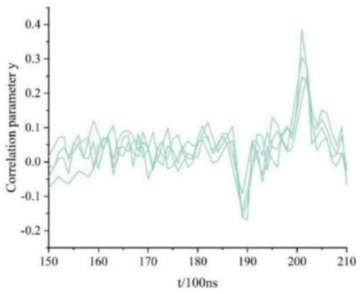


Figure 7: Standard SM4 correlation analysis results

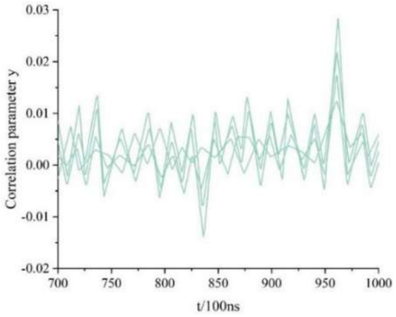


Figure 8: Common SM4 multiplication mask correlation analysis results

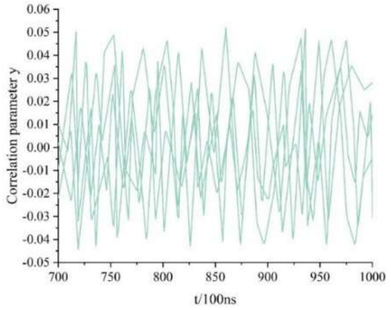


Figure 9: The correlation analysis results of our scheme

The performance metrics of the above three circuit modules are compared in Table 1. At the same frequency, the area of the algorithm implemented in this paper is 25% larger than that of the conventional SM4 multiplication mask algorithm, but its resistance to side-channel analysis is greatly enhanced. For security cryptographic chips with high security requirements, security is more important than implementation cost.

Table 1: Performance comparison of three algorithms

Algorithm	Frequency /MHz	Throughput /(Mb·s^{-1})	Area /gates	Mask range	Analysis cost
Standard SM4	75	350	25000	Unprotected	2200 curves
Common SM4 multiplication mask	75	350	32000	Just cover up the S box	25000 curves
Our Scheme	75	350	40000	Full cover	7×10^5~1.05×10^6 curves

IV. Conclusion

The author designed a cryptographic algorithm based on finite fields, selected the SM4 algorithm, performed byte substitution, and designed countermeasures against power-based side-channel attacks. A masking defense scheme for the SM4 algorithm was designed to enhance the cryptographic algorithm's resistance to attacks.

After adding the mask, the key distributions obtained from the first four rounds of attacks are [126, 25, 41, 128], [49, 172, 126, 158], [193, 73, 54, 26], and [184, 133, 129, 167]. There are no obvious peaks in the guessed values of each byte in each round of keys, indicating that the cryptographic algorithm in this paper can resist CPA attacks. Through DPA attacks, the subkey 0x8236B0105, which is identical to the standard value, is obtained. At the same frequency (75MHz), the masking area of the algorithm proposed in this paper is increased by 60% and 25% compared to the standard SM4 algorithm and the ordinary SM4 multiplication masking algorithm, respectively, significantly enhancing its resistance to side-channel analysis and improving its security.

References

[1] Guangxu, Y. (2021). Research on computer network information security based on improved machine learning. Journal of Intelligent & Fuzzy Systems, 40(4), 6889-6900.

[2] Abraham, A., Dutta, P., Mandal, J. K., Bhattacharya, A., & Dutta, S. (2018). Emerging technologies in data mining and information security. Proceedings of IEMIS-2018.

[3] Schuetz, S. W., Benjamin Lowry, P., Pienta, D. A., & Bennett Thatcher, J. (2020). The effectiveness of abstract versus concrete fear appeals in information security. Journal of Management Information Systems, 37(3), 723-757.

[4] Safa, N. S., Von Solms, R., & Furnell, S. (2016). Information security policy compliance model in organizations. Computers & security, 56, 70-82.

[5] Ahmad, Z., Ong, T. S., Liew, T. H., & Norhashim, M. (2019). Security monitoring and information security assurance behaviour among employees: An empirical analysis. Information & Computer Security, 27(2), 165-188.

[6] Dacier, M. C., Dietrich, S., Kargl, F., & König, H. (2016). Network attack detection and defense: security challenges and opportunities of software-defined networking. In Dagstuhl Seminar (Vol. 16361, p. 6).

[7] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. IEEE Communications Surveys & Tutorials, 22(3), 1909-1941.

[8] Liu, L., Zhang, L., Liao, S., Liu, J., & Wang, Z. (2021). A generalized approach to solve perfect Bayesian Nash equilibrium for practical network attack and defense. Information Sciences, 577, 245-264.

[9] Swami, R., Dave, M., & Ranga, V. (2019). Software-defined networking-based DDoS defense mechanisms. ACM Computing Surveys (CSUR), 52(2), 1-36.

[10] Chan, S. H., & Janjarasjit, S. (2019). Insight into hackers' reaction toward information security breach. International Journal of Information Management, 49, 388-396.

[11] Yinfeng, L. (2015). The Network Security Management System Design Against the New Virus Invasion. Open Automation and Control Systems Journal, 7, 1492-1498.

[12] Al-Harrasi, A., Shaikh, A. K., & Al-Badi, A. (2021). Towards protecting organisations' data by preventing data theft by malicious insiders. International Journal of Organizational Analysis, 31(3), 875-888.

[13] Niu, Y., Du, W., & Tang, Z. (2022). Computer Network Security Defense Model. In Journal of Physics: Conference Series (Vol. 2146, No. 1, p. 012041). IOP Publishing.

[14] Arogundade, O. R. (2023). Network security concepts, dangers, and defense best practical. Computer Engineering and Intelligent Systems, 14(2).

[15] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., & Kambhampati, S. (2020). A survey of moving target defenses for network security. IEEE Communications Surveys & Tutorials, 22(3), 1909-1941.

[16] Panda, M. (2016, October). Performance analysis of encryption algorithms for security. In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES) (pp. 278-284). IEEE.

[17] Mushtaq, M. F., Jamel, S., Disina, A. H., Pindar, Z. A., Shakir, N. S. A., & Deris, M. M. (2017). A survey on the cryptographic encryption algorithms. International Journal of Advanced Computer Science and Applications, 8(11).

[18] Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. International Journal of Communication Networks and Information Security, 12(2), 256-272.

[19] Abikoye, O. C., Haruna, A. D., Abubakar, A., Akande, N. O., & Asani, E. O. (2019). Modified advanced encryption standard algorithm for information security. Symmetry, 11(12), 1484.

[20] Al-Shabi, M. A. (2019). A survey on symmetric and asymmetric cryptography algorithms in information security. International Journal of Scientific and Research Publications (IJSRP), 9(3), 576-589.

[21] Zhang, Q. (2021, January). An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption. In 2021 2nd international conference on computing and data science (CDS) (pp. 616-622). IEEE.

[22] Bhanot, R., & Hans, R. (2015). A review and comparative analysis of various encryption algorithms. International Journal of Security and Its Applications, 9(4), 289-306.

[23] Poh, G. S., Chin, J. J., Yau, W. C., Choo, K. K. R., & Mohamad, M. S. (2017). Searchable symmetric encryption: Designs and challenges. ACM Computing Surveys (CSUR), 50(3), 1-37.

[24] Rajesh, S., Paul, V., Menon, V. G., & Khosravi, M. R. (2019). A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. Symmetry, 11(2), 293.

[25] Ye, G., & Huang, X. (2017). An efficient symmetric image encryption algorithm based on an intertwining logistic map. Neurocomputing, 251, 45-53.

[26] Niu, Z., Zheng, M., Zhang, Y., & Wang, T. (2019). A new asymmetrical encryption algorithm based on semitensor compressed sensing in WBANs. IEEE Internet of Things Journal, 7(1), 734-750.

[27] Lalem, F., Laouid, A., Kara, M., Al-Khalidi, M., & Eleyan, A. (2023). A novel digital signature scheme for advanced asymmetric encryption techniques. Applied Sciences, 13(8), 5172.

[28] Verma, G., Liao, M., Lu, D., He, W., Peng, X., & Sinha, A. (2019). An optical asymmetric encryption scheme with biometric keys. Optics and Lasers in Engineering, 116, 32-40.

[29] Zhang, Q. (2021, January). An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption. In 2021 2nd international conference on computing and data science (CDS) (pp. 616-622). IEEE.

[30] Murad, S. H., & Rahouma, K. H. (2021). Hybrid cryptography for cloud security: Methodologies and designs. In Digital Transformation Technology: Proceedings of ITAF 2020 (pp. 129-140). Singapore: Springer Singapore.

[31] L Guo,A L Wang,S K Li,J P Pang,K Xin,S X Fan & F P Liu. (2019). Authentication algorithm of secure digital halftone watermark based on SM4 algorithm. IOP Conference Series: Materials Science and Engineering,563(5).

[32] Siyang Yu,Kenli Li,Keqin Li,Yunchuan Qin & Zhao Tong. (2016). A VLSI implementation of an SM4 algorithm resistant to power analysis. Journal of Intelligent & Fuzzy Systems,31(2),795-803.

[33] Ali Asim,Khan Muhammad Asif,Ayyasamy Ramesh Kumar & Wasif Muhammad. (2022). A novel systematic byte substitution method to design strong bijective substitution box (S-box) using piece-wise-linear chaotic map. PeerJ. Computer science,8,e940-e940.

[34] Sesibhushana Rao Bommana,Sreehari Veeramachaneni,Syed Ershad & MB Srinivas. (2025). Mitigating side channel attacks on FPGA through deep learning and dynamic partial reconfiguration. Scientific Reports,15(1),13745-13745.