RESEARCH ON OPTIMIZATION OF HYBRID ENCRYPTION ALGORITHM BASED ON DATA PROTECTION

Yan-Feng LIU[1, *], Qing-Gang LIU[2]

The coming of Big Data Era brings a great challenge to the security protection of data transmission. If the security of data cannot be guaranteed in the process of transmission, it is easy to be interfered with the outside world, which will cause the phenomenon of information delay, loss and so on. In order to avoid the problem of data security, this paper introduces a hybrid encryption algorithm, through the data encryption calculation, reduce the data in the transmission process of security threats. In this paper, 3DES (triple data encryption algorithm) and RSA (public key encryption algorithm) algorithm are chosen as the research object, and the advantages and disadvantages of 3DES and RSA algorithm are compared and analyzed. The 3DES algorithm is characterized by high efficiency, simple algorithm and low system overhead, and it is suitable for large amount of data encryption, while the RSA algorithm has the advantages of high decomposition security, easy implementation and identity authentication, etc., but the process of encryption and decryption takes a long time. In view of these characteristics, this paper optimizes the mixed encryption flow, guarantees the security transmission and management of the key through the RSA algorithm, reduces the security risk in the key distribution process, at the same time, 3DES algorithm improves the efficiency of data encryption and decryption. Experimental results show that the hybrid encryption algorithm performs very well in data encryption and decryption efficiency and improving system security, which provides a new method and idea for the development of data security protection technology.

Keywords: data protection; 3DES; algorithm; hybrid encryption; RSA

1. Introduction

With the rapid development of information technology, data has become an indispensable core asset in modern society [1]. Security threats such as data leakage, illegal access and tampering are becoming increasingly serious, which puts higher requirements on data protection technologies [2]. In recent years, hybrid encryption algorithms have become a research hotspot because of its ability to combine the advantages of different algorithms and provide a higher level of security. In 2021, Hayouni et al. proposed a hybrid encryption system based on AES and RSA, effectively improving the security of data in the cloud [3]. In 2021, Hafsa et al. proposed a hybrid strategy for encrypting RSA encryption keys using

* Corresponding author

[1] School of Information Engineering, Shaanxi Xueqian Normal University, Xi'an 710100, China, e-mail: 10008@snsy.edu.cn

[2] School of Information Engineering, Shaanxi Xueqian Normal University, Xi'an 710100, China, e-mail: 34021@snsy.edu.cn

AES algorithm to solve the problem of the slow encryption speed of RSA algorithm [4], which significantly improved the encryption efficiency. In 2021, Banani's team explored the stability of 3DES algorithm in data encryption and the efficiency of ECC algorithm in key exchange [5]. In 2021, Liu et al. emphasized the importance of lightweight encryption algorithms [6]. In 2021, the hybrid encryption system proposed by Tan et al in the field of blockchain data security highlights the new challenges of data protection in distributed environments [7]. In 2021, Munir et al. explored the application of hybrid encryption algorithms in medical information systems [8], emphasizing the critical importance of data integrity and privacy protection. Data encryption is the main means to ensure information security. Encryption algorithms include both public-key cryptography and symmetric encryption. Symmetric encryption algorithm has the characteristics of high efficiency, simple algorithm, low system overhead, and suitable for large amounts of data encryption, representing algorithms such as 3DES algorithm [9], IDEA algorithm and RC5 algorithm. Easy to implement and decompose is a unique feature of public-key cryptography algorithms, identity authentication and so on. The representative algorithms include RSA algorithm and Elgamal algorithm.

Through comparison, it is found that although the hybrid encryption algorithm shows great potential in many fields, the optimization research for specific application scenarios is still insufficient. In this paper, an innovative hybrid encryption system is proposed. By optimizing the combination of 3DES and RSA algorithms, it aims to achieve comprehensive optimization of encryption efficiency, security and resource utilization [10]. By improving the key management policy and optimizing the encryption process, the efficiency and security of the 3DES algorithm in data encryption are improved. The asymmetric encryption feature of RSA algorithm is used to achieve secure key transmission and management and reduce the security risk in the process of key distribution. The optimized 3DES algorithm is combined with RSA algorithm to form an efficient and secure hybrid encryption system. The performance of the proposed hybrid encryption system in terms of encryption speed, resource consumption and security was evaluated through theoretical analysis, algorithm design and experimental verification, and compared with the traditional encryption system. The research of hybrid encryption algorithm provides a new idea and method for data protection and security technology, which is of great significance for the development of data protection technology.

2. Hybrid encryption algorithm

2.1 Principle of 3DES algorithm

The Triple Data Encryption Algorithm (3DES) is a symmetric encryption algorithm based on DES. An enhanced version of the DES algorithm designed to improve security by increasing the key length and number of encryptions [11]. It uses three 56-bit keys to complete the encryption process through three DES encryption operations. It enhances the security of the DES algorithm by increasing the key length and the number of encryption operations. Specifically, 3DES uses three 56-bit keys (K1, K2, K3) to perform encryption through three consecutive DES encryption processes with a slight modification in the second step. The encryption process can be mathematically expressed as follows:

$$C = DES_{K3}(DES_{K2}^{-1}(DES_{K1}(P)) \tag{1}$$

where P is the plain text, C is the cipher text, DES_{K} denotes the DES encryption function with key K, and DES_{K}^{-1} denotes the corresponding decryption function. Note that the second operation is a decryption step, introducing the concept of “EDE” (Encrypt-Decrypt-Encrypt) mode.

2.2 Principle of RSA algorithm

RSA is a kind of public-key encryption algorithm, whose security mainly depends on the difficulty of large integer decomposition and is widely used. The generation of the public and private keys depends primarily on the given two large prime numbers p and q, as follows:

$$n = p \times q \tag{2}$$

This formula is to multiply two prime numbers p and q, the result n will be as modulus, in the encryption and decryption process will be used.

$$\phi(n) = (\mathrm{p}-1) \times (\mathrm{q}-1) \tag{3}$$

This formula is a Euler’s totient function, and Euler’s totien function $\phi(n)$ is defined for any positive integer n as the number of positive integers that are less than n and that are prime to n. For $n = p \times q$, $\phi(n)$ is (p-1) × (q-1), because all numbers less than n are prime to n except for p and q multiples.

$$\gcd(e, \phi(n)) = 1 \tag{4}$$

This formula is part of selecting the public key *e*. E must be reciprocal with $\phi(n)$ to ensure that there is an inverse of *e* with respect to $\phi(n)$.

$$ed \equiv 1(\bmod \phi(n)) \tag{5}$$

The formula is to compute the modular inverse of *e* with respect to $\phi(n)$, D is an integer, satisfying (*e* × d) mod $\phi(n)$ = 1. This d will be part of the private key. The existence of modular inverse elements is guaranteed by the mutual quality of *e* and $\phi(n)$.

Encryption of a plain text message M (represented as an integer less than n) with the public key is given by:

$$\mathrm{C} = \mathrm{M}^{\mathrm{e}} \bmod n \tag{6}$$

The formula is to decrypt ciphertext C using the private key (d, n). The decryption process is that the d power of C is modularized to n to obtain the original plain text message M.

Decryption of the cipher text C with the private key is:

$$M = C^{d} \bmod n \tag{7}$$

The formula is to decrypt ciphertext C using the private key (d, n). The decryption process is that the d power of C is modularized to n to obtain the original plain text message M.

2.3 Advantages and disadvantages of two algorithms

3DES algorithm and RSA algorithm have their own advantages and disadvantages in key management, encryption efficiency and identity authentication, as shown in Table 1. In order to lay a foundation for algorithm optimization, the advantages and disadvantages of the two algorithms are analyzed and compared.

Table 1

3DES algorithm and RSA algorithm advantages and disadvantages comparison table

	3DES	RSA
Type	Symmetric Encryption	Asymmetric Encryption
Key Management	Difficult to manage and replace	Easy to manage and update
Encryption/Decryption Speed	High efficiency, suitable for large data	Relatively slow, not suitable for frequent encryption
Security	Good for confidentiality, but key distribution is risky	High decomposition security, good for authentication and digital signatures
Key Length	3 × 56-bit (total 168-bit)	Varies, commonly 1024-bit, 2048-bit, or more
Authentication and Digital Signatures	Not directly supported	Easily implemented for authentication and digital signatures
Algorithm Complexity	Moderate, due to repeated DES operations	High, especially for large key sizes
Suitable Applications	Large-scale data encryption	Key exchange, authentication, and digital signatures

In summary, 3DES offers high encryption efficiency suitable for bulk data, while RSA excels in key distribution, authentication, and digital signatures due to its asymmetric nature and high security based on large integer factorization. However, RSA’s encryption and decryption processes are computationally intensive, limiting its use for frequent encryption of large data volumes [12].

2.4 Mixed encryption algorithm

Based on the analysis of 3DES algorithm and RSA algorithm, we analyzed the advantages and disadvantages of two algorithms of RSA encryption and decryption algorithms to propose a faster encryption and decryption which can mix the encryption system authentication and digital signature [13]. The implementation process is as follows:

1) information (plain text) using DES encryption key;

2) using RSA key information before the DES encryption;

3) final mixing of information transfer;
4) the receiver is after receiving the information;
5) with RSA DES decryption key information;
6) and then obtain RSA decryption key information to decrypt the cipher text information;
7) finally we can get the information (plain text) what we want.

3. Algorithm optimization and hybrid encryption algorithm design

3.1 RSA algorithm optimization

From fast exponential law and two grandchildren Theorem aspects of RSA algorithm optimization:

(1) Optimize the RSA algorithm by flash index: utilization of computing power in the RSA system is extremely high, it was simply involves detection, encryption, decryption, and multiplicative inverse. As a key part of the RSA algorithm, time-consuming problems have restricted its wide application. We use the fast exponential algorithms to solve the optimization problem of computing power. Here are a fast algorithm for $x^r \bmod n$ computing, as shown in Table 2.

Table 2

Fast algorithm table for optimization

Serial umber	Step by step
(1)	*a=x, b=r, c=1*;
(2)	When b is not 0, perform the following steps:
(3)	If b is an odd number (b% ! = 0):
(4)	Update c to (c * a)% n, that is, multiply the current a to result c, and take the module n.
(5)	Then (regardless of the parity of b) , a is updated to (a * a)% n, that is, a multiplies by itself and takes a module n, ready for the next iteration.
(6)	Shift b one bit to the right (b >>= 1), which is equivalent to b = b/2, decreasing the value of b.
(7)	When b becomes 0, the loop ends, returning c as the result of x ^ r% n.

The pseudo-code of the Quickexponet function as shown in Table 3.

Table 3

The pseudo-code of the Quickexponet function table

Nmber	Code
1	int Quickexponet(int x, int r, int n) {
2	int a = x;
3	int b = r;
4	int c = 1;
5	while (b != 0) {
6	if (b % 2 != 0) {
7	c = (c * a) % n;
8	}

```
            a = (a * a) % n;
            b = b / 2;
        }
        return c;
    }
```

The code above defines a function called Quickexponet that implements a fast power algorithm (also known as fast exponent). A fast power algorithm is an algorithm for efficiently computing expressions in the form of x r mod n, where x, r, and n are integers. The main advantage of this algorithm is that it can be done in a time complexity much lower than that of directly calculating the r power of x before modularizing n, especially for very large r values.

(2) The optimization algorithms of Chinese Sunzi theorem (Chinese remainder theorem): In order to pursue the efficiency, we cannot just choose a small value, but to have the computations by using the optimization algorithms of Chinese Sunzi theorem. Chinese remainder theorem (one of the most useful theorem of number theories) is a set number of classes remaining to reconstruct an integer within a range of this group is the remaining number of classes of pairwise prime integers modulo obtained [14].

In the data encryption system, the following method tells us about how to speed up calculations modulo: find out the private key *d* and *n* the same order of magnitude slower decrypted. Chinese remainder theorem can be used to reduce its magnitude to simplify the calculation [15].

Remainder Theorem: Let $m_1 \ , m_2 \ ,\dots m_k$ It is a positive integer prime to each other, then $X \equiv a_i \bmod m_i$, i=1,2…k The only positive integer solution to $x = a_1 M_1 y_1 + a_2 M_2 y_2 + a_k M_k y_k \pmod M$.

Among them M=$m_1 \ , m_2 \ ,\dots m_k \ , M_i = M / m_i \ , y_i = M_i^{-1} \bmod m_i \ , i =$ 1,2,3…k.

$$x = m^k \bmod n \tag{8}$$

In this formula, m is the original number to be encrypted or calculated. k is the number of times the cardinality m needs to multiply by itself. n represents the divisor of the result that requires a modulo operation. x is the result of m to the k power taking modules of n.

$$x_1 = m^k \bmod p \tag{9}$$

In this formula, the power operation, m^k is m Times k. Modular operation, Mod p is the remainder obtained by dividing the result of the power operation by p. Assignment, which assigns the result of a modulo operation to x_1, that is, x_1 becomes the result of the modulo operation.

$$x_2 = m^k \bmod q \tag{10}$$

In this formula, the power operation, m^k is m Times k. Modular operation, Mod q is the remainder obtained by dividing the result of the power operation by q. Assignment, which assigns the result of a modulo operation to x_2, that is, x_2 becomes the result of the modulo operation.

$$m^k = (x_1 y_1 q + x_2 y_2 p) \bmod n = x \bmod n \tag{11}$$

This formula describes a specific modular optimization technique that uses Chinese Remainder Theorem -Chinese Remainder THEORCRT CRT) to speed up the DECRYPTIONRSA RSA. x1, x2, y1, y2are the intermediate variables which are used to store some intermediate results of modular exponentiation. They multiply p and q and add them to get m ^ k Mod n.

$$y_1 = q^{-1} \bmod p \tag{12}$$

The formula y1 = q^{-1} mod p means that under the operation of module p, the inverse of q is y1 something. Here, p is a positive integer, q is a number in the range of an integer of p times the remainder of 1(that is, q & LT; p and q is prime to p), and y1 something is an integer satisfying q, y1 ≡1mod p.

$$y_2 = p^{-1} \bmod q \tag{13}$$

The formula y2 = p^{-1} mod q means that under the operation of module q, the inverse of p is y2 something. Here, q is a positive integer, p is a number in the range of an integer of q times the remainder of 1(that is, p& LT; q and p is prime to q), and y2 something is an integer satisfying p, y2 ≡1mod q.

Chinese remainder theorem is widely used in the RSA algorithm encryption and decryption process, this method also requires primes p and q, instead of the key generation algorithm to output private key (d, n) with (d, p, q). Remainder Theorem accelerates the decryption process as shown in Table 4.

Table 4

Remainder Theorem accelerate the decryption process table

Serial Number	Step by step
(1)	Compute $c_p = M^d \bmod p$, $c_q = M^d \bmod q$;
(2)	The use of Fermat's Little Theorem, calculate $c_p = M^{d_1} \bmod p$, $c_q = M^{d_2} \bmod q$, Among them $d_1 = d \bmod (p-1)$, $d_2 = d \bmod (q-1)$;
(3)	Multiplicative inverse calculation by using the extended Euclidean algorithm $c_1 = q^{-1} \bmod p$, $c_2 = p^{-1} \bmod q$;
(4)	Using the Chinese remainder theorem to calculate c, $c = [c_p (q^{-1} \bmod p) q + c_q (p^{-1} \bmod q) p] \bmod n$.

It can be seen using the Chinese remainder theorem to calculate the ecryption modulus (cipher text, the key and the plain text data) is smaller than the original.

(3) Large prime number generation and detection: large prime numbers generated by using random incremental search methods. The existing literature has proven to be less than the number of random incremental search method [16]. We researched a prime number before, to test about lnN integers. It can be seen from the prime number theorem, in the vicinity of a prime number N of the average interval lnN integers. Direct refusal even, in fact, about as long as the test (lnN) / 2 integers. In order to enhance the detection efficiency, we can detect simply excluding even numbers during pre-treatment, and further screening and detecting Improper Primes simply by using a small prime number which is divisible. Several tests can be made by a prime integer close to the probability of 1.0. The whole process seems tedious, while in fact, it is not so complicated, the implementation of this process is to get a new pair of keys.

3.2.3 DES algorithm optimization

Although the 3DES algorithm itself performs well in the encryption and decryption efficiency, in the mixed encryption system, it can still be optimized by the following ways: combining RSA algorithm, using RSA public key to encrypt and transmit the key of 3DES, to ensure security during key transfer [17]. Optimize the integration process of 3DES and RSA, reduce the redundant steps in the process of data transmission, and improve the overall efficiency of encryption.

3.3 Design of hybrid encryption algorithm

Combined with the optimized 3DES and RSA algorithm, a hybrid encryption algorithm is designed, the specific process as shown in Table 5.

Table 5

Design of hybrid encryption algorithm table

Step	Process
Key generation	Generates the RSA key pair (public key, private key) and the 3DES key.
Key Encryption	The 3DES key is encrypted using the RSA public key to obtain the encrypted key (E_K)
Data Encryption	Using the 3DES key to encrypt the plain text data to obtain ciphertext (C_Data).
Data encapsulation	encapsulating the encrypted key (E_K) and ciphe for (C_Data) transmission.
Data transmission	transmission of encapsulated data over a network to a receiver.
Data decryption	The receiver decrypts the encryption key (E_K) using the RSA private key to obtain the 3DES key (K). The cipher text (C_Data) is decrypted using the 3DES key (K) to obtain the clear text data.

3.4 Digital signature process

RSA encryption and decryption algorithm functions not only used for authentication and digital signature aspect. Using a hash function as the authentication code can increase efficiency and reduce the time signature, improve the degree of confidentiality, it can be generated by a hash function to generate a hash value as a message authentication code [18]. Signature process as shown in Table 6.

Table 6

Digital signature process table

Number	Step by step
(1)	The sender will be calculated by the hash function M to produce a hash value h, h = H (M), H (M) is a fixed-length hash value, M is the message to be transmitted;
(2)	The sender private key encryption h, generating a digital signature $S = h^d \bmod n$;
(3)	The recipient and the sender of the message M with the signature S received;
(4)	The recipient decrypts the signature with the public key to obtain H S, H is calculated by h ‘, if h’ = h, then the message has not been transferred with correct message; if h ‘≠ h, the sources described there are problems.

This avoids the problem of sources and intermediaries changing message to occur during a message transmission.

4. Experiment and result analysis

In this chapter, the design of the experiment, the test environment, the test data and the experimental results are described in detail.

4.1 Experimental design

In order to verify the hybrid encryption optimization algorithm based on 3DES and RSA algorithm proposed in this paper, further experimental verification. The main purpose of the experiment is to evaluate the performance of the algorithm in encryption efficiency, resource consumption and security.

4.2 Test environment

Experimental testing environment: i7-6700 CPU, 8G memory, Visual-Studio 2020 development tools, Windows10 operating system, The test data comes from the financial system data of a university.

4.3 RSA encryption and signature efficiency rate test

Table 7

Classical algorithm and improved algorithm to generate large prime time comparative table

Generate digits	The number of produce	Classical algorithm total time (sec)	Average time (sec)	Improved algorithm total time (sec)	Average time (sec)
50	1000	803.15	0.80	903.10	0.90

100	1000	1606.51	1.60	1507.00	1.50
150	1000	36538.94	36.53	26089.45	26.08

There are 512 primes, for example, to test a large number of prime generation efficiency, as shown in Table 7. Improved efficiency of the algorithm and the classical algorithm. The experimental structure shows that the increase of trumpet time is obviously improved with the increase of prime digits. Suffice to generate prime shorten time to improve the efficiency of the RSA signature is feasible. There used 1024 RSA signature prime rate to test and compared them with traditional RSA signatures. As shown in Table 8, improved signature calculation process efficiency is obvious, which can greatly improve the speed of RSA encryption and decryption that can be effectively used in large data transfer message authentication.

Table 8

Improvement signature with the traditional signature RSA time comparative table

Number of trials	Traditional RSA signature Processed (ms)	Improved signature Processed (ms)
1	3350	380
2	3560	375
3	3450	366
4	3390	340
5	3460	320

There are prime numbers used for RSA signature rate test, comparing the traditional and improved RSA signature computation. As shown in Table 8, improved signature computation can increase the speed of encryption and decryption, signature efficiency is very obvious. Therefore, the large data transfer message authentication can effectively use RSA signature [19].

4.4 Efficiency comparative analysis

Using the existing equipment, a test study was conducted on the improved hybrid encryption algorithm and the AES-RSA hybrid encryption algorithm. Through testing with different sizes of information, the improved hybrid encryption algorithm was found to be approximately 7.8% faster than the AES-RSA hybrid encryption algorithm. Experimental results demonstrate that the improved encryption algorithm outperforms the AES-RSA hybrid encryption algorithm. Comparative analysis as shown in Table 9.

Table 9

Encryption time test table

Number of trials	Information byte	AES-RSA hybrid encryption algorithm(ms)	Improved hybrid encryption algorithm(ms)
1	60	295	276
2	90	343	311
3	120	406	385

4	150	454	416
5	200	557	507

4.5 Resource consumption analysis

A comparison of memory usage between the two algorithms as shown in Table 10. The improved hybrid encryption algorithm exhibits approximately 7% less memory usage than the AES-RSA hybrid encryption algorithm. Upon further analysis, it is found that the improved hybrid encryption algorithm outperforms the AES-RSA hybrid encryption algorithm in terms of a certain resource consumption (memory usage).

Table 10

Memory usage comparison table

Algorithm	Data Size(MB)	Memory Usage (MB)	Memory Utilization Rate (%)
AES-RSA hybrid encryption algorithm	10	78	0.95
Improved hybrid encryption algorithm	10	72	0.88

4.6 Performance evaluation of hybrid encryption algorithm

In order to evaluate the hybrid encryption algorithm proposed in this paper, experiments are designed to test its performance in encryption speed, resource consumption and security. The experimental results show that the combination of the optimized RSA algorithm and the 3DES algorithm, the hybrid encryption algorithm can improve the encryption efficiency and resource utilization while maintaining high security [20].

Pecifically, 3DES algorithm optimization makes the data encryption process more efficient, suitable for the rapid encryption of large amounts of data. The asymmetric encryption characteristic of RSA algorithm is used to realize the secure key transmission and management, which reduces the security risk in the process of key distribution. In addition, the encryption and decryption efficiency of RSA is further improved by introducing optimization methods such as fast index algorithm and Chinese remainder theorem algorithm.

4.7 Security analysis

In terms of security, testing with bank payment data has demonstrated that the improved hybrid encryption algorithm combines the advantages of both the 3DES algorithm and the RSA algorithm. The efficiency and simplicity of 3DES ensure the security of data during transmission, while the high decomposition security and authentication capabilities of RSA enhance the security of key transmission and management. Furthermore, by optimizing the key management strategy and encryption process, additional security risks have been further mitigated. To further illustrate the security performance of the improved hybrid encryption algorithm, Security Testing Results as shown in Table 11.

Table 11

Security testing results table

Algorithm	Encryption Strength	Key Management Security	Resistance to Attack
3DES	Medium	Moderate	Moderate
RSA	High	High	High
Improved 3DES-RSA	Very High	Very High	Very High

5. Conclusions

This paper introduces an optimized hybrid encryption algorithm that effectively balances encryption efficiency and security by combining 3DES and RSA. Specifically, the modular exponentiation operations within the RSA algorithm have been optimized using a fast exponentiation algorithm, significantly reducing the computational complexity of both encryption and decryption processes and thus enhancing their processing speeds. Additionally, the Chinese Remainder Theorem (CRT) has been integrated into the RSA decryption process, minimizing the size of intermediate values and further improving decryption efficiency. Furthermore, the process of generating large prime numbers using a random incremental search method has been improved. By excluding even numbers and leveraging smaller primes, the detection time has been successfully shortened, thereby improving the overall encryption efficiency. The hybrid approach combines the efficiency of the symmetric encryption algorithm 3DES with the robust security of the public-key encryption algorithm RSA. This combination addresses the challenges associated with using RSA alone for key transmission and management. To ensure the integrity and authenticity of information transmission and reduce the risk of tampering during transit, RSA encryption and decryption algorithms are introduced for authentication and digital signature purposes. To further strengthen the persuasiveness of the paper, a comprehensive security analysis and resource evaluation have been conducted. This in-depth assessment reveals that the proposed optimized hybrid encryption algorithm not only enhances data encryption efficiency and security but also simplifies the complexity of key management and reduces the computational overhead of encryption and decryption, thereby minimizing system resource consumption and improving resource utilization.

Although this paper has made remarkable achievements in hybrid encryption algorithm, there are still some limitations and possible deficiencies. With the increase of encryption strength, especially in large-scale data encryption, the demand of algorithms for computing resources may increase sharply, which may lead to the decrease of processing speed. The use of long keys increases the need for storage space, especially when large numbers of keys are involved in storage and management. Although this paper proposes using RSA for key management to enhance security, the secure storage and distribution of key is still a problem that needs continuous attention. Whether the security of the encryption system is threatened or not depends on the security of the key in the process of transmission. The complexity of hybrid encryption algorithm is higher than that of single encryption algorithm, which may increase the potential security vulnerabilities and attack surface.

The forthcoming research section touches upon potential areas of development, such as the utilization of hardware acceleration and blockchain integration. By expanding this section to encompass more definite research directions or methodologies, we aim to provide a clearer roadmap outlining future investigations. In this regard, we will further study and develop encryption algorithms with lower computational complexity and higher security to meet the increasing demand of data processing and security. Additionally, we will explore the application of hardware acceleration technology in encryption algorithms, utilizing GPU, FPGA, and other hardware devices to improve the speed of encryption and decryption. The key management system based on blockchain technology will also be studied, enhancing the security of the key by leveraging the characteristics of the blockchain. Furthermore, different encryption algorithms will be further fused and optimized to create a more efficient and secure hybrid encryption system. The application of intelligent algorithms in encryption algorithms will be investigated to improve the intelligence and automation level of the algorithms. Lastly, we will strengthen cross-integration with other fields to build an all-round, multi-level data protection system.

Acknowledgements

Shaanxi Fundamental Science Research Project for Mathematics and Physics (Grant No.23JSY051), Research on Privacy Protection Algorithms in Big Data Computing.

REFERENCES

[1] *M. A. M. Abu-Faraj, Z. A. Alqadi,* Using highly secure data encryption method for text file cryptography, International Journal of Computer Science & Network Security, Vol. **12**, pp. 53-66, 2021.

[2] *C. Xu,* Encrypted transmission method of wireless communication network data based on RSA algorithm, Changjiang Communication, Vol. **37**, Iss. 05, pp. 133-135, 2024.

[3] *H. Hayouni, M. Hamdi,* A novel energy-efficient encryption algorithm for secure data in WSNs,
The Journal of Supercomputing, Vol. **77**, Iss. 5, pp. 4754-4777, 2021.

[4] *A. Hafsa, A. Sghaier, J. Malek, et al.,* Image encryption method based on improved ECC and modified AES algorithm, Multimedia Tools and Applications, Vol. **80**, Iss. 13, pp. 19769-19801, 2021.

[5] *S. Banani, S. Thiemjarus, K. Wongthavarawat, et al.,* A dynamic light-weight symmetric encryption algorithm for secure data transmission via BLE beacons, Journal of Sensor and

Actuator Networks, Vol. **11**, Iss. 1, pp. 2-12, 2021.

[6] *S. Niu, W. Liu, S. Han, et al.,* A data-sharing scheme that supports multi-keyword search for electronic medical records. PLoS ONE, Vol. **16**, Iss. 1, pp. e0244979, 2021.

[7] *C. M. S. Tan, G. P. Arada, A. C. Abad, et al.,* A hybrid encryption and decryption algorithm using Caesar and Vigenere Cipher, In Journal of Physics: Conference Series, Vol. **1997**, Iss. 1, pp. 12-21, 2021.

[8] *N. Munir, M. Khan, T. Shah, et al.,* Cryptanalysis of nonlinear confusion component based encryption algorithm, Integration, Vol. **79**, pp. 41-47, 2021.

[9] *N. Munir, M. Khan, M. M. Hazzazi, et al.,* Cryptanalysis of internet of health things encryption scheme based on chaotic maps, IEEE Access, Vol. **9**, pp. 105678-105685, 2021.

[10] *B. Seth, S. Dalal, D. N. Le, et al.*, Secure cloud data storage system using hybrid Paillie-Blowfish algorithm, Computers Materials & Continua, Vol. **67**, Iss. 1, pp. 779-798, 2021.

[11] *S. J. Sabeena, S. AntelinVijila,* Identification of better encryption algorithm in securing data, Journal of Survey in Fisheries Sciences, Vol. **10**, Iss. 4, pp. 889-896, 2023.

[12] *D. K. Shukla, V. K. Dwivedi, M. C. Trivedi,* Encryption algorithm in cloud computing, Materials Today: Proceedings, Vol. **37**, pp. 1869-1875, 2021.

[13] *D. Zhang, R. Sun,* Research on data security and privacy protection of blockchain cloud computing based on chaotic cipher and RSA algorithm, Electronics, Automation and instrumentation, Iss. 07, pp. 128-132, 2024.

[14] *C. Gao, H. P. Gao, Q. X. Wu,* Research on a weak key attack method of RSA2048, China Security & Protection, Iss. 07, pp. 98-101, 2024.

[15] *P. Kuppuswamy, S. Q. Y. A. K. Al-Maliki, R. John, et al.,* A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm, Bulletin of Electrical Engineering and Informatics, Vol. **12**, Iss. 2, pp. 1148-1158, 2023.

[16] *M. Hamdi, J. Miri, B. Moalla*, Hybrid encryption algorithm (HEA) based on chaotic system, Soft Computing, Vol. **25**, Iss. 3, pp. 1847-1858, 2021.

[17] *S. Ibrahım, A. Zengin, S. Hızal, et al.,* A novel data encryption algorithm to ensure database security, Acta Infologica, Vol. **7**, Iss. 1, pp. 1-16, 2023.

[18] *Y. S. Li, L. Cao, G. L. Zheng, et al.,* Improved RSA dynamic cryptographic accumulator-based anonymous batch authentication scheme for Internet of Vehicles, Computers and Electrical Engineering, Vol. **117**, pp. 109261-109275, 2024.

[19] *G. Viswanath, P. V. Krishna,* Hybrid encryption framework for securing big data storage in multi-cloud environment, Evolutionary Intelligence, Vol. **14**, pp. 691-698, 2021.

[20] *H. Abroshan*, A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms, International Journal of Advanced Computer Science and Applications, Vol. **12**, Iss. 6, pp. 31-37, 2021.