The Application of AES-SM2 Hybrid Encryption Algorithm in Big Data Security and Privacy Protection

Pingyun Huang[1], Guizhou Liao[2], Jianhong Ren[3]*
School of Information Engineering, Jiangxi Vocational and Technical College of Communications, Nanchang, 330013, China[1, 2]
Jiangxi Education Evaluation and Assessment Institute, Nanchang, 330038, China[3]

***Abstract*—In the times of big data, information security and privacy protection have become important issues facing today's society. To address big data's security and privacy problems, research designs and implements a hybrid encryption method using advanced encryption standard algorithms and standard encryption module 2 algorithms for encryption operations. This method utilizes Advanced Encryption Standard encryption algorithms to encrypt plaintext data without calling any encryption libraries. It improves the key extension method and security analysis of Advanced Encryption Standard algorithms. The experimental results show that by changing one key, the confusion range of the improved Advanced Encryption Standard algorithm is 62 ± 6, while the confusion range of the traditional Advanced Encryption Standard algorithm is 63 ± 7. The encryption time of the RSA algorithm is 16.50ms higher than that of Standard Encryption Module 2. The Advanced Encryption Standard scheme improved by Standard Encryption Module 2+ has the fastest decryption speed, followed by RSA+Advanced Encryption Standard scheme, and finally Standard Encryption Module 2+Advanced Encryption Standard scheme. The hybrid encryption algorithm proposed by the research institute can encrypt sensitive information in big data without leaking plaintext information, effectively protecting sensitive information in big data. This scheme can effectively protect sensitive information in big data and provide new ideas for big data in terms of network security and privacy protection.**

***Keywords*—AES; SM2; privacy protection; encryption algorithm; data security**

I. Introduction

In the past few years, as the quick growth of Internet, the popularity of the network has become widespread, and people's lives are also unknowingly changed by the network. Previous data analysis methods can not satisfy people's growing information needs, and big data, as a new type of data analysis method, has rapidly developed into a critical driving force for social advancement due to its advantages of speed, efficiency, comprehensiveness, and massive amount [1-2]. While big data brings enormous benefits and convenience to society, its privacy and security issues are increasingly prominent. Passwords are the most important part of ensuring information security and privacy protection (SPP), and also one of the most important technical means to guarantee data SPP in the era of big data. The commonly used cryptographic algorithms currently include symmetric cryptographic algorithms, asymmetric cryptographic algorithms, and some new cryptographic algorithms developed with big data [4]. Among them, the Advanced Encryption Standard (AES) data packet length is 128 bits, consisting of 128 message digests. It can resist various known and effective attack methods and is currently one of the most secure data encryption standards. Standard Encryption Module 2 (SM2) is a public key cryptography system composed of 8-bit random integers (RSA). The symmetric encryption algorithm is simple and efficient, but vulnerable to attacks. Asymmetric encryption algorithms are secure and efficient, but their decryption speed is slow [5]. To solve the problems of slow speed and short key length in the SM2 cryptosystem, this study focuses on the private data transmission security in cloud computing environments. An improved mixed encryption method of AES and SM2 is proposed, and the security and effectiveness of the method are studied and analyzed.

II. Literature Review

For the SPP in big data itself, various encryption technologies can be used to achieve the purpose of privacy protection. Kumar AS proposed a hybrid soft computing protection and recovery strategy with big data analysis to address the privacy leakage problem caused by malicious propagation in online social networks. By introducing an improved teaching method for optimizing fish schools, abnormal users in the network were classified, and a strategy with deep belief neural networks was proposed to reduce the number of abnormal users. After evaluation, this method had significant advantages in performance indicators such as detection success rate and detection accuracy [6]. The Wu team classified users based on their reactions to data viruses to prevent large-scale damage and privacy leaks caused by viruses spreading on social networks. To limit virus spread and protect data, the company implemented incentives and developed protection and recovery strategies to minimize infected users and increase immune users. Experiments denoted that the proposed model could better describe the spread of viruses on the Internet, and verify the privacy protection mechanism of big data [7]. To protect privacy and avoid the security crisis of CEC in social Internet of Things (IoT) systems, Zhang put forward a privacy protection method with data interference and adversarial training. Through the application of the adversarial pattern generation method with the firefly algorithm, the time complexity of traditional algorithms was decreased by an order of magnitude. The experiment expressed that the model had good anti-interference ability and helped multiple organizations achieve data usage and sentence information in accordance with user privacy protection, data security, and government regulations [8]. Zhang and other researchers proposed a privacy-based blockchain industrial IoT data security sharing model to ensure the secure sharing of resources in the industrial IoT. By using authentication techniques to protect user personal information, encrypted shared resources were stored in the off chain database of the blockchain, and blockchain logging technology was used to track and explain illegal access. Through analysis, the model had good performance [9]. Scholars such as Sachi N M developed an efficient lightweight-integrated blockchain model to solve the deficiencies of the IoT. Through the generated coverage network, well-equipped resources could be merged into a public blockchain to verify dedicated security and privacy. Finally, the model was optimized through lightweight consensus algorithms, certificate free encryption, and distributed throughput management solutions. The experimental results indicated that ELIB exhibited the highest performance under multiple evaluation parameters [10].

AES is a type of group encryption, and its variable key length makes the algorithm more flexible in application, which has been studied by many scholars. Velliangiri et al. proposed a secure multimedia big data content protection system for optimizing and maintaining big data storage. By integrating the key values of AES and SHA-256, novel key values were generated, improving the security level. After verification, this scheme only required a small storage space, had high computational efficiency, and had better performance than existing schemes [11]. To raise the encryption speed of XTS-AES, scholars such as An proposed a technology to achieve high-speed GPU encryption by modifying XTS-AES to a form that is conducive to parallel operations. By analyzing the calculation, multiple operations were replaced with a single table reference, and the parts that can be optimized were given. Then, the process that must be sequentially calculated through table reference technology was skipped for calculation. The results indicated that the method performed well [12]. Researchers such as Jin proposed using a small amount of training data to achieve efficient deep learning-based side channel analysis in response to the problem that threat models cannot collect sufficient data. They trained models with different byte median side channel leakage characteristics using multi byte synchronous training methods. The outcomes indicated that this method had good robustness, and the success rate of recovering AES keys could be raised by 250% [13]. Ueno et al. proposed an optimized encryption standard hardware architecture that supports encryption and decryption (ED), which improved hardware efficiency through multiplication offset. Based on shared key scheduling data paths, it could work in real-time in the proposed architecture. This technology performed AES encryption when block parallelism was not available and could be applied to any type of architecture [14]. The Esfahani team introduced the Evict+Reload attack based on T-table AES implementation in response to cache based attacks. In the preprocessing stage, it was used to analyze all temporal features when using known keys to execute AES. During the utilization phase, complete key bytes were obtained through traditional Evict+Reload attacks. After verification, this technology could resist cache based attacks [15].

In summary, with the continuous increase in data volume and diversification of network application scenarios, single type passwords have become inadequate in dealing with complex network security and privacy issues. Therefore, it is particularly critical to study SPS methods based on AES-SM2 hybrid encryption algorithm that are suitable for big data environments.

III. Security and Privacy Protection based on AES-SM2 Hybrid Encryption Scheme

Research conducts corresponding improvements to the AES encryption algorithm and uses programming to encrypt a small amount of data to enhance its security. For large files, an encryption library with guaranteed encryption speed is used to encrypt the randomly generated AES key using the SM2 encryption algorithm, and the encrypted ciphertext is saved on personal storage devices.

A. AES-SM2 Hybrid Encryption Scheme Design

In the times of big data, massive amounts of data come in two types: structured and unstructured. At the same time, the scale of private data is also enormous, so when encrypting it, it is necessary to balance efficiency and ease of use. Traditional data types are relatively single, requiring less data to be encrypted, stored, managed, and analyzed, and can only be achieved through relatively simple encryption mechanisms. Although traditional encryption methods can ensure high security and integrity, they cannot simultaneously satisfy the demands of efficiency and real-time [16]. However, big data has the features of being massive, diverse, fast, and low value density, making traditional encryption techniques unable to meet its encryption requirements. Currently, the encryption processing of private data in big data can usually be divided into two categories. One is to search for privacy information in the data space through data sampling, objectively reducing the size of the ciphertext, making the encryption of the ciphertext more targeted, and thereby improving the bit rate of the ciphertext [17]. However, in some applications, companies only use data sampling techniques to encrypt important data, making it difficult to prevent attackers from mining sensitive information through other information. The second is to use distributed computing, such as MapReduce, to place a large amount of private data on different machines and encrypt it in parallel, thereby greatly improving the ED speed of private data.

From a security perspective, the study uses a random method to generate initial keys, and then uses the SM2 algorithm to encrypt the randomly generated initial keys, and stores the initial keys in personal storage devices. The AES-SM2 hybrid encryption process is indicated in Fig. 1. Firstly, the file to be saved is transmitted to the hardware client via a personal computer. At this time, the data is transmitted via USB and remains in plaintext state. After completing the transmission and user selection functions, the client encrypts the received file plaintext data using the AES encryption algorithm. At this time, the hardware client randomly generates the initial key used by the AES encryption algorithm. After the encryption is completed, the file is converted from the initial plaintext state to ciphertext state and uploaded to the cloud server of the third-party cloud storage service provider selected by the user. In the previous steps, the randomly generated key was encrypted using the National Secret SM2 encryption algorithm, and the public key used at this time was pre generated and saved on the hardware client. On this basis, using spatial information hiding technology that replaces LSB, the original key is hidden together with the user's photo for future use. In this way, the files that need to be saved will be saved on the cloud server, and the encrypted keywords will be separated from the ciphertext.

In the decryption process, it first downloads the encrypted ciphertext stored on the cloud storage server, then extracts the key to be decrypted from the private storage device. Then the private key stored on the hardware terminal is used to decrypt the SM2 algorithm, and the initial key is obtained. Then the obtained initial key is used to decrypt the data, thereby obtaining plaintext information and restoring the content of the file.

In response to the user's need to encrypt plaintext information, AES-128 with a key length of 128 bits and 10 rounds of encryption is studied for implementation. The implementation is denoted in Fig. 2. In the encryption, the plaintext block and randomly generated sub keys need to be encrypted in one round, and then encrypted through multiple rounds of encryption loops to generate an initial set of keys. The key extension function is utilized to generate 10 sub keys. This method is completed by multiple steps such as byte substitution. The column mixing conversion step does not participate in the last encryption [18].

Fig. 3 is the key extension of the AES encryption algorithm’s schematic diagram. The key extension method of AES encryption algorithm is to directly extend the key, and the algorithm itself has high running efficiency. However, if the attacker only obtains the key once, they can infer that all sub keys, that is, sub keys and seed keys have some equivalence, thereby reducing the security. The fundamental reason why the AES algorithm is efficient is that it uses sub keys directly generated from the original key, and then uses the previous key for the next encryption. Generally speaking, the next sub key can be obtained from the previous one, and the previous sub key can also be inferred from the next one, ensuring that the security of the two sub keys is equivalent. The inference from front to back is the operation performed in ordinary password operations, while the inference from back to front is the action performed when password parsing is cracked. If a method that can ensure efficient reasoning while making it difficult to implement backward inference can be found, it can avoid attack methods such as energy and square.

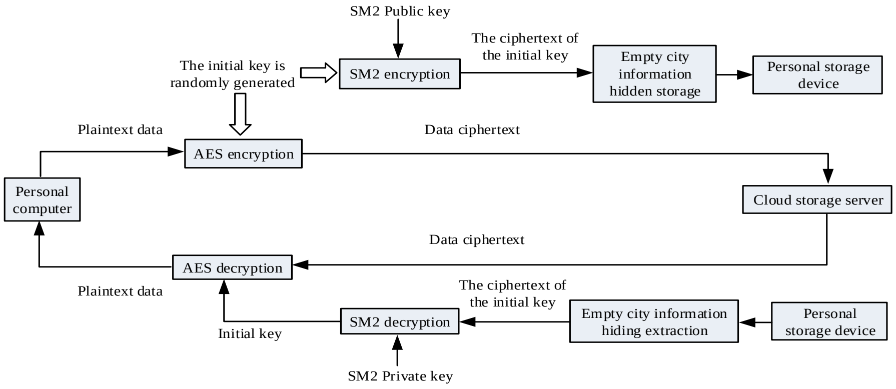


Fig. 1. AES-SM2 hybrid encryption process.

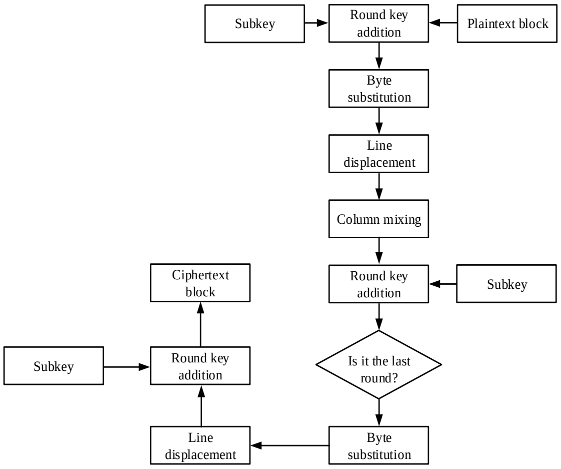


Fig. 2. AES encryption flow chart.

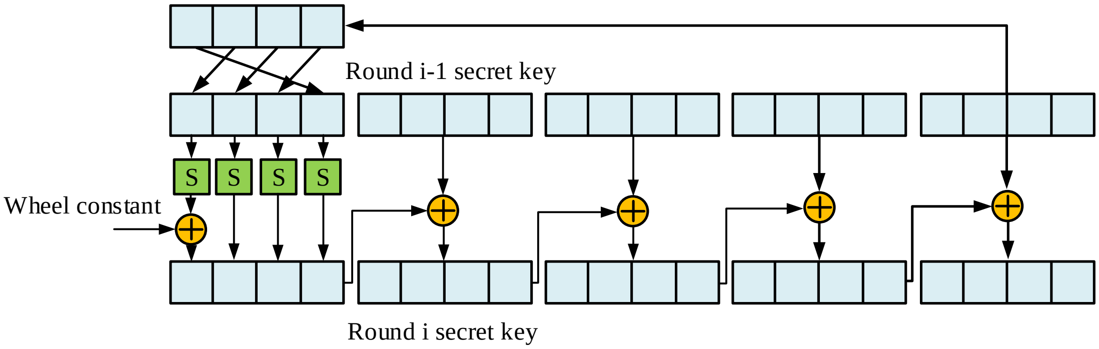


Fig. 3. Schematic diagram of AES key extension.

The round key encryption technique utilizes a mixture of sub keys and columns generated by the system. In the process of generating each round key, the system first randomly generates an initial key, and then uses the key inflation function to calculate each round key. Because both round key encryption and byte replacement transformation perform operations on the state matrix in a column manner, each row shift transformation will break the column configuration and enhance the security of encryption. The encryption operation of AES can be completed through the above steps.

B. Implementation of AES-SM2 Hybrid Encryption Algorithm

The SM2 algorithm is an asymmetric encryption technology based on asymmetric cryptography. When encrypting and decrypting plaintext, two types of keys should be used simultaneously. One is a key that can be made public to the public, and the other is a private key that the decryptor holds and cannot be made public. During data transmission, the encryptor encrypts it with the decryptor's public key, and then decrypts it using their own private key. The encryption communication model of asymmetric encryption technology is shown in Fig. 4. Compared to the AES algorithm, SM2 is more sensitive to key length. Its advantage is that even if it is deciphered, it will not disclose plaintext information [19]. The SM2 cryptosystem has greatly improved security compared to the AES algorithm, but its speed is slower.

Currently, in the big data times, there are many privacy protection methods, among which the advantages of hybrid encryption technology are very obvious. SM4 is a symmetric encryption algorithm with a key length of only 128 bits and poor flexibility. Under the same level of security, AES has faster ED speeds [20]. While meeting security requirements, higher requirements have been raised for the computation and processing of massive data. In view of this, the study combines AES and SM2 elliptic curve encryption methods to achieve big data privacy protection.

The solution of the elliptic curve algorithm mainly relies on the elliptic curve equation, while the SM2 algorithm is built on an elliptic curve over a finite field F_q. When q is an odd prime number, $q = p$, and $p > 2^{191}$ are set, the finite area F_q is called the prime field F_p. When q is a power of 2, which is $q = 2^m$, and m is a prime number larger than 192, the finite region F_q is called the binary extended region F_{2^m}. The expression for SM2 algorithm on prime field F_q is shown in Eq. (1).

$$y^2 = x^3 + ax + b \tag{1}$$

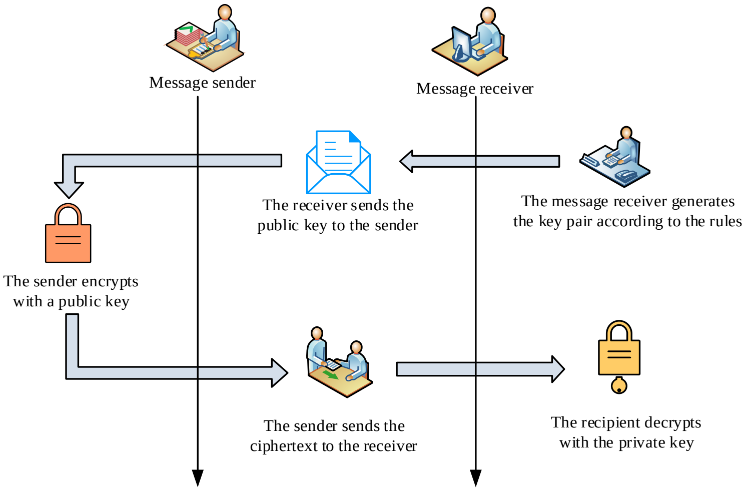

Message sender
Message receiver
The receiver sends the public key to the sender
The message receiver generates the key pair according to the rules
The sender encrypts with a public key
The sender sends the ciphertext to the receiver
The recipient decrypts with the private key

Fig. 4. Encryption communication model of asymmetric encryption technology.

In Eq. (1), $a, b \in F_p$, and a, b satisfy $\left(4a^3 + 27b^2\right) \bmod p \neq 0$. The SM2 algorithm is defined on the binary extended domain F_{2^m} using the Eq. (2).

$$y^2 + xy = x^3 + ax^2 + b \tag{2}$$

In Eq. (2), $a, b \in F_{2^m}$, and $b \neq 0$. Due to the direct relationship between the discrete logarithm problem of elliptic curves and the security of the SM2 algorithm, it is crucial to choose elliptic curves based on finite field F_q security. Assuming the values of the elliptic curve coefficients a and b are given, the unique elliptic curve equation is determined as Eq. (3).

$$y^2 = x^3 - x \tag{3}$$

Fig. 5 shows the affine coordinate graph of Eq. (3), with its base point G set. The parameters involved include the scale q of the finite field F_q and the element $a, b \in F_q$ defined in the elliptic curve $E\left(E_q\right)$. The base point $G = \left(x_G, y_G\right)\left(G \neq 0\right)$ on $E\left(E_q\right)$, where x_G and y_G are the two elements in F_q.

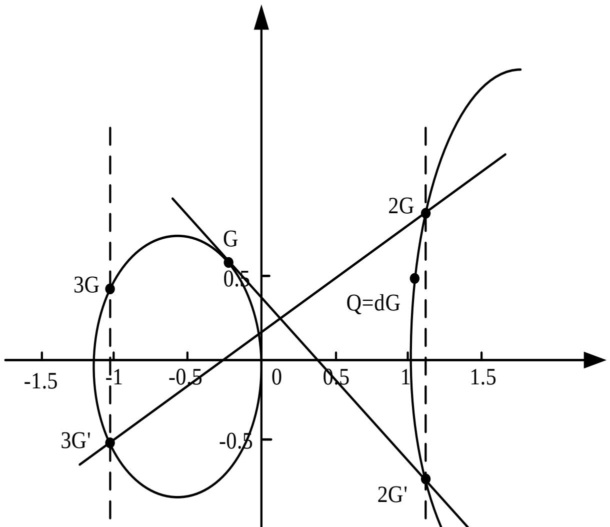


Fig. 5. Affine coordinates of elliptic curves.

An integer d is generated through a random number generator. The integer d is used as the private key, and require d to meet $d \in [1, n-2]$. The d multiplier P of G is calculated using the base point G, as shown in Eq. (4).

$$P = \left(x_p, y_p\right) = [d]G \tag{4}$$

The key pair (d, P) is obtained, where d serves as the algorithm's private key and P serves as the algorithm's public key. Then the point C_1 on the elliptic curve is calculated, as shown in Eq. (5).

$$C_1 = kG = \left(x_1, y_1\right) \tag{5}$$

In Eq. (5), k is a random number, and kG is a multiplication operation. When using elliptic curves for encryption, the main operation is to perform double point operations on the elliptic curve. The point kP_B of the elliptic curve is calculated, as shown in Eq. (6).

$$kP_B = \left(x_2, y_2\right) \tag{6}$$

In Eq. (6), P_B is the public key of user B. And it converts the data types of the horizontal and vertical coordinates x_2 and y_2 into bit strings. When generating the key pair, first is to utilize a random number generator to generate the integer $d \in [1, n-2]$, and calculate Eq. (7).

$$P_B = \left(x_p, y_p\right) = [d]G \tag{7}$$

Because this design uses the SM2 encryption algorithm to encrypt and decrypt the initial key of AES, it is inconvenient for users to generate too many public and private keys for the SM2 algorithm. The study first randomly generates a set of key pairs, and then sets them as a fixed key pair in the subsequent ED process, and stores them on the hardware client for user management and use, as shown in Eq. (8).

$$t = KDF\left(x_2 \| y_2, Mlen\right) \tag{8}$$

In Eq. (8), $Mlen$ is the length of the plaintext bit to be encrypted, $KDF(\;)$ is the key derivation function required for encryption, and the use of the key derivation function is to derive the required key data from the shared secret bit string. The hash algorithm uses the SM3 algorithm. XOR is performed on the corresponding bytes of t and M during the calculation of the intermediate variable t, as shown in Eq. (9).

$$C_2 = M \oplus t \tag{9}$$

In Eq. (9), M represents encrypted plaintext data. The ciphertext is calculated using Eq. (10) again.

$$C_3 = Hash\left(x_2 \| M \| y_2\right) \tag{10}$$

In Eq. (10), $Hash(\ )$ is the password hash function. Finally, the ciphertext is output as shown in Eq. (11).

$$C = C_1 \| C_2 \| C_3 \tag{11}$$

When decrypting the SM2 algorithm, the plaintext C_1 is first extracted from the ciphertext C. And type conversion is performed, then C_2 is extracted and calculated in Eq. (12).

$$dC_1 = (x_2, y_2) \tag{12}$$

Then the data types of the horizontal and vertical coordinates are converted into bit strings, as shown in Eq. (13).

$$t = KDF\left(x_2 \| y_2, Klen\right) \tag{13}$$

If the key data bit string t is all 0, it should stop decryption and report an error. It separates the ciphertext C_2 corresponding to the plaintext in the ciphertext information C and decrypts it into plaintext, as shown in Eq. (14).

$$M_1 = C_2 \oplus t \tag{14}$$

In Eq. (14), M_1 is the decrypted message obtained. The hash value of the decrypted plaintext M_1 is calculated using Eq. (14).

$$u = Hash\left(x_2 \| M_1 \| y_2\right) \tag{15}$$

Finally, it extracts the C_3 bit string from C and compares it with u to see if it is consistent. If there is any inconsistency between u and C_3, an error will be reported and exit. Finally, plaintext M_1 is output.

Performance Testing of Privacy Protection Algorithms based on Hybrid Encryption

Research was conducted to test the ED speed, signature and verification speed, and memory usage of encryption algorithms. The symmetric encryption algorithm and hash algorithm were compared through multiple tests on the same number of blocks as well as different numbers of blocks, while the signature and verification signature speeds of asymmetric encryption algorithms were compared using block sizes that are equivalent in security.

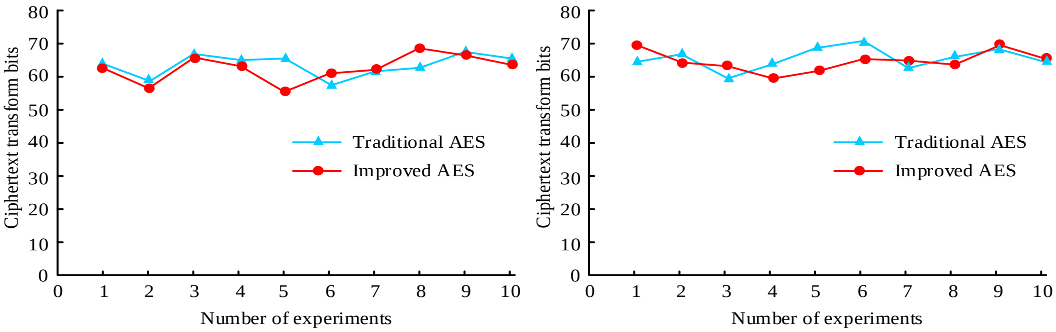
C. *Comparison of ED Time between AES Algorithm and SM2 Algorithm*

Based on Windows and Ubuntu, two development tools, VS Code and PyCharm, were adopted. It was developed using front-end and back-end separation under the B/S architecture. The advantage of this approach was that both the front-end and back-end could be independently repaired, reducing system maintenance costs, improving local performance, and reducing backend pressure. Table I shows Algorithm test environment.

TABLE I. ALGORITHM TEST ENVIRONMENT

Hardware/Software	Version/Model
CPU	Intel Core i7-4720HQ
Memory capacity	16G
Operating system	Windows 10 64bit、Ubuntu
IDE	VS Code 、 PyCharm Professional Ed
Blockchain project	Hyperledger Fabri
Front-end development framework	Vue.js v3.0.5+Eleme
Back-end development framework	Flask v1.1.4

The study compared the scalability and obfuscation performance of AES based on 128bit plaintext, and conducted 10 experiments. The outcomes are indicated in Fig. 6. From Fig. 6 (a), by changing one plaintext, the diffusion range of the improved AES was 63 ± 6, while the diffusion range of the conventional AES algorithm was 64 ± 5. From Fig. 6 (b), by changing one key, the confusion range of the improved AES algorithm was 62 ± 6, while the confusion range of the traditional AES algorithm was 63 ± 7. From this, this method had better security without affecting the original scalability and obfuscation of AES.

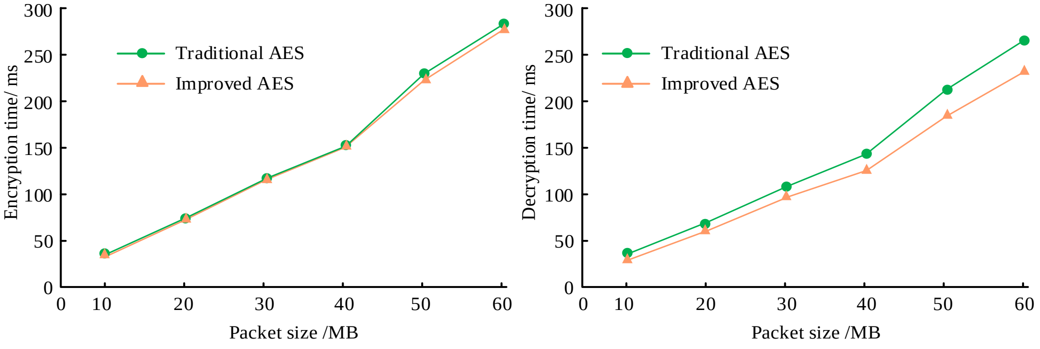


(a) Diffusion comparison

(b) Confounding comparison

Fig. 6. Diffusion and confounding of traditional and improved AES algorithms confusion comparison.

The study also tested the ED time of the AES algorithm. Due to its fast computation and decryption speed, its ED speed was set to around 60MB. The test findings are denoted in Fig. 7. Fig. 7 (a) showcases a comparison of encryption time. From the figure, the improved AES was basically the same as the traditional AES in terms of encryption time. This is because of the fact that the computational complexity of the AES algorithm does not change during column mixing, and it still involves two multiplication operations and four XOR operations. So, using a double symmetric key could enhance the security of passwords without reducing AES encryption speed. Fig. 7 (b) shows a comparison of encryption time. From the figure, compared to before the improvement, the improved AES encryption time was significantly reduced. Due to the use of the optimal column mixing operation, the computational complexity of the inverse column mixing operation during decryption could be greatly reduced. Therefore, the improved AES algorithm not only enhanced key security, but also enhanced decryption speed.



(a) Encryption time (b) Decryption time

Fig. 7. Comparison of encryption and decryption time of AES algorithm

Performance testing was conducted on SM2, as it was only used for ED of symmetric keys, the data that needs to be encrypted was very small, and a plaintext data length of 128 bits was used as the length of the plaintext data. Table II compared the encryption speeds of SM2 and RSA algorithms under the same security conditions. From the table, under the same security, the encryption time of RSA algorithm was 16.50 milliseconds higher than the encryption speed of SM2. Among them, the RSA algorithm with a length of 3072 bits took about 1204 milliseconds to generate a key pair, while the 256 bit key pair of SM2 took about 360 milliseconds to generate. The RSA algorithm had a higher key generation speed and storage space than SM2. So, overall, it was reasonable to use the SM2 algorithm to encrypt symmetric keys.

TABLE II. COMPARISON OF SM2 ENCRYPTION TIME WITH RSA ENCRYPTION TIME

Class number	SM2	RSA
First group /ms	32.6	14.2
The second group /ms	33.8	13.3
The third group /ms	34.1	15.6
The fourth group /ms	33.5	16.5
The fifth group /ms	33.2	15.1
Average time /ms	33.44	14.94

D. *Performance Testing of Hybrid Encryption Algorithms*

The study selected blocks of 16, 64, 256, 1024, 8192, 16384 for experiments and compared the differences in memory usage size among these algorithms. The size of memory usage was determined by two factors: average memory size and runtime. The experiment outcomes are denoted in Fig. 8. In the respect of memory utilization, SM4 and AES had similar storage space, while 3DES was larger than other methods. So, in the encryption module, SM4 or AES-128 was chosen as the symmetric encryption algorithm.

The study tested the ED time of SM2 and improved AES, and also compared SM2+AES and RSA+AES. The experiment outcomes are denoted in Fig. 9. From Fig. 9 (a), the RSA+AES algorithm was relatively fast in encryption time, while the SM2+AES algorithm and SM2+AES algorithm had little difference in encryption time. Due to the fact that the key pair generation speed in RSA algorithm is much lower than SM2, and the storage space required by RSA algorithm is also much larger than SM2, RSA+AES did not have significant advantages compared to the other two methods. Compared with the SM2+AES algorithm, the SM2+AES algorithm used a double symmetric key to enhance the security of the key without reducing the encryption speed. From Fig. 9 (b), the decryption speed of the SM2+improved AES scheme was the fastest, followed by the RSA+AES scheme, and finally the SM2+AES scheme. Therefore, the improvement of the AES algorithm was effective. In summary, the hybrid cryptosystem proposed by the research institute had significantly improved security, ED speed, and other aspects.

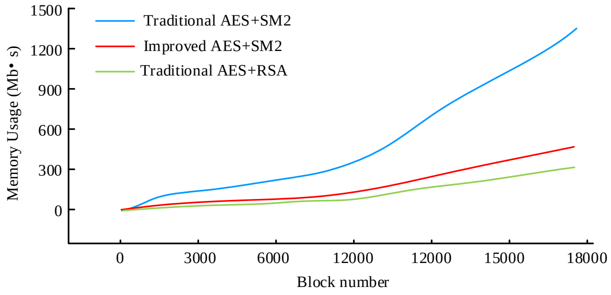
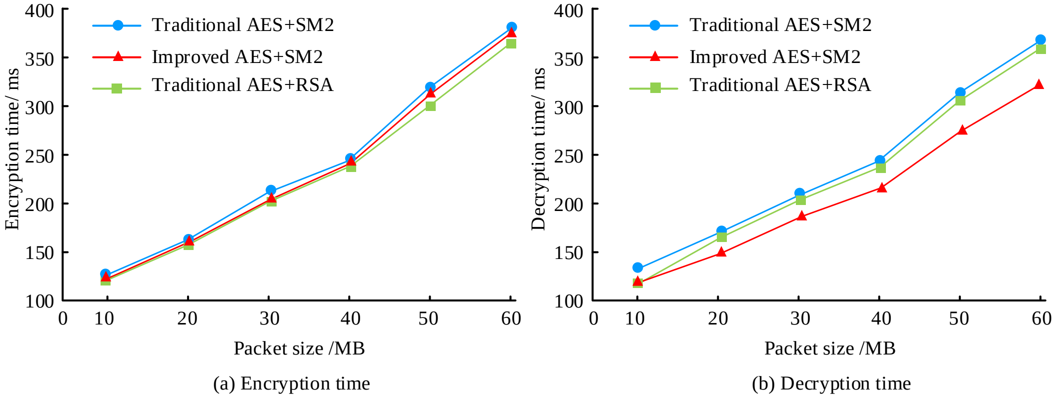


Fig. 8. Memory usage comparison.



(a) Encryption time

(b) Decryption time

Fig. 9. Comparison of encryption and decryption time of hybrid algorithms.

The study tested the throughput of the system within 100 minutes, and the results are shown in Fig. 10. When the overall number of transactions was small, the throughput of the system was only slightly higher than that of the Bitcoin system. When the number of transactions tended to stabilize, the throughput of the system would fluctuate according to probability, about twice that of the Bitcoin system. At the same time, when there was a high demand for practical applications, methods such as reducing sampling values or mining difficulty could be used to improve the average processing speed of the system, but it would also weaken the constraints on high computing nodes. Overall, the system placed greater emphasis on data security and confidentiality, and in terms of performance, it could basically meet practical needs.

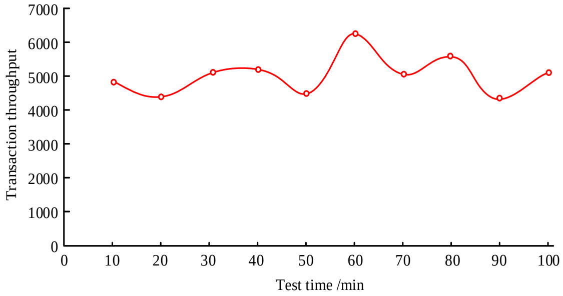


Fig. 10. System throughput.

IV. Conclusion

To guarantee the security and privacy of big data in cloud computing, encryption technology can be applied to ensure the accuracy and confidentiality of data in the cloud. This study focused on the privacy and security issues of big data transmission and storage in the cloud environment, deeply analyzed the merits and demerits of various existing encryption methods. An encryption method that combines an asymmetric encryption algorithm was designed based on the national secret SM2 with an improved AES encryption algorithm. While ensuring the security of big data in the cloud environment, the ED speed was accelerated to ensure the security of data. The outcomes indicated that the diffusion range of the improved AES was 63 ± 6, while the diffusion range of the conventional AES algorithm was 64 ± 5. This method had better security without affecting the original scalability and obfuscation of AES. When the AES algorithm performed column mixing, its computational complexity did not change, and it still performed two multiplication operations and four XOR operations. The encryption time of the RSA algorithm was 16.50ms higher than that of SM2. The AES scheme improved by SM2+ had the fastest decryption speed, followed by the RSA+AES scheme, and finally the SM2+AES scheme. Therefore, the improvement of the AES algorithm was effective. In summary, the hybrid cryptosystem proposed by the research institute has significantly improved security, ED speed, and other aspects. The method proposed by the research institute is only to ensure the transmission of privacy information in the cloud environment. Saving privacy information in ciphertext to the cloud is, in a sense, a security guarantee, but there are also potential risks. Next, it can fully leverage the advantages of cloud computing by combining public and private clouds to ensure the storage security of user privacy data.

Fundings

The research is supported by 2022 Jiangxi Provincial Department of Education Science and Technology Research Project "Research and Development of Strong and Weak Electricity Management Network Platform Based on BIM+WebGL" (Project Number: GJJ2205220).

References

[1] Wen Y P, Liu J X, Dou W C, Xu X L, Cao B Q, Chen J J. Scheduling workflows with privacy protection constraints for big data applications on cloud. Future Generation Computer Systems, 2020, 108(13):1084-1091.

[2] Usman A M, Abdullah M K. An assessment of building energy consumption characteristics using analytical energy and carbon footprint assessment Model. Green and Low-Carbon Economy, 2023, 1(1): 28-40.

[3] Aryavalli S N G, Kumar G H. Futuristic vigilance: Empowering chipko movement with cyber-savvy IoT to safeguard forests. Archives of Advanced Engineering Science, 2023, 1(8): 1-16.

[4] Liu K, Sun Y, Yang D. The administrative center or economic center: Which dominates the regional green development pattern. A case study of shandong peninsula urban agglomeration, china. Green and Low-Carbon Economy, 2023, 1(3), 110-120.

[5] Yang P, Xiong N N, Ren J. Data security and privacy protection for cloud storage: A survey. IEEE Access, 2020, 8(99):131723-13140.

[6] Kumar A S, Revathy S. A hybrid soft computing with big data analytics based protection and recovery strategy for security enhancement in large scale real world online social networks. Theoretical computer science, 2022, 927(12):15-30.

[7] Wu Y K, Huang H Y, Wu N Y, Wang Y, Bhuiyan M Z A, Wang T. An incentive-based protection and recovery strategy for secure big data in social networks. Information Sciences, 2020, 508:79-91.

[8] Zhang P, Wang Y, Kumar N, Jiang C X, Shi G Wei. A security and privacy-preserving approach based on data disturbance for collaborative edge computing in social IoT systems. IEEE Transactions on Computational Social Systems, 2021, 9(1):97-108.

[9] Zhang Q, Li Y, Wang R, Liu L, Tan Y, Hu J J. Data security sharing model based on privacy protection for blockchain-enabled industrial Internet of Things. International Journal of Intelligent Systems, 2020, 36(1):94-111.

[10] Sachi N M, Ramya K C, Rani S, Gupta D, Shankar K, Lakshmanaprabu S K, Khanna A. An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy. Future Generation Computer Systems, 2020, 102(2):1027-1037.

[11] Velliangiri S, Naga R D G. Hybrid crypto techniques for secured multimedia big data content protection system (SMBDCPS). International Journal of E-Collaboration, 2021, 17(2):1-21.

[12] An S, Seo S C. Designing a new XTS-AES parallel optimization implementation technique for fast file encryption. IEEE Access, 2022, 36(10):25349-25357.

[13] Jin C, Zhou Y. Enhancing deep-learning based side-channel analysis through simultaneously multi-byte training. The Computer Journal, 2022, 66(11):2674–2704.

[14] Ueno R, Morioka S, Miura N, Matsuda K, Nagata K, Bhasin S, Mathieu Y, Graba T, Danger J L, Homma N. High throughput/gate AES hardware architectures based on datapath compression. IEEE Transactions on Computers, 2020, 69(4):534-548.

[15] Esfahani M, Soleimany H, Aref M R. Enhanced cache attack on AES applicable on ARM-based devices with new operating systems. Computer Networks, 2021, 198(27):407-415.

[16] Cao H, Wu Y, Bao Y, Feng X, Wan S, Qian C. UTrans-Net: A model for short-term precipitation prediction. Artificial Intelligence and Applications. 2023, 1(2): 106-113.

[17] Ly A, El-Sayegh Z. Tire wear and pollutants: An overview of research. Archives of Advanced Engineering Science, 2023, 1(1): 2-10.

[18] Garai S, Paul R K, Kumar M. Intra-annual national statistical accounts based on machine learning algorithm. Journal of Data Science and Intelligent Systems, 2023, 2(2): 12-15.

[19] Shi J, Yu Q, Yu Y, Wang L H, Zhang W Z.P rivacy protection in social applications: A ciphertext policy attribute-based encryption with keyword search. International journal of intelligent systems, 2022, 37(12):12152-12168.

[20] Zhang Q, Zhang X, Wang M, Li X H. DPLQ: Location-based service privacy protection scheme based on differential privacy. IET information security, 2021, 15(6):442456.