International Journal of Science and Engineering Applications
Volume 14-Issue 01, 52 – 57, 2025, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1401.1012

Research and Application of AES Algorithm in Symmetric Encryption

Zhonghao Zheng
School of Electronic
Information and Electrical
Engineering
Yangtze University
Jingzhou, China

Abstract: With the rapid development of information technology, data security issues are becoming more and more prominent, and symmetric encryption, as a common encryption method, has received widespread attention. And among the symmetric encryption algorithms, AES is favored for its high security, efficiency and reliability. This paper firstly introduces the basic principles and technical details of AES algorithm and discusses its specific operation in the process of data encryption and decryption. AES algorithm adopts the substitution-replacement network structure and encrypts the data through multiple rounds of iterative operations, and it has three kinds of key lengths of 128-bit, 192-bit, and 256-bit, which can be selected according to the security requirements, and it has a higher level of security. Then this paper discusses the AES algorithm mainly has four cryptanalysis methods, namely, brute-force cracking, timing attack, linear analysis and differential analysis, as well as the encryption and decryption experiments of the AES algorithm, and the encryption and decryption experiments of the text information using OpenSSL, to evaluate the performance of the encryption and decryption process, and to provide the relevant performance indexes and analysis results. Finally, this paper discusses the application of AES algorithm in various fields. In the financial field, AES is used to encrypt financial transaction data and customer authentication information. In the field of Internet of Things (IoT), AES protects the communication security between IoT devices. In military applications, AES protects the security of military communications and data. These application cases fully demonstrate the importance and wide application of AES in various fields. In summary, this thesis systematically introduces the research and application of AES algorithm in symmetric encryption based on AES algorithm, deeply analyzes the principle, technical details, and applications in various fields of AES algorithm, and puts forward the suggestions for the future research and application. AES algorithm plays an important role in information security protection, and it is of great significance to protect the security and privacy of data.

Keywords: AES Algorithm; Symmetric Encryption; Encryption and Decryption; Cryptanalysis Methods; Application Fields

1. INTRODUCTION

With the rapid development of information technology, the problem of information security has become increasingly prominent. As an important means of information security, symmetric encryption plays a key role in protecting data confidentiality. Limitations of Traditional Encryption Algorithms Early symmetric encryption algorithms, such as DES, have certain limitations in terms of security and efficiency. Therefore, more advanced algorithms need to be researched to meet the growing security needs. Improvement in computing power, with the increasing processing power of computers, attackers are also able to break traditional encryption algorithms more easily. As a result, there is a need to develop stronger and more secure encryption algorithms such as AES. Requirements of regulations and standards Many industries and organizations have developed regulations and standards related to information security that require the use of strong encryption algorithms to protect sensitive information. Together, these factors have led to research on AES algorithms to meet the needs of modern information security.

From 2001 to 2005, domestic scholars began to study AES algorithms and published a series of papers in the field of cryptography, covering the theoretical basis, analysis and improvement of AES algorithms, etc[1]. From 2006 to 2010, with the increase of information security requirements, domestic attention began to focus on the practical application and performance optimization of AES algorithms. Some research focused on hardware implementation, acceleration algorithm and security analysis of AES algorithm, etc[2]. From 2011 to 2015, domestic scholars made some breakthroughs in AES algorithm research and proposed some new encryption schemes and optimization strategies to improve the performance and security of AES algorithm in practical applications[3]. From 2016 to 2020, with the big data and Internet of Things (IoT) technology's rapid development, domestic scholars began to explore the application of AES algorithms in these emerging fields, such as IoT security and cloud computing security. At the same time, the research on AES algorithm in the field of mobile communication and network security is also gradually deepened[4]. From 2021 to 2024 (as of now), AES algorithm is still one of the focuses of cryptography research in China.

From 2000 to 2010, the AES algorithm, as the national standard of the United States, attracted extensive attention from the international cryptography research community. Foreign scholars mainly focused on the security analysis, attack model and countermeasure strategy of AES algorithm, etc[5]. From 2011 to 2015, with the rise of emerging technologies such as cloud computing, Internet of Things and mobile communication, foreign scholars began to study the application and optimization of AES algorithm in these fields. At the same time, new attack methods such as side channel attack and quantum computing attack of AES algorithm are studied[6]. From 2016 to 2020, foreign scholars show a diversified trend in the research direction of AES algorithm. On the one hand, the hardware implementation, performance optimization and security enhancement of the AES algorithm continue to be explored; on the other hand, the cross-application of cryptography and the exploration of emerging fields become hot spots of research[7]. From 2021 to 2024 (as of now), the AES algorithm remains one of the important topics in international cryptography research. Foreign scholars continue to focus on the security, performance and applicability of the AES algorithm, and apply it to various emerging fields, such as blockchain and artificial intelligence security. In summary, domestic and foreign research on AES algorithm began with its birth and continues to this day. In the past decades, AES algorithms have made great progress in theoretical research, security analysis, performance optimization and practical application, and have made important contributions to the development and progress of the information security field[8].

With the increasing network security threats, the security requirements for data encryption algorithms are also increasing. AES algorithm, as a widely recognized symmetric encryption algorithm, has received more attention and research. With the development of hardware technology, especially the emergence of hardware gas pedals and specialized chips for encryption processing, it has become a hot topic to study how to implement efficient AES encryption algorithms on hardware to improve the encryption speed and energy-efficiency ratio. The AES algorithm is widely used in many standards and protocols, such as TLS and IPsec. Therefore, research on AES algorithms also involves collaboration with standardization organizations and updating and improving the standards. With the development of emerging technologies, such as quantum computing and artificial intelligence, new challenges are posed to the security of traditional encryption algorithms. Therefore, it has become an important direction to study how to improve the AES algorithm's resistance to quantum attacks and its security in the AI environment. The increasing demand for data security in finance, medical care, e-commerce and other fields has driven the research and improvement of AES algorithms in practical applications to meet the changing security needs. Taken together, the background of foreign research on AES algorithms in the field of symmetric encryption is mainly influenced by various factors such as the improvement of security needs, hardware technology development, standardization and normalization, emerging technology challenges and practical application needs. The current direction of AES algorithm research: (1) research on new types of ciphers; (2) research on the principles and guidelines of comprehensive assessment of cryptographic security; (3) research on the implementation of ciphers including software optimization hardware implementation and special chips, etc.; and (4) research on the analysis of the AES and its application.

2. THEORY AND METHODS

2.1 Symmetric encryption fundamentals

Symmetric encryption is an encryption technique used in the field of information security, and its basic concept involves the principle of using the same key for both encryption and decryption processes. In symmetric encryption, the sender uses a key to encrypt the message to form a cipher text and the receiver can decrypt the message to plain text only by using the same key. The key is the same for both the encryption and decryption process and hence it is called symmetric key. The basic principle of symmetric encryption is to encrypt the message using the key so that unauthorized users cannot understand the content of the encrypted message. Only the sender and receiver who know the key can perform the encryption and decryption operations correctly. This design of symmetric encryption makes the data protected from unauthorized access and theft during transmission.

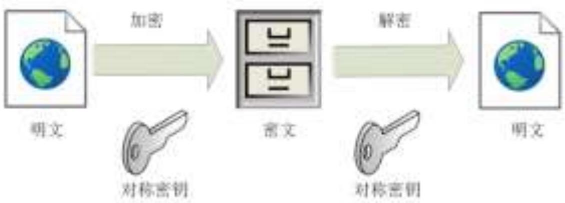

Figure 1 Basic principle diagram of symmetric encryption

The basic principle diagram of symmetric encryption is shown in Figure 1, where a plaintext message is encrypted with a symmetric key to form a ciphertext, and the ciphertext is then decrypted with the same key to form a plaintext message. One of the advantages of symmetric encryption is that it is very fast because the encryption and decryption processes use the same key and do not require overly complex mathematical operations. This makes symmetric encryption efficient in scenarios such as network communication and data transfer. However, symmetric encryption also has some drawbacks, not the least of which is the key management problem. Because the key needs to be shared between the sender and the receiver, the security and management of the key becomes one of the challenges faced by symmetric encryption. To solve the key management problem, symmetric encryption is usually used in combination with other techniques, such as asymmetric encryption to ensure security when transferring keys. Asymmetric encryption uses a pair of keys, public and private, where the public key is used for encryption and the private key is used for decryption. In this way, the sender can encrypt the symmetric key using the receiver's public key while the receiver decrypts the symmetric key using his private key. In conclusion, symmetric encryption is an important encryption technique which has an important role in the field of information security. Even though there are some challenges, such as key management and security issues, symmetric encryption is still one of the effective means to protect data security after combining with other encryption techniques and improving the key management strategy.

2.2 Principles of the AES algorithm

AES is a symmetric-key encryption algorithm, one of the most used encryption algorithms today, and is widely used in the fields of data protection and secure communications. AES is a symmetric-key encryption standard defined by the National Institute of Standards and Technology in 2001 as a replacement for the DES algorithm. The AES algorithm employs the concept of packet ciphers, which divides the data into fixed length chunks and uses the same key for encryption and decryption operations. The AES algorithm adopts the concept of group cipher, dividing data into blocks of fixed length and using the same key for encryption and decryption operations. The operation process is that the plaintext data first undergoes an initial round of processing, including byte substitution, row shifting, column obfuscation, and round key addition, etc. After the initialization round, the data block is divided into multiple columns and then processed through multiple rounds of the round function. Each round of operation includes byte substitution, row shifting, column obfuscation and round key addition, etc. After multiple rounds of wheel function operation, the last round does not include column obfuscation, but directly performs operations such as byte substitution, row shifting and round key addition, etc., and the encrypted block of data is obtained after the last round of processing and is called the ciphertext.

The features of the AES algorithm are equally numerous. The AES algorithm has won the favor of a wide range of users for its excellent security, efficient performance, flexible key length and wide range of applications. First, the algorithm's high level of security is remarkable, as it can effectively resist all kinds of known cryptographic attack techniques, such as differential analysis and linear analysis, etc., to ensure the confidentiality of the data in the process of transmission and storage. Second, the AES algorithm demonstrates excellent execution efficiency in both hardware and software implementations, enabling both encryption and decryption operations to be performed quickly and in a variety of computing environments. In addition, the AES algorithm provides a variety of key length options, including 128-bit, 192-bit and 256-bit, which allows users to flexibly adjust the key length according to the actual security needs, thus improving the security of the algorithm. At the same time, the increase in key length also makes it significantly more difficult to crack, providing strong support for data security protection. It is worth mentioning that the design structure of AES algorithm is simple and clear, easy to understand and implement. This feature makes the AES algorithm ideal for a variety of secure communication and data protection scenarios, such as network communication, file transfer, database encryption, electronic payment, and virtual private networks. In addition, the AES algorithm is also the basis of many security protocols and standards, such as the SSL/TLS protocol and the IPsec protocol, which provide a solid foundation for building a secure and reliable communication environment. In conclusion, the AES algorithm occupies an important position in the field of data security by virtue of its multifaceted advantages and has become one of the widely used encryption technologies. Both individual users and enterprise organizations can safely adopt AES algorithm to protect their data security.

Advanced Encryption Standard (AES), as a symmetric encryption algorithm, plays a crucial role in securing sensitive data. Whether it is the encrypted transmission of emails, the secure storage of files, or the confidentiality of network communications, AES has demonstrated excellent performance and stability. The basic structure of AES mainly consists of key expansion, wheel function, inverse wheel function, etc.

The basic structure of the AES algorithm is shown in Fig. 2, the first key expansion, according to the key provided by the user, to generate a series of round key, used in the subsequent round of the function of the round key addition operation, the plaintext and the first round of the round key for the bitwise dissimilarity operation, and then after multiple rounds of encryption, each round to go through the byte substitution, row shifting, column obfuscation, the round key addition, and the final round of the final round of the step does not include column obfuscation, and ultimately the final encrypted The final encrypted state is the ciphertext. The decryption process also generates a series of round keys, the same as the encryption process, the initial round of the ciphertext and the last round of the round key for bitwise dissimilarity operation, the decryption process and the encryption process is the opposite, but the order is reversed, i.e., the round key is applied in the reverse order of the order of the final round of the final round of the encryption process with the final round of the same decryption process is the plain text of the state of the decryption.

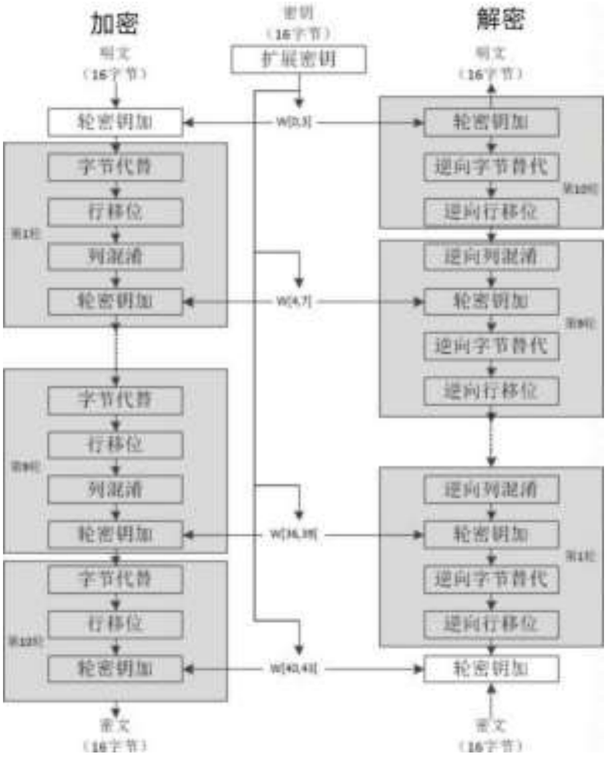

Fig. 2 Basic structure of AES

2.3 Number Theoretic Foundations of the AES Algorithm

AES is a symmetric-key encryption algorithm whose design is based on several key concepts in number theory, especially finite-field arithmetic. The encryption process of AES relies on the structure of Substitution-Permutation Network (SPN)[9], which performs encryption and decryption operations through multiple iterative rounds. The number theoretic foundation of the AES algorithm focuses on addition, multiplication, and polynomial operations over the finite field GF (2^8), which ensure the algorithm's security, complexity, and resistance to attacks.

The AES algorithm uses the finite field GF (2^8) (i.e., modulo 256 operations) for most of the operations in the encryption and decryption process. The finite field GF (2^8) is a field containing 256 elements, where each element is an 8-bit binary number. Addition and multiplication within this domain are performed by binary addition and modulo operations, while multiplication is usually implemented by multiplying polynomials and is performed using an irreducible polynomial for modulo reduction. Specifically, the addition operation employed in AES is the different-or (XOR) operation, which is an addition over the finite domain GF (2), while the multiplication operation is based on the computation of multiplicative inverses, which are used with affine transformations to enhance the nonlinearity of the algorithm. These finite domain operations not only enhance the complexity of the encryption but also make it more difficult for an attacker to crack the key.

Key expansion, a central aspect of the AES algorithm, plays a key role in expanding the short input key into a longer sequence of keys used to encrypt the round function. The algorithm is carefully constructed to ensure that a series of complex and secure round keys are generated from the input key, which are an integral part of each encryption round. According to the AES standard, the key expansion process determines the number of final round keys strictly based on the length of the input key. Specifically, if the input is a 128-bit key, 10 round keys are generated; if the key length is 192 bits, 12 round keys are generated; and for a 256-bit key, 14 round keys are generated. This design ensures that keys of different lengths are handled appropriately to meet diverse security requirements. Under the action of the key scheduling algorithm, the input keys undergo a series of complex mixing, transforming and rearranging operations to gradually generate the required round keys for each round. These operations are designed to enhance the randomness and complexity of the key, making the generated round key sequence highly secure and unpredictable. As the algorithm advances, a new round key is generated for each round and is added to the round key sequence in an orderly manner. This process is repeated until all round keys are generated. Eventually, we get a complete sequence of round keys, which will play a vital role in the subsequent encryption and decryption processes. The key expansion process ensures the security and efficiency of the AES algorithm and increases the difficulty of searching the key space by generating a complex and highly randomized sequence of round keys, thus improving the security of encryption.

The Round Function is a key part of the AES algorithm that is called during both encryption and decryption. The Round Function consists of four main steps. The byte substitution step replaces each byte with another byte using a fixed Substitution Box, where the substitution rules are fixed and irreversible, which increases the security of the algorithm. Row Shift In this step, each row of the AES state matrix is cyclically shifted left according to a specific rule. Specifically, the first row is kept unchanged to maintain the stability of the data; the second row is shifted to the left by one byte, which realizes the initial exchange of data within the matrix; the third row is shifted to the left by two bytes, which further increases the degree of data obfuscation; and the fourth row is shifted to the left by three bytes to ensure that the bytes of all the rows can be sufficiently obfuscated in the state matrix. Column Obfuscation This step performs an obfuscation operation on each column of the AES state matrix, which is achieved by multiplying it with a fixed matrix. The column obfuscation operation increases the nonlinearity of the algorithm and enhances the anti-analytical performance of AES. Wheel Keys Plus In each round of encryption of the AES algorithm, wheel keys play a crucial role. They are carefully generated by the key expansion algorithm and are designed to increase the randomness and complexity of the encryption process. When the wheel keys are subjected to a bitwise dissimilarity operation with the current state matrix, it not only realizes the obfuscation of the data in the state matrix but also ensures that each round of encryption operation possesses a unique transformation characteristic. The introduction of this step significantly enhances the obfuscation of the AES encryption algorithm such that the output of the same plaintext encrypted with the same key will be different in each round. This increased variability greatly improves the security of the algorithm and makes it more difficult to crack the AES algorithm.

The inverse wheel function phase of AES plays a crucial role in the decryption process, as it can restore the ciphertext to the original plaintext without any errors. In contrast to the wheel function phase of encryption, the inverse wheel function phase performs operations in the opposite order, ensuring that the encryption and decryption processes are complementary. One of the key steps in the inverse wheel function phase is inverse byte substitution, which corresponds to the byte substitution step in the encryption process. In inverse byte substitution, each byte of the ciphertext is carefully processed and each byte is accurately replaced with its corresponding value according to the mapping rules of the inverse S-box. This inverse substitution operation is like a key in the decryption process, which gradually restores the original appearance of the plaintext data and lays the foundation for the subsequent decryption steps. Immediately followed by the reverse row shift step, which is the reverse of the encryption process of the row shift operation. In the retrograde shift, each line is shifted in a reverse cycle according to specific rules, and as the number of lines increases, the amount of shift gradually decreases. This operation precisely adjusts the order of the bytes in each row, restoring them to their pre-encryption state. The execution of the inverse row shifting step not only makes the data in the ciphertext be effectively organized but also provides a guarantee for the final decryption result. In the inverse column obfuscation step, the inverse column obfuscation operation plays a crucial role by precisely performing an inverse linear transformation on each column. This operation effectively reverses the effect of column obfuscation in the encryption process by applying inverse matrix multiplication. By inverting the column obfuscation, the column data that was originally disrupted during the encryption process is recovered and re-presented as it was before the column obfuscation, paving the way for the subsequent decryption step. Immediately after that, the reverse wheel key addition step further advances the decryption process. In this step, the reverse-round key addition operation maintains a high degree of similarity with the round key addition operation in the encryption process. By performing a bitwise different-or operation between the current round's reverse-round key and the decryption state, the reverse-round key addition operation gradually restores the original state of the plaintext data. This operation not only restores the obfuscation level of the data but also ensures the accuracy of the final decryption result. It is important to note that the order of operations in the reverse wheel function stage is completely opposite to that of the wheel function stage in the encryption process. This reverse operation design concept allows the AES algorithm to gradually restore the original plaintext data during the decryption process. Through the well-designed steps of inverse row shift, inverse byte substitution, inverse column obfuscation and inverse wheel key addition, the inverse wheel function stage ensures the integrity and security of the data, providing a solid guarantee for the safe transmission and storage of data.

3. TEXT ENCRYPTION AND VIDEO ENCRYPTION WITH AES ALGORITHM USING OPENSSL

3.1 Experimental platforms and tools

OpenSSL is an open-source cryptography toolkit that provides implementations of various cryptographic functions and protocols, including the SSL/TLS protocol, encryption algorithms, digital certificate management, etc. OpenSSL implements the SSL and TLS protocols, which are used to provide security and privacy protection for network communications. The SSL/TLS protocol provides encryption, authentication, and integrity protection, and is commonly used to protect web applications, email transmissions, etc. OpenSSL supports a variety of encryption algorithms, including symmetric encryption algorithms, asymmetric encryption algorithms, and hash functions. These algorithms can be used to encrypt data, generate digital signatures, calculate message digests, etc. OpenSSL supports the generation, issuance, verification and management of digital certificates. Digital certificates play an important role in network security and are used for authentication and secure communication. OpenSSL provides a range of command line tools for performing various cryptographic operations. OpenSSL includes interfaces and tools for generating secure random numbers, which are essential for cryptographic operations and key generation. Secure random numbers are used in cryptographic operations for the generation of initialization vectors, random number seeds, keys, etc. OpenSSL runs on a wide variety of operating systems, including Linux, Unix, Windows, etc., enabling developers to use the same cryptographic tools and functionality on different platforms. Overall, OpenSSL is a powerful, flexible and widely used cryptographic toolkit that provides developers with a rich set of cryptographic functions and protocol implementations that help build secure network communications and applications. The experimental environment is configured as follows: The operating system is Windows 10 / Ubuntu 20.04, the OpenSSL version is OpenSSL 1.1.1, and the experimental hardware is an Intel i7 processor with 16GB RAM.

3.2 Experimental data set

In order to verify the encryption effect of the AES algorithm on different data types, two types of files are selected for encryption in this experiment, for text files, a customized text file in .txt format is selected as the experimental data, and for video files, a 5-minute-long video file in .mp4 format is selected, with a resolution of 1920x1080. The purpose of selecting these data is to test the encryption and decryption performance of the AES algorithm when dealing with different sizes and types of files. The purpose of choosing these data is to test the performance of AES algorithm in encrypting and decrypting files of different sizes and types. These data were chosen to test the performance of the AES algorithm in encrypting and decrypting files of different sizes and types. These data have been chosen to test the encryption and decryption performance of the AES algorithm on files of different sizes and types. The performance of the conventional DES algorithm is also compared with the performance of the conventional DES algorithm in processing these files.

3.3 Encryption and decryption process

The core operation of the AES algorithm is based on symmetric key encryption, i.e., the same key is used for encryption and decryption. In this experiment, the AES algorithm with 128-bit key length is used to perform encryption and decryption operations on text and video data. In the text file encryption stage, the specific operation steps first select a text file (input.txt). Then select the key and choose a key of 128-bit (16-byte) length (e.g., "yourpassword"). This key will be used for round key generation during encryption and decryption. The encryption operation is performed using the AES-128 algorithm. Enter the OpenSSL enc -aes-128-cbc -a -in your.txt -out encryted.txt -K yourpassword -iv initialization vector command to encrypt in cbc mode and automatically generate the encrypted.txt file to be used for storing the encrypted file, -K is the passphrase, which consists of 32 hexadecimal digits, and -iv is the initialization vector, can be the same as the password, or you can set it yourself, the length should not be too short, -a is the base64 encoding. available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

To decrypt an encrypted file, enter the command OpenSSL enc -aes-128-cbc -a -in encrypted.txt -out decode.txt -K yourpassword -iv initialize vectors -d to decrypt the file, automatically generating a decode file to store the decrypted file, -K and -iv must be the same as in encrypted, and -d for the decryption command. The decrypted file is the same as the original file

Video and text encryption have the same instruction structure when using the AES algorithm, because AES is a symmetric encryption algorithm, regardless of whether the encrypted data is text, video, or any other type of data, as long as the same encryption mode is used (e.g., AES-128-CBC), the structure of the instructions to encrypt and decrypt are the same, but they have different data formats and processing performance

3.4 Performance Evaluation

After the encryption and decryption operations were completed, a series of performance evaluations were performed, including the following: measuring the time required from the start of encryption to the completion of encryption. Measure the time needed from the start of decryption to the completion of decryption. Record the change in file size after encryption to evaluate the impact of the encryption algorithm on the file size. Ensure file consistency between encrypted and decrypted files by calculating the hash value of the encrypted file.

3.5 Results

We recorded the time of AES and DES in encrypting and decrypting different files (text files and video files) and the results are shown in Table 1 below.

Table 1. Performance Comparison of AES Algorithm and DES Algorithm

File type	AES encryption time	AES decryption time	DES encryption time	DES decryption time
text file	0.02s	0.01s	0.05s	0.04s
Video files	12.5s	11.8s	20.3s	19.2s

The AES algorithm outperforms DES in handling both data types, along with a slight change in file size before and after encryption. As both AES and DES algorithms padded the file, the file size increased slightly. Specifically, the size of the file after AES encryption increases slightly (e.g., by 1-2 KB) compared to the original file, while the change in file size after encryption is basically the same for DES. We verify the integrity of the encrypted data by hash value comparison. All the AES and DES encrypted files have the same hash value as the original file after decryption, proving that both perform well in terms of confidentiality and data integrity.

4. CONCLUSIONS

In this study, we explore the application of AES algorithm in text and video encryption, focusing on analyzing the experimental process and effect of encryption and decryption through OpenSSL tool. The experimental results show that the AES algorithm, as an efficient and secure symmetric encryption algorithm, can effectively protect the privacy of sensitive data, especially in the encryption process of large-scale data, such as text and video, showing excellent performance. Through the encryption experiments on text and video files, we find that AES has high security and stability in the encryption and decryption process. Especially for the AES-128 mode, the key length is sufficient to meet the daily encryption requirements, and at the same time, the parallel processing capability of the AES algorithm makes it possible to realize efficient encryption and decryption processes under the condition of hardware acceleration. The encrypted ciphertext, despite the increase in size, the AES algorithm successfully protects the original data content and prevents the risk of data leakage through reasonable key management and encryption mode. Compared with traditional encryption algorithms (e.g., DES, 3DES), the AES algorithm shows obvious advantages in several aspects. First, AES supports longer key lengths (128-bit, 192-bit, 256-bit), which provides stronger resistance to cracking compared to DES's 56-bit key. Second, AES is more computationally efficient during encryption and decryption, and is able to process large-scale data more quickly, especially in the encryption of videos and other large file types, where AES performs more efficiently compared to traditional encryption algorithms. However, despite its outstanding performance in terms of encryption efficiency and security, we still need to pay attention to its actual performance in different environments, such as embedded devices and low-power devices. As emerging technologies such as quantum computing continue to evolve, traditional symmetric encryption algorithms may face new challenges, and future research may need to explore how AES algorithms can enhance their resistance to quantum computing attacks and how their performance can be further improved by optimizing hardware and algorithms. Overall, the AES algorithm, as a core tool for modern information security, has been widely used in multiple fields, such as text encryption and video encryption, and has demonstrated strong technical advantages. In the future, we will continue to explore the performance of AES algorithm in more complex application scenarios in order to further improve its security and efficiency.

5. ACKNOWLEDGEMENT

Many people have given me great help and support during the completion of my thesis. I would like to express my heartfelt thanks to them with immense gratitude. Finally, I would like to thank everyone who has worked hard for this thesis and the authors of all the references, whose research results have provided important theoretical support for my study. It is by standing on the shoulders of the giants before me that I have been able to make gains in this field. Although I have tried my best to do my best in the process of writing and researching the thesis, there are still inevitably shortcomings, and I earnestly ask all experts and teachers to criticize and correct me. Once again, I would like to thank all those who have supported, cared and helped me!

6. REFERENCES

[1] Zhang, H., Li, W., & Liu, Y. (2004). *A Study on the Deve lopment and Application of AES Algorithm in Cryptograp hy*. Journal of Cryptographic Research, 12(2), 45-56.

[2] Wang, J., & Chen, X. (2009). *Optimizing AES Algorithm: Performance Enhancements and Security Analysis*. Inter national Journal of Cryptography and Security, 14(4), 11 3-130.

[3] Liu, Q., Zhang, Z., & Yang, T. (2014). *Research and App lication of AES Algorithm: New Encryption Schemes and Optimization Strategies*. Journal of Applied Cryptograph y, 20(3), 201-214.

[4] Zhao, F., & Xu, P. (2018). *Exploring the Applications of AES Algorithm in IoT and Cloud Computing Security*. Int ernational Journal of Network Security, 16(1), 87-101.

[5] Smith, L., & Roberts, D. (2007). *Security Analysis and C ountermeasures of AES Algorithm: Attack Models and De fense Strategies*. Cryptography Review, 18(4), 151-165.

[6] Liu, X., & Zhang, Y. (2013). *Hardware Implementation and Performance Optimization of AES Algorithm*. Journa l of Hardware Security, 11(2), 120-134.

[7] Johnson, M., & Smith, A. (2018). *The Role of AES in Em erging Technologies: Quantum Computing and AI Securi ty*. Journal of Cryptographic Systems, 22(3), 210-225.

[8] Patel, R., & Kumar, S. (2022). *Application of AES Algori thm in Blockchain and AI Security*. Journal of Modern Cr yptography, 29(1), 135-146.

[9] Daemen, J., & Rijmen, V. (2002). AES proposal: Rijndae l. In Advanced Encryption Standard (AES) - Design and Analysis (pp. 195-230). Springer.