Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Design of finite-domain based cryptographic algorithms in information security and their attack resistance analysis methods

Mingzhi Qi[1,]✉
[1.] *College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, Shandong, 266044, China*

ABSTRACT
Currently, the severity of information leakage is increasing, and attacks and protection against cryptographic devices have become a research hotspot in the field of information security. In order to increase the security of SM4 algorithm structure against side channel attack, the paper focuses on the protection scheme of adding masks to cryptographic circuits to resist DPA attack, and proposes a cipher algorithm design method of finite domain additive coding. Experimentally, it is proved that the additive coding SM4 algorithm used in this paper can correctly and efficiently perform encryption, and the encryption efficiency is improved by 56.54%~82.42% than the general SM4 algorithm. Meanwhile, it has the security against 1st-order and 2nd-order side-channel attacks, and the success rate against attacks reaches 93.67%, which is higher than that of the compared algorithms by 5.34%~21.00%. It also proves that the scheme has high security against side channel attacks and can provide a reliable solution for the information security of wireless LAN.

Keywords: Finite domain, SM4 algorithm, additive mask, cryptographic algorithm, attack resistance, information security

1. Introduction

Cryptographic algorithms are important cornerstones in the field of information security, and their design and analysis play a vital role in protecting users' privacy and confidential data. First of all, the design of cryptographic algorithms should have certain principles [1-3]. One of them is confidentiality, i.e., ensuring that only authorized users can decrypt the ciphertext and obtain the original plaintext information. In order to achieve confidentiality, common cryptographic algorithms include symmetric key encryption algorithms and asymmetric key encryption algorithms [4-6].

✉ Corresponding author.
E-mail address: qimingzhi2024@163.com (M. Qi).
Received 05 March 2024; Revised 10 May 2024; Accepted 05 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-249

Symmetric key encryption algorithms use the same key to encrypt and decrypt plaintext. Its design principle is based on the confidentiality of the key and a secure encryption function. Common symmetric key encryption algorithms are DES, AES and RC4. These algorithms are designed with full consideration of security requirements such as key length, key expansion process, substitution and obfuscation operations [7-10]. Asymmetric key encryption algorithms, on the other hand, use two related keys: public key and private key. The public key is publicly available and can be used to encrypt plaintext, but only the owner of the private key can decrypt the ciphertext. Common asymmetric key encryption algorithms are RSA, DSA and ECC [11-14]. These algorithms are based on the intractability of mathematical problems, such as the decomposition of large prime numbers or the discrete logarithm problem, which ensures the security of the key. Secondly, during the design process of cryptographic algorithms, it is also necessary to consider the possible means of attack adopted by attackers. Common attack means include exhaustive attack, chosen plaintext attack and chosen ciphertext attack [15-18].

Literature [19] examined three encryption algorithms DES, AES and Blowfish based on considering performance metrics such as execution time, memory and throughput. It is pointed out through experiments that Blowfish algorithm has the best performance in comparison to the other two algorithms. Literature [20] emphasizes the importance of cryptography and states that there is a wide variety of cryptographic algorithms used to secure networks, and research on new cryptographic algorithms is currently underway with the aim of developing more advanced communication security techniques. Literature [21] evaluated the efficiency of encryption techniques such as AES and DES based on the security and performance of image data in cloud environment. These algorithms are analyzed and tested through statistical analysis, performance analysis and other analytical tests so as to analyze the efficiency of each algorithm in terms of security and performance. Literature [22] has comparatively analyzed various symmetric encryption algorithms such as DES and AES. The performance of the algorithms is also evaluated by encryption and decryption time, throughput and memory usage. Finally the implementation was possible using Java program on computers based on different operating systems. Literature [23] points out the widespread use of the Internet and the use of encryption algorithms and compares the results with the Arduino benchmarking results based on benchmarking of the most well-known encryption algorithms on the RaspberryPi platform. Literature [24] systematically describes cloud computing and its related issues of data privacy, reliability, and anonymity. It also discusses the various encryption algorithms used to improve the security of the cloud. Literature [25] provides the evaluation of symmetric and asymmetric encryption algorithms using different types of files such as binary files and image files. The evaluation parameters such as encryption time and decryption time are also used to compare these encryption algorithms. The simulation results show the effectiveness of the algorithms. Literature [26] analyzed the comparison of cryptographic encryption algorithms in terms of their key characteristics and explored their performance cost based on some selected key standards and selected algorithms such as CAST128, AES, Blowfish for the study. Literature [27] systematically introduces cryptography and analyzes various cryptosymmetric algorithms such as TripleDES, AES and asymmetric key encryption algorithms such as RSA and analyzes them in terms of their data security capabilities, characteristics and other aspects. Literature [28] compares the performance of three cryptographic algorithms DES, 3DES and AES in e-voting systems. A comprehensive comparison of the effectiveness of various algorithms was carried out by comparing the criteria including speed, key size and the results pointed out that AES performs better than other algorithms.

The study introduces the gray SM4 algorithm from the key extension part that generates the wheel key and the part that realizes the function of encrypting and decrypting the data information. Aiming at the deficiencies that SM4 cryptographic algorithm can be attacked by CPA attack and DPA attack, the additive mask protection scheme based on SM4 cryptographic algorithm is proposed, and the additive mask process of this scheme is derived in detail. On this basis, the diffusion rate is introduced

to simulate and test the SM4 algorithm of additive masking, and several sets of test data are selected to compare and analyze the encryption time consuming between the general SM4 algorithm and the SM4 algorithm of this paper, to explore the encryption performance of the SM4 algorithm of additive masking in this paper. Then the verification experiments of first-order DPA attack and second-order DPA attack are carried out to explore the attack resistance of SM4 algorithm with addition mask. Finally, other cryptographic algorithms are selected to conduct comparative experiments of time-consuming and successful defense to test the superiority of the proposed method in terms of encryption efficiency and security.

2. Finite domain based cryptographic algorithm design

Wireless LAN has been widely used because of the advantages of open interconnection, easy deployment, flexible access, and economic price, but there are also great security risks.WAPI standard is a wireless LAN security protocol that uses SM4 algorithm to ensure the confidentiality of data in wireless LAN. Based on the information security of wireless LAN, the implementation of SM4 algorithm hardware encryption is studied.

2.1. SM4 algorithm

SM4 packet cipher algorithm is abbreviated as SM4 algorithm, which has a packet length and a key length of 128 bits, and consists of two parts, the encryption/decryption algorithm and the key extension algorithm, each of which adopts a nonlinear iterative structure of 32 rounds and has the same arithmetic operation in each round, consisting of the different-or operation, the cyclic-shift operation, and the S-box of the nonlinear device.

2.1.1. Key Expansion Algorithm.For the purpose of illustration of the algorithm, symbol z_2^m is used to denote the m bit vector, symbol $\lll i$ denotes that the 32 bit data is cyclically shifted left by i bits after this operation, and symbol $\oplus$ denotes the different-or operation.

The key expansion algorithm in the SM4 algorithm is used to generate the round key used in the encryption/decryption algorithm. Let the 128-bit input key be $MK=\left(MK_0,MK_1,MK_2,MK_3\right)\in\left(z_2^{32}\right)^4$, after the key expansion algorithm to obtain the wheel key for $rk_i\in z_2^{32}$, $i=0,1,\ldots,31$. Order $K_i\in z_2^{32}$, $i=0,1,2,\cdots,35$, the algorithm to generate the wheel key of the specific process can be expressed as follows:

$$\left(K_0,K_1,K_2,K_3\right)=\left(MK_0\oplus FK_0,MK_1\oplus FK_1,MK_2\oplus FK_2,MK_3\oplus FK_3\right) \tag{1}$$

$$rk_{\mathrm{i}}=K_{\mathrm{i}+4}=K_i\oplus T'\left(K_{i+1}\oplus K_{i+2}\oplus K_{i+3}\oplus CK_i\right) \tag{2}$$

Where, $FK_i\in z_2^{32}$, $i=0,1,2,3$ are the system parameters and its hexadecimal representation is in the form of $FK_0=A3B1BAC6$, $FK_1=56AA3350$, $FK_2=677D9197$, $FK_3=B27022DC$. $i=0,1,2,\ldots,31$, $CK_i\in z_2^{32}$ are the fixed parameters used in the algorithm and T' denotes a synthetic transformation.

The T'-transform is compounded by the nonlinear transform τ-transform and the linear transform L'-transform, i.e., $T'(\cdot)=L'\left(\tau(\cdot)\right)$.

Let the input of the τ-transform be $A=\left(a_0,a_1,a_2,a_3\right)\in\left(z_2^8\right)^4$ and the output be $B=\left(b_0,b_1,b_2,b_3\right)\in\left(z_2^8\right)^4$. The τ-transform can be expressed as follows:

$$\begin{aligned}B&=\left(b_0,b_1,b_2,b_3\right)=\tau(A)\\&=\left(Sbox\left(a_0\right),Sbox\left(a_1\right),Sbox\left(a_2\right),Sbox\left(a_3\right)\right)\end{aligned} \tag{3}$$

Since the output of the nonlinear transform r transform results in the input of the linear transform L' transform, the input of the L' transform can be expressed as $B=\left(b_0,b_1,b_2,b_3\right)\in\left(z_2^8\right)^4$. The L' transform can be expressed as follows:

$$L'(B)=B\oplus(B<<<13)\oplus(B\square<23) \tag{4}$$

2.1.2. Encryption and decryption algorithms.In SM4 algorithm, the encryption/decryption algorithm mainly accomplishes the encryption/decryption operation of plaintext/ciphertext.

Let the 128-bit plaintext used for encryption be $X=\left(X_1,X_2,X_3,X_4\right)\in\left(z_2^{32}\right)^4$, and the 128-bit ciphertext obtained after encryption by the algorithm be $Y=\left(Y_1,Y_2,Y_3,Y_4\right)\in\left(z_2^{32}\right)^4$. The round keys used in the encryption process are $rk_i\in z_2^{32}$ and $i=0,1,\ldots,31$, then the whole process of encrypting the plaintext information can be expressed as follows:

$$\begin{aligned}X_{i+4}&=F\left(X_i,X_{i+1},X_{i+2},X_{i+3},rk_i\right)\\&=X_i\oplus T\left(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i\right)\end{aligned} \tag{5}$$

$$\left(Y_0,Y_1,Y_2,Y_3\right)=R\left(X_{32},X_{33},X_{34},X_{35}\right)=\left(X_{35},X_{34},X_{33},X_{32}\right) \tag{6}$$

where F is the wheel function, R is the inverse transform, and T is the synthetic transform, i.e., $T(\cdot)=L\left(\tau(\cdot)\right)$, which is also composite of the nonlinear transform τ transform and the linear transform L transform, but where the linear transform L transform is:

$$\begin{aligned}L(B)&=B\oplus(B<<<2)\oplus(B<<<10)\\&\oplus(B\square<<18)\oplus(B<<<24)\end{aligned} \tag{7}$$

2.1.3. S-box.As the only nonlinear device in the SM4 algorithm the S-box is an important guarantee for the security of the whole algorithm.

A finite domain multiplication based inverse is utilized to construct the S-box:

$$Sbo\left(x\right)a=I\left(a\cdot A_1+C_1\right)\cdot A_2+C_2 \tag{8}$$

where a is the input of the S-box, $I(\cdot)$ is an arithmetic operation to accomplish the inverse of the elements in the parentheses, and the cyclic matrix $A_1,A_2\in GL(8,2)$ and row vector $C_1,C_2\in GF\left(2^8\right)$ are used to affine transform the input of the S-box so that the input of the S-box is transformed into the finite domain $GF\left(2^8\right)$, which can be expressed as respectively:

$$A_1=A_2=\begin{pmatrix}11100101\\11110010\\01111001\\10111100\\01011110\\00101111\\10010111\\11001011\end{pmatrix} \tag{9}$$

$$C_1=C_2=(1,1,0,0,1,0,1,1) \tag{10}$$

From Eq. (8), it can be seen that in order to take this approach to realize the S-box it is necessary to complete two affine transformations and one inverse operation on the $GF\left(2^8\right)$-domain. The inverse operation on the $GF\left(2^8\right)$-domain is a nonlinear operation, and its realization is relatively more complicated. In order to be able to simplify this operation, the complexity of the operation is simplified

by the method of introducing a new composite domain into which the inverse operation on a finite domain is transformed to be performed.

It is divided into the following three steps:

1) Transform the input of the S-box into the finite domain 8GF(2) by affine transformation, and then transform the elements in this domain into the composite domain by using homomorphic mapping.

2) The inverse operation on the finite domain GF(28) is transformed into the one in the composite domain one at a time.

3) Perform an isomorphic inverse transformation on the result of the inverse operation to transform the result of the inverse operation in the composite domain C into the finite domain F, and then perform an affine inverse transformation to yield the 8-bit output of the S-box:

$$T_1 = \begin{pmatrix} 01011110 \\ 01111100 \\ 11010000 \\ 01010000 \\ 00101110 \\ 11001110 \\ 00001010 \\ 00101101 \end{pmatrix}, T_1^{-1} = \begin{pmatrix} 00110000 \\ 10100100 \\ 10011000 \\ 10110100 \\ 01011010 \\ 10010010 \\ 01011000 \\ 01010001 \end{pmatrix} \tag{11}$$

$$T_2 = \begin{pmatrix} 1000 \\ 1110 \\ 1100 \\ 0001 \end{pmatrix}, T_2^{-1} = \begin{pmatrix} 1000 \\ 1010 \\ 0110 \\ 0001 \end{pmatrix} \tag{12}$$

Where T_1 and T_2 are the homomorphic mappings from finite domain $GF\left(2^8\right)$ to composite domain $GF\left(\left(2^4\right)^2\right)$ and from finite domain $GF\left(2^4\right)$ to composite domain $GF\left(\left(2^2\right)^2\right)$, and T_1^{-1} and T_2^{-1} are their inverses.

From the realization process of the whole S-box, the affine transformations and homomorphic mappings in steps (1) and (3) are linear transformations, which are relatively simple to realize. And the matrices $A_1, A_2 \in GL(8,2)$ and row vectors $C_1, C_2 \in GF\left(2^8\right)$ for affine transformations are already given by Eqs. (9) and (10), while the matrices for homomorphic mapping are given by Eqs. (11)-(12). Step (2) mainly involves the inverse operation on the composite domain, which is much more complicated compared to the general linear transformation, therefore, this paper will focus on the derivation process of the inverse transformation on the composite domain.

The following relation is satisfied between a binary polynomial $a(x)$ and its multiplicative inverse $a(x)^{-1}$ on a finite domain:

$$a(x)a(x)^{-1} = 1 \bmod M(x) \tag{13}$$

where $M(x)$ denotes the irreducible polynomial in that finite domain. Secondly, the transformation of the finite domain $GF\left(2^8\right)$ to the composite domain $GF\left(\left(\left(2^2\right)^2\right)^2\right)$ is obtained from the irreducible polynomial by a quadratic algebraic expansion. $GF\left(\left(2^n\right)^m\right)$

For all arithmetic operations on the composite domain it is necessary to modulo two domains to produce polynomials $Q(y)$ and $P(x)$. where $Q(y)$ is a binary polynomial used to construct the subdomain $GF\left(2^n\right)$, which can be expressed as:

$$Q(y) = \mathrm{y}^n + q_{n-1}\mathrm{y}^{n-1} + \cdots + q_1\mathrm{y} + 1, q_i \in GF(2), i = 1,2,\cdots,n-1 \tag{14}$$

$P(x)$ is a polynomial over the domain $GF\left(2^n\right)$ and is used to construct the composite domain $GF\left(\left(2^n\right)^m\right)$, which can be expressed as:

$$P(x)=mx^m+q_{m-1}x^{m-1}+\cdots+q_1x+q_0, q_i\in GF\left(2^n\right), i=0,1,\cdots,m-1 \tag{15}$$

The irreducible polynomials for quadratic algebraic expansions in finite fields can be generated by the above methods, which can be denoted as respectively:

$$GF\left(2^2\right): x^2+\mu_0x+\theta \tag{16}$$

$$GF\left(\left(2^2\right)^2\right): x^2+\mu_1x+\varepsilon \tag{17}$$

$$GF\left(\left(\left(2^2\right)^2\right)^2\right): x^2+\mu_2x+v \tag{18}$$

In order to simplify the operation, the parameter 1 in Eqs. (16)-(18) are made equal to μ_0,μ_1,μ_2 and taken to be $\theta=\{1\}$, $\varepsilon=\{10\}$, and $v=\{1001\}$. It should be noted that when θ,ε,v is taken to be different values, the corresponding binary mapping matrices are generated differently.

Carrying out the derivation of the inverse operation process on the composite domain, for the inverse operation on the finite domain $GF\left(2^8\right)$ to the composite domain $GF\left(\left(2^4\right)^2\right)$, it is necessary to first transform the element on the finite domain $GF\left(2^8\right)$ to the composite domain $GF\left(\left(2^4\right)^2\right)$ through the homomorphic mapping T_1, and for any element in the composite domain $GF\left(\left(2^4\right)^2\right)$, which can be expressed as a linear polynomial on the $GF\left(2^4\right)$ domain by Eq. (15), therefore, for an element g on the composite domain $GF\left(\left(2^4\right)^2\right)$ can be expressed as $g=\gamma_1y+\gamma_0$, where $\gamma_1,\gamma_0\in GF\left(2^4\right)$, and the inverse of this element is denoted by $d=\delta_1y+\delta_0$, where $\delta_1,\delta_0\in GF\left(2^4\right)$, is then obtained from Eq. (13) and Eq. (18):

$$gd=\left(\gamma_1y+\gamma_0\right)\left(\delta_1y+\delta_0\right)\bmod\left(y^2+y+v\right)=1 \tag{19}$$

Simplify equation (19) and then make the coefficients of the corresponding terms on the left and right sides of the simplified equation equal:

$$\begin{cases}\gamma_1\delta_0+\gamma_0\delta_1+\gamma_1\delta_1=0\\ \gamma_0\delta_0+\gamma_1\delta_1v=1\end{cases} \tag{20}$$

Enter and seek out:

$$\begin{cases}\delta_1=\left(\gamma_1^2v+\gamma_1\gamma_0+\gamma_0^2\right)^{-1}\gamma_1\\ \delta_0=\left(\gamma_1^2v+\gamma_{\gamma_0}+\gamma_0^2\right)^{-1}\left(\gamma_0+\gamma_1\right)\end{cases} \tag{21}$$

2.2. SM4 Algorithm Additive Mask Protection Technique

The S-box in SM4 algorithm is highly susceptible to DPA attack and masking technique as the most common protection technique against power analysis attack. The multiplicative masks of cryptographic algorithms suffer from the defect of zero-value attacks, and the new masking mask scheme has a smaller hardware implementation area compared to other mask protection schemes. The protection scheme adopts a combination of multiplicative and additive masks, which is able to resist zero-value attacks, thus ensuring the security of side-channel power analysis attacks. Therefore, this paper focuses on the additive mask technique in the SM4 algorithm.

By finite fields it is known that any element on the domain $GF\left(2^8\right)$ can be represented by a linear polynomial on the domain $GF\left(2^4\right)$, e.g., a belongs to the elements on the domain $GF\left(2^8\right)$, a_m, a_n belong to the elements on the domain $GF\left(2^4\right)$, which can be expressed as:

$$a = a_m x + a_n \tag{22}$$

When no mask is added to the S-box, the procedure for inverse on the composite domain $GF\left(2^4\right)$ domain is as follows:

$$\left(a_m x + a_n\right)^{-1} = a_m' x + a_n' \tag{23}$$

$$a_m' = a_m \times d' \tag{24}$$

$$a_n' = \left(a_m + a_n\right) \times d' \tag{25}$$

$$d = \left(a_m^2 \times p_0\right) + \left(a_m \times a_n\right) + a_n^2 \tag{26}$$

$$d' = d^{-1} \tag{27}$$

where p_0 is a quadratic expansion polynomial in the domain of $GF\left(2^4\right)$, and a_m' and a_n' are the inverse of a_m and a_n, respectively. The above derivation process does not add a mask when the inverse of the process on the composite domain $GF\left(2^4\right)$, based on this process, when the S-box added when the mask, assuming that the mask is l, $l \in GF\left(2^8\right)$, S-box input is $(a+l)$, and similarly based on the finite domain know that any element on the $GF\left(2^8\right)$-domain can be represented by a linear polynomial on the $GF\left(2^4\right)$-domain as:

$$(a+l) = \left(a_m + l_m\right) x + \left(a_n + l_n\right) \tag{28}$$

The process of adding a mask in the S-box to the inverse over the composite domain $GF\left(2^4\right)$ can be based on the inference equation for the unadded mask in the S-box as follows:

$$\left(\left(a_m + l_m\right) x + \left(a_n + l_n\right)\right)^{-1} = \left(a_m' + l_m'\right) + \left(a_n' + l_n'\right) \tag{29}$$

where a_m', a_n', l_m', l_n' is the inverse of a_m, a_n, l_m, l_n, respectively, and both belong to elements on the domain $GF\left(2^4\right)$:

$$\begin{aligned} a_m' + l_m' &= f_a\left(\left(a_m + l_m\right), \left(d' + l_d'\right), l_m, l_m', l_d'\right) \\ &= a_m \times d' + l_m' \end{aligned} \tag{30}$$

$$\begin{aligned} a_n' + l_n' &= f_b\left(\left(a_m' + l_m'\right), \left(a_n + l_n\right), \left(d' + l_d'\right), l_n, l_m', l_n', l_d'\right) \\ &= \left(a_m + a_n\right) \times d' + l_n' \end{aligned} \tag{31}$$

$$\begin{aligned} d + l_d &= f_c\left(\left(a_m + l_m\right), \left(a_n + l_n\right), p_0, l_m, l_n, l_d\right) \\ &= a_m^2 \times p_0 + a_m \times a_n + a_n^2 + l_d \end{aligned} \tag{32}$$

$$d' + l_m' = f_d\left(d + l_d, l_d, l_d'\right) = d' + l_d' \tag{33}$$

Where the unmasked inference formula is converted to the masked inference formula in the functions f_a, f_b, f_c and f_d are in all belong to the composite domain $GF\left(2^4\right)$, and for the function f_a, replace a_m with $a_m + l_m$ and d' with $d' + l_d'$, respectively. Then there are:

$$\left(a_m + l_m\right) \times \left(d' + l_d'\right) = a_m \times d' + l_m \times d' + a_m \times l_d' + l_m \times l_d' \tag{34}$$

It can be concluded that $a_m \times d' + l_m'$ in the inference equation (26) is one of the parts, and the additive mask produces mask factors that should be eliminated in their entirety, including $\left(a_m + l_m\right) \times l_d'$, $\left(d' + l_d'\right) \times l_m$, $l_m \times l_d'$, and l_d'. Then the correction function f_a is defined as:

$$f_a=(r,s,t,u,v)=r\times s+s\times t+r\times v+t\times v+u \tag{35}$$

where $r=(a_m+l_m)$, $s=(d'+l_d')$, $t=l_m$, $u=l_m'$, and $v=l_d'$ all belong to the elements of domain $GF(2^4)$ compared to $f_a((a_m+l_m),(d'+l_d'),l_m,l_m',l_d')$. The expected result is: $a_m\times d'+l_m'$. Setting $u=l_m'=l_m$, $v=l_d'=l_h$, it can be expressed as follows:

$$\begin{aligned}f_a=&\underbrace{(a_m+l_m)\times(d'+l_h)}_{dm4}\\&+\underbrace{(d'+l_h)\times l_m}_{c_7}+\underbrace{(a_m+l_m)\times l_h}_{c_1}\\&+\underbrace{l_m}_{c_6}+\underbrace{l_m\times l_h}_{c_5}\end{aligned} \tag{36}$$

Similarly correction function f_b is defined as:

$$\begin{aligned}f_b=(r,s,t,u,v,w,x)=&\,r+s\times t\\&+t\times u+s\times x+v+w+u\times x\end{aligned} \tag{37}$$

where $r=a_m'+l_m'$, $s=a_n+l_n$, $t=d'+l_d'$, $u=l_n$, $v=l_m'$, $w=l_n'$, $x=l_d'$ belong to the elements of domain $GF(2^4)$. Then it can be expressed as:

$$\begin{aligned}f_b=&(a_m\times d'+l_m)+\underbrace{(a_n+l_n)\times(d'+l_m)}_{dm5}\\&+\underbrace{(d'+l_m)\times l_n}_{c_8}+\underbrace{a_n+l_n)\times l_m}_{c_2}\\&+\underbrace{l_n}_{c_9}+\underbrace{l_m}_{c_6}+\underbrace{l_n\times l_m}_{c_5}\end{aligned} \tag{38}$$

Similarly correction function f_c is defined as:

$$\begin{aligned}f_c=(r,s,t,u,v,w)=&\,r^2\times t+r\times s+s^2+r\times v\\&+s\times u+u^2\times t+v^2+u\times w+u\end{aligned} \tag{39}$$

where $r=a_m+l_m$, $s=a_n+l_n$, $t=p_0$, $u=l_m$, $v=l_n$, $w=l_d$ are all elements of domain $GF(2^4)$. Then it can be expressed as:

$$\begin{aligned}f_c=&\underbrace{(a_m+l_m)^2\times p_0}_{dm1}+\underbrace{(a_m+l_m)\times(a_n+l_n)}_{dm2}+\underbrace{(a_n+l_n)^2}_{dm3}\\&+\underbrace{(a_m+l_m)\times l_n}_{c_1}+\underbrace{(a_n+l_n)\times l_m}_{c_2}\\&+\underbrace{l_m^2\times p_0}_{c_3}+\underbrace{l_n^2}_{c_4}+\underbrace{l_m\times l_n}_{c_5}+\underbrace{l_m}_{c_6}\end{aligned} \tag{40}$$

In the inverse derivation process, the introduction of the correction function f_a,f_b,f_c,f_d, function dmi, $i=1,2,3,4,5$, said the masked intermediate data, c_j said that after the mask S box in the $GF(2^4)$ domain in the inverse operation of the results of adding the correction term, which $j=1,2,\ldots,9$. The above derivation completes the S-box add mask from the $GF(2^8)$-domain conversion to the $GF(2^4)$-domain on the inverse process, in order to further simplify the computational complexity, the same way to add the mask after the data can be simplified from the $GF(2^4)$-domain operation to the $GF(2^2)$-domain to complete the operation in $GF(2^2)$ multiplication of inverse operations can be equated with the squaring operation, that is:

$$(x+l)^{-1}=(x+l)^2=x^2+l^2 \tag{41}$$

The whole above completes the derivation of the S-box add mask in the SM4 cryptographic algorithm, and next completes the random mask for the whole cryptographic algorithm. Let the input 128bit plaintext as $X=(X_i,X_{i+1},X_{i+2},X_{i+3})$, $X_i,X_{i+1},X_{i+2},X_{i+3}\in(Z_2^{32})$, random mask $128bit\ M=(M_0,M_1,M_2,M_3)$, $M_0,M_1,M_2,M_3\in(Z_2^{32})$, plaintext and mask for dissimilarity to get a new plaintext as:

$$\begin{aligned} X' &= (X_i',X_{i+1}',X_{i+2}',X_{i+3}') \\ &= (X_i\oplus M_0,X_{i+1}\oplus M_1,X_{i+2}\oplus M_2,X_{i+3}\oplus M_3) \end{aligned} \tag{42}$$

The round operation after adding the mask is as follows, where $i=1,2,\ldots,32$:

$$\begin{aligned} X_{i+4}' &= X_i'\oplus T(X_{i+1}'\oplus X_{i+2}'\oplus X_{i+3}'\oplus rk_i) \\ &= X_i'\oplus L(\tau(X_{i+1}'\oplus X_{i+2}'\oplus X_{i+3}'\oplus rk_i)) \\ &= X_i'\oplus L(\tau(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i\oplus M_1\oplus M_2\oplus M_3)) \\ &= X_i'\oplus L(\tau(X_{i+1}\oplus X_{i+2}\oplus X_{i+3}\oplus rk_i)\oplus(M_1\oplus M_2\oplus M_3)A_1A_2) \\ &= X_{i+4}\oplus M_0\oplus L((M_1\oplus M_2\oplus M_3)A_1A_2) \end{aligned} \tag{43}$$

The SM4 cryptographic algorithm with the addition of masks for round operations yields the equation as:

$$\begin{aligned} (X_{i+1}',X_{i+2}',X_{i+3}',X_{i+4}') = (&X_{i+1}\oplus M_1,X_{i+2}\oplus M_2,X_{i+3}\oplus M_3,X_{i+4} \\ &\oplus M_0\oplus L((M_1\oplus M_2\oplus M_3)A_1A_2)) \end{aligned} \tag{44}$$

For the result of the above operation, only the mask needs to be eliminated and the other redundant terms are subjected to the different-or operation.

3. Encryption performance and attack resistance analysis

The performance and attack resistance of the finite domain based SM4 algorithm of this paper is verified through experiments as detailed below.

3.1. Encryption Performance Analysis

3.1.1. Diffusion rate analysis.The encryption function and decryption function only change one of the parameters in each encryption (decryption), and the rest of the parameters have stability, and the ciphertext (plaintext) obtained from multiple encryption (decryption) is analyzed, and the encryption (decryption) diffusion rate is the ratio of the differences on each of the parameters. The lower the probability that the same plaintext is the same as the ciphertext obtained after encryption with different keys, the higher the diffusion rate is, and the better the protection for attackers to understand the attack in the case of local keys. Diffusivity has positive correlation with cipher avalanche effect as well as security. If the diffusion rate of the encryption algorithm is higher than 0.5, then it is in full diffusion state.

Experiments on the encryption performance of the algorithm in this paper to implement the simulation test, encryption group is a different bit decimal number, the number of rounds of the implementation of encryption and decryption operations is 0 to 15 rounds, encryption group to obtain the detection results in the 32-bit decimal number of the conditions of the test results, the encryption process in the plaintext and the key to the ciphertext of the diffusion rate as shown in Figure 1, the ciphertext and the key to the plaintext of the diffusion rate of the decryption process as shown in

Figure 2. When the number of encryption rounds is higher than 7 then the diffusion is good. The key-to-ciphertext reaches full diffusion occurs when the number of encryption rounds is higher than 5. The complete diffusion of key to plaintext occurs when the number of decryption rounds is higher than 6. These results fully prove that the encryption and decryption speed of the algorithm in this paper is fast and the encryption strength is high, and the security is high.

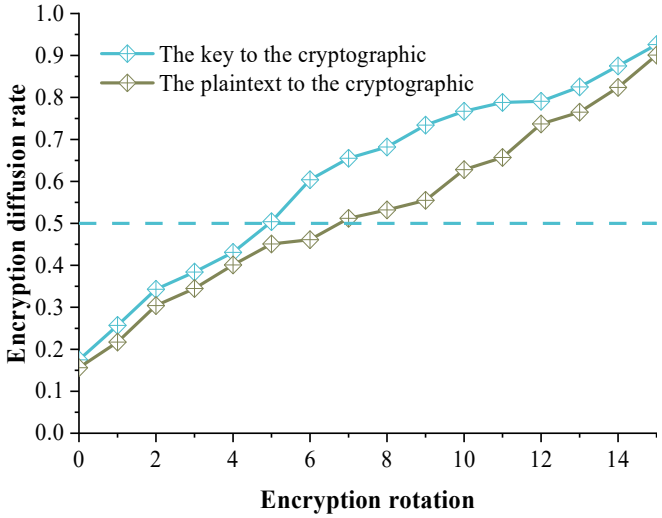


Fig. 1. Diffusion rates of plaintext and key for ciphertext during encryption process

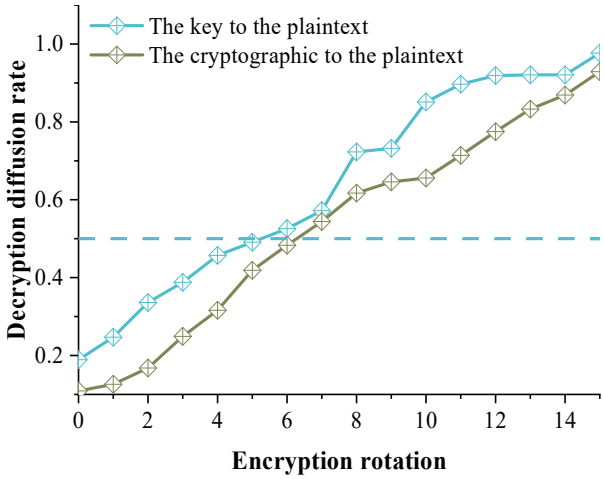
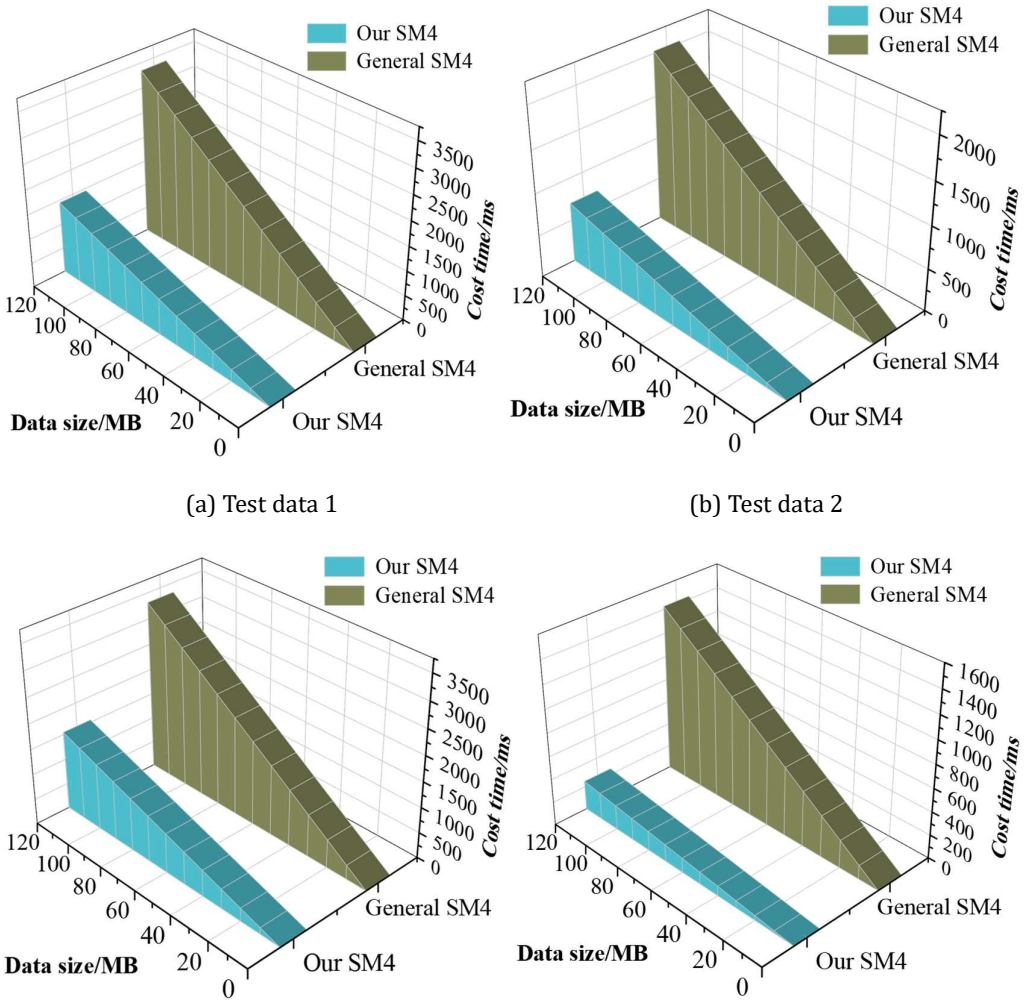


Fig. 2. Diffusion rates of ciphertext and key for plaintext during encryption process

3.1.2. Encryption efficiency analysis.In this paper, several sets of test data ranging from 10~120MB of data length (Test data 1~Test data 4) are constructed, each group of test data contains 100,000 short data, and then the encryption effect of the general SM4 algorithm and the SM4 algorithm of the addition mask in this paper on the experimental platform is compared, and the data encryption test results are shown in Figure 3, and (a)~(d) is the encryption time of the four sets of test data. Both the generalized SM4 algorithm and the SM4 algorithm with additive masks in this paper, the processing time of both for large-scale data increases linearly with the increase of data size.

When encrypting data of the same size, compared with the generalized SM4 algorithm, the encryption rate of this paper's SM4 algorithm on Test data 1 can reach 82.51MBps, and the encryption elapsed time decreases by 56.54% on average, while the encryption rate on Test data 2 can be as high as 169.78MBps, and the encryption elapsed time decreases by 65.13% on average. The performance of this paper's SM4 algorithm on Test data 3 is similar to that of Test data 1, while its performance on

Test data 4 far exceeds that of the generalized SM4 algorithm, with its encryption rate reaching up to 478.47MBps and encryption time consumed dropping by 82.42% on average, which indicates that the data parallel processing advantage brought by this paper's additive masks is very obvious, and it is not only able to speed up the encryption performance of large-scale data, and is also suitable for batch encryption of large-scale data.



(a) Test data 1 (b) Test data 2

(c) Test data 3 (d) Test data 4

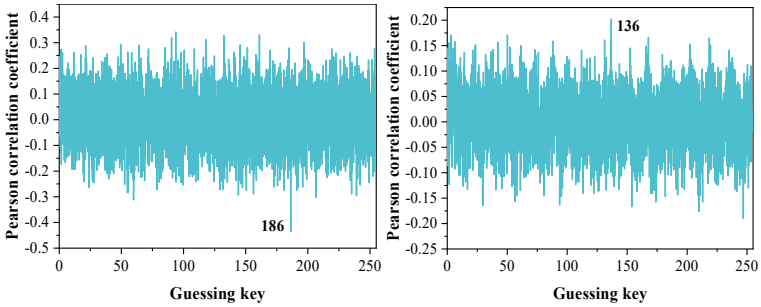
Fig. 3. Data encryption test results

3.2. Anti-Aggression Analysis

3.2.1. 1st Order DPA Attack Authentication.The Hamming distance from the state output of the 8th round to the state output of the 9th round is chosen as the leakage model, and the DPA attack is implemented on the generalized SM4 algorithm and the SM4 algorithm of the additive mask in this paper, with the same initial key used for encryption. The correlation curves are plotted, and the 1st order DPA attack verification results are shown in Fig. 4, with the horizontal coordinates of the guessing key 0 to 255 and the vertical coordinates of the corresponding Pearson correlation coefficients.

The correlation curve for unmasked SM4 is shown in Fig. 4(a), demonstrating the results of the attack on the extended key byte 0. It can be seen that the highest peak of the correlation occurs at position 186, indicating that this 1 byte of the key has been successfully cracked, and in fact only 800 power consumption curves need to be analyzed to successfully crack all 16 bytes of the key. The results

of the energy analysis attack on the SM4 algorithm for the additive mask are very different, as shown in Figure 4(b), which also demonstrates the attack on the extended key byte 0. It can be seen that the peak of the correlation curve is not obvious and the peak occurs at position 136, which does not match the real key and the cracking fails. In fact, even if 100000 power consumption curves are analyzed, any one of the 16 bytes of the key cannot be successfully cracked. The above results prove that the scheme proposed in this paper has the energy to resist the 1st order DPA attack.



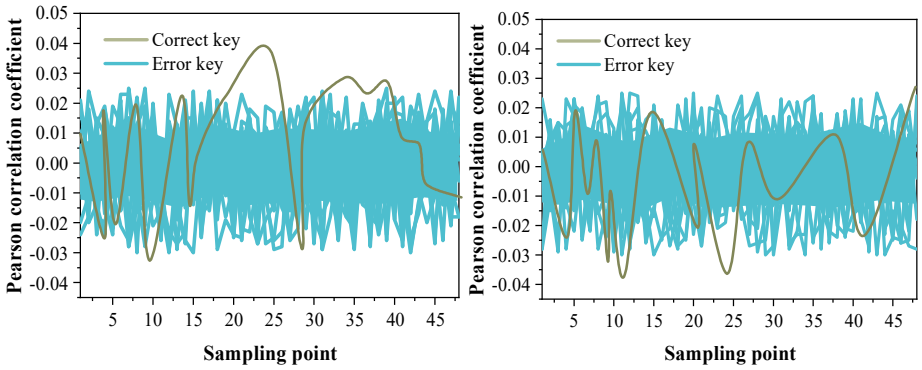
(a) General SM4 (b) Our SM4

Fig. 4. The results of the first-stage DPA attack

3.2.2. 2nd Order DPA Attack Verification.To further illustrate the security of the scheme against side channel attacks, a 2nd order energy analysis attack is implemented in this paper. Using the joint leakage of the 8th round state output and the 9th round state output, the absolute value of the difference is chosen as the preprocessing function for the energy trace. The 2nd order DPA analysis attack is implemented for the 1st order mask SM4 algorithm and the SM4 algorithm of the additive mask in this paper, and the correlation curves are plotted with the horizontal coordinates as the sampling points and the vertical coordinates as the Pearson correlation coefficients.

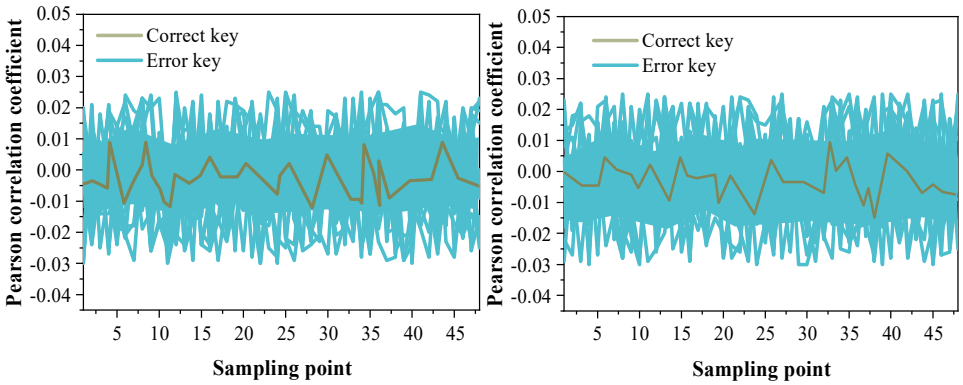
The correlation curves obtained by attacking the 1st order mask SM4 algorithm are shown in Fig. 5, demonstrating the attack on the extended key byte 7 and byte 8. The green curves are the curves corresponding to the correct key and the blue curves are the curves corresponding to the incorrect key. The green curve is significantly more prominent than the blue curve, indicating that this byte key was successfully cracked. Analyzing 20000 power consumption curves, 12 bytes out of 16 byte keys can be successfully cracked.

The results of the DPA analysis attack on the SM4 algorithm with additive masks are very different, and Figure 6 shows the results of the attack on key byte 7 and byte 8. It can be seen that the correlation curve corresponding to the correct key is completely masked in the blue curve and the crack fails. Analyzing 10000 power consumption curves, it also cannot successfully crack any of the 16 bytes of the key. Through the above comparison, it can be concluded that the SM4 algorithm of the additive mask in this paper has a higher resistance to side-channel attack than the 1st-order mask SM4, and can effectively defend against the 2nd-order energy analysis attack.



(a) An attack on byte 7 (b) An attack on byte 8

Fig. 5. The results of the second-stage DPA attack first order mask SM4



(a) An attack on byte 7 (b) An attack on byte 8

Fig. 6. The results of the second-stage DPA attack our SM4

3.3. Comparative analysis of algorithms

The traditional attack function is used to simulate the network attack function and 300 attacks are performed on the network nodes. Comparison of encryption and decryption time of this paper's algorithm, DES algorithm, RSA algorithm and IDEA algorithm is shown in Fig. 7 and Table 1. For the encryption and decryption operations of the data, this paper's algorithm takes the least time to encrypt and decrypt the key compared to DES algorithm, RAS algorithm, and IDEA algorithm, spending 3.15ms, and successfully fights against the attacks the most, with a defense success rate of 93.67%, which is higher than the comparison algorithms by 5.34% to 21.00%. It fully proves that the proposed SM4 algorithm with additive mask is more secure in data information protection.

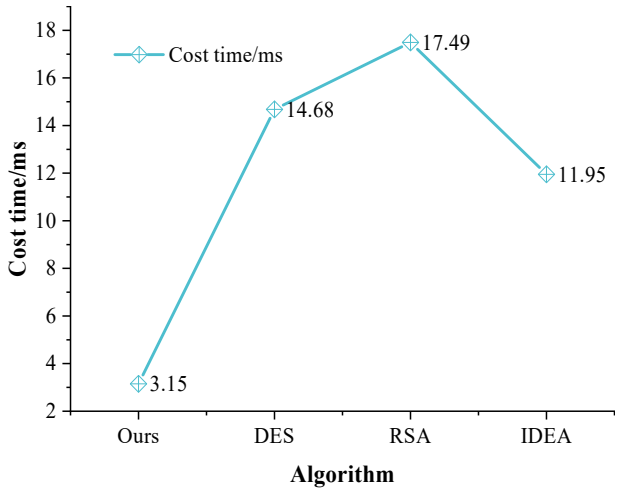


Fig. 7. Comparison of encryption and decryption time

Table 1. Comparison of algorithms

Algorithm	Update time/ ms	Number of attacks	Successful defense	Success rate/%
Ours	38.91	300	281	93.67%
DES	85.22	300	249	83.00%
RSA	47.18	300	265	88.33%
IDEA	55.73	300	218	72.67%

4. Conclusion

Wireless LAN shows a rapid development under the strong support of wireless communication technology and computer technology. At the same time, its security risks are becoming more and more prominent. In this paper, we design the SM4 algorithm protection technology of additive mask for its information security problems, and carry out performance inspection and attack resistance analysis. The main research results are as follows:

1) The encryption rate of this paper's SM4 algorithm with additive mask for different test data sets is higher than that of the generalized SM4 algorithm, and the encryption elapsed time decreases by 56.54%~82.42%, which has high encryption efficiency, less computing time, and can quickly form complex keys to ensure the security of data information.

2) The anti-DPA attack performance of the SM4 algorithm after addition mask is simulated and tested by using first-order and second-order DPA attack respectively, and the protection scheme in this paper can effectively resist the first-order and second-order DPA attack, which verifies the anti-attack property of the SM4 algorithm after addition mask.

3) In the algorithm comparison experiments, the SM4 algorithm with addition mask in this paper spends the lowest encryption and decryption time, and obtains the highest defense success rate of 93.67% in 300 attacks, which is improved by 5.34%~21.00% compared with other cryptographic algorithms.

In this paper, the design and implementation of SM4 cryptographic algorithm is based on finite domain and achieves better performance results. However, there are still a number of aspects for further improvement: (1) The main work of this paper focuses on algorithm optimization, and the exploration of GPU hardware architecture is not sufficient. (2) For the SM4 algorithm, the S-box can be further extended and improved by using the inverse of the composite domain to reduce its area and resource overhead, and the high throughput optimization of other state secret algorithms and homomorphic cryptographic algorithms can be considered in the future.

References

[1] 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).

[2] Abood, O. G., & Guirguis, S. K. (2018). A survey on cryptography algorithms. International Journal of Scientific and Research Publications, 8(7), 495-516.

[3] Najm, H., Hoomod, H. K., & Hassan, R. (2020). A proposed hybrid cryptography algorithm based on GOST and salsa (20). Periodicals of Engineering and Natural Sciences (PEN), 8(3), 1829-1835.

[4] Mazher, A. N., Waleed, J., & MaoLood, A. T. (2021). Developed Lightweight Cryptographic Algorithms for The Application of Image Encryption: A Review. Journal of Al-Qadisiyah for computer science and mathematics, 13(2), Page-11.

[5] Hamouda, B. E. H. H. (2020). Comparative study of different cryptographic algorithms. Journal of Information Security, 11(3), 138-148.

[6] Mousavi, S. K., Ghaffari, A., Besharat, S., & Afshari, H. (2021). Security of internet of things based on cryptographic algorithms: a survey. Wireless Networks, 27(2), 1515-1555.

[7] Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017, August). Comprehensive study of symmetric key and asymmetric key encryption algorithms. In 2017 international conference on engineering and technology (ICET) (pp. 1-7). IEEE.

[8] Sohal, M., & Sharma, S. (2022). BDNA-A DNA inspired symmetric key cryptographic technique to secure

cloud computing. Journal of King Saud University-Computer and Information Sciences, 34(1), 1417-1425.

[9] Nyangaresi, V. O., Ahmad, M., Alkhayyat, A., & Feng, W. (2022). Artificial neural network and symmetric key cryptography based verification protocol for 5G enabled Internet of Things. Expert Systems, 39(10), e13126.

[10] Murtaza, A., Pirzada, S. J. H., & Jianwei, L. (2019, January). A new symmetric key encryption algorithm with higher performance. In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-7). IEEE.

[11] Zhang, J., Yu, Y., Fan, S., Zhang, Z., & Yang, K. (2020). Tweaking the asymmetry of asymmetric-key cryptography on lattices: KEMs and signatures of smaller sizes. In Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part II 23 (pp. 37-65). Springer International Publishing.

[12] Rani, S., & Kaur, H. (2017). Technical Review on Symmetric and Asymmetric Cryptography Algorithms. International Journal of Advanced Research in Computer Science, 8(4).

[13] Dong, X., Randolph, D. A., & Rajanna, S. K. (2020, March). Enabling privacy preserving record linkage systems using asymmetric key cryptography. In AMIA Annual Symposium Proceedings (Vol. 2019, p. 380).

[14] Boicea, A., Radulescu, F., Truica, C. O., & Costea, C. (2017, May). Database encryption using asymmetric keys: a case study. In 2017 21st International Conference on Control Systems and Computer Science (CSCS) (pp. 317-323). IEEE.

[15] Kholidy, H. A. (2019, May). Towards a scalable symmetric key cryptographic scheme: Performance evaluation and security analysis. In 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-6). IEEE.

[16] Uppu, R., Wolterink, T. A., Goorden, S. A., Chen, B., Škorić, B., Mosk, A. P., & Pinkse, P. W. (2019). Asymmetric cryptography with physical unclonable keys. Quantum Science and Technology, 4(4), 045011.

[17] Saravanan, P., & Kalpana, P. (2018). Novel reversible design of advanced encryption standard cryptographic algorithm for wireless sensor networks. Wireless Personal Communications, 100, 1427-1458.

[18] Henriques, M. S., & Vernekar, N. K. (2017, May). Using symmetric and asymmetric cryptography to secure communication between devices in IoT. In 2017 International Conference on IoT and Application (ICIOT) (pp. 1-4). IEEE.

[19] Ramesh, A., & Suruliandi, A. (2013, March). Performance analysis of encryption algorithms for Information Security. In 2013 international conference on circuits, power and computing technologies (ICCPCT) (pp. 840-844). IEEE.

[20] Acharya, K., Sajwan, M., & Bhargava, S. (2013). Analysis of cryptographic algorithms for network security. International Journal of Computer Applications Technology and Research, 3(2), 130-135.

[21] Rahul, B., & Kuppusamy, K. (2023). Efficiency Analysis of Cryptographic Algorithms for Image Data Security in Cloud Environment. IETE Journal of Research, 69(9), 6053-6064.

[22] Lemma, A., Tolentino, M., & Mehari, G. (2015). Performance analysis on the implementation of data encryption algorithms used in network security. International Journal of Computer and Information Technology, 4(4), 711-717.

[23] El-Haii, M., Chamoun, M., Fadlallah, A., & Serhrouchni, A. (2018, October). Analysis of cryptographic algorithms on iot hardware platforms. In 2018 2nd Cyber Security in Networking Conference (CSNet) (pp. 1-5). IEEE.

[24] Khan, S. S., & Tuteja, R. R. (2015). Security in cloud computing using cryptographic algorithms.

International Journal of Innovative Research in Computer and Communication Engineering, 3(1), 148-154.

[25] 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.

[26] Semwal, P., & Sharma, M. K. (2017, September). Comparative study of different cryptographic algorithms for data security in cloud computing. In 2017 3rd international conference on advances in computing, communication & automation (ICACCA)(Fall) (pp. 1-7). IEEE.

[27] Walia, A. G. N. K. (2014). Cryptography Algorithms: A Review. International Journal of Engineering Development and Research, 146.

[28] Soomro, S., Belgaum, M. R., Alansari, Z., & Jain, R. (2019, August). Review and open issues of cryptographic algorithms in cyber security. In 2019 International Conference on Computing, Electronics & Communications Engineering (iCCECE) (pp. 158-162). IEEE.