Proceedings of the 3rd International Conference on Mechatronics and Smart Systems
DOI: 10.54254/2755-2721/135/2025.21089

Dynamic Encryption Overview

Hangjie Li[1,a,*]

[1]*Department of Software Engineering, School of Software, Sichuan University, Sichuan, China*
a. 2022141470279@stu.scu.edu.cn
**corresponding author*

Abstract: With the rapid development of information technology and the growing demand for network security, traditional static encryption methods have gradually failed to meet the requirements of modern environments for flexibility and security. Dynamic encryption technology has emerged to dynamically adjust the encryption policy according to the network status, data type and actual needs, thus providing a more efficient and flexible protection program. This paper summarizes the basic principles of dynamic encryption technology and its encryption and decryption process, focusing on the analysis of the acquisition of dynamic key. Compared to static encryption, dynamic encryption is able to cope with different threat scenarios in changing environments, enhancing data security and availability. The article also reviews the current research progress and major challenges facing dynamic cryptography, and explores future directions, especially the potential for applications in emerging areas such as quantum cryptography, artificial intelligence, and privacy computing. Dynamic encryption technology shows great application prospects in dealing with future complex security threats and safeguarding data privacy.

Keywords: Dynamic Encryption, dynamic key, dynamic algorithm, network security

1. Introduction

With the continuous progress of network attacks, the demand for data security is increasing. Traditional static encryption technology has limitations in fixed encryption methods and key management, making it difficult to adapt to the rapidly changing security threats in the modern network environment[1,2]. In this case, dynamic encryption has emerged as an innovative encryption technology to meet the demand for flexible data security. Dynamic encryption is an encryption technology that can adjust encryption strategies and algorithms in real time according to the actual environment and needs, and ensures the security of data in the process of transmission and storage through flexible encryption keys, algorithm selection and dynamic changes in encryption mode.

The core of dynamic encryption technology lies in its flexibility and adaptability: it can adaptively adjust the key as well as the encryption method in scenarios such as changes in the security environment and different degrees of data importance. For example, in the cloud computing environment, different data of users may have different security requirements, and at the same time, it is also necessary to balance the data security and equipment performance, at this time, according to the importance of the private information contained in the data packet, the encryption method of choosing the data packet for the transmission of the split-channel encryption is very suitable[3]. This adaptability greatly enhances the application potential of dynamic encryption in the fields of network

DOI: 10.54254/2755-2721/135/2025.21089

security, data privacy protection, etc., especially in cloud computing, blockchain, Internet of Things and other modern fields that require high data security have a wide range of application value[4-8].

In cloud computing, dynamic encryption is mainly used to solve the security problems in data storage and transmission. Shared resources and cross-domain access in cloud environments make it difficult for traditional encryption methods to adapt to different security requirements. Dynamic encryption can flexibly adjust encryption algorithms and keys according to data sensitivity, network conditions, and user behavior to improve data security while strengthening access control. In the blockchain field, dynamic encryption helps safeguard smart contract and transaction privacy. Static encryption schemes for blockchain are usually used to ensure data immutability, but dynamic encryption can respond to different security needs and enhance transaction privacy and tamper resistance through adaptive key management and protocol adjustment. In IoT, dynamic encryption is used to secure inter-device communication. Internet of Things (IoT) devices usually have limited resources, and dynamic encryption dynamically adjusts the encryption strength according to network conditions and data sensitivity to reduce computational overhead and effectively prevent data leakage and attacks.

The purpose of this review is to systematically analyze the core methods of dynamic encryption and to discuss its future development directions and challenges. Based on literature research and application case analysis, this paper divides the core methods of dynamic encryption into dynamic keys as well as dynamic algorithms. Under this structure, this review will deeply explore the implementation of the techniques in each method and analyze the challenges in the current applications.

2. Dynamic Encryption Fundamentals

The idea of dynamic encryption originates from the complexity and dynamism of security requirements in modern network environments. With the rapid development of information technology, the traditional static encryption method exposes certain limitations and is difficult to cope with the increasing complexity and diversity of network attacks.

The idea of dynamic encryption mainly comes from the limitations of traditional encryption techniques. Traditional encryption techniques mostly use static keys and fixed algorithms, and the whole encryption system will face a security threat once an attacker obtains the key or cracks the algorithm if the key has not been replaced [9]. In addition, the static encryption model is difficult to meet the needs of dynamic scenarios in modern networks, such as simultaneous access by multiple users, multi-layer data sharing, and real-time communication scenarios. These limitations have given rise to the need for encryption to evolve dynamically. Dynamic encryption is also influenced by the idea of moving target defense, which is to make it more difficult for attackers to predict and crack by constantly changing the configuration of the network, system, or data [10]. Dynamic encryption improves the security of data protection by frequently updating the encryption, making it difficult for attackers to get hold of the key or decryption algorithm.

Dynamic encryption is a highly adaptable and flexible encryption technique whose properties can be summarized in the following aspects:

(1) Real-time and dynamic: Dynamic encryption can adjust encryption strategies in real time according to environmental changes. For example, when the network bandwidth is limited, dynamic encryption can choose a lower complexity encryption algorithm to ensure the efficiency of data transmission; and when sensitive data is transmitted, it can be switched to a high-strength encryption method to improve security. This ability to adjust dynamically gives Dynamic Encryption a unique advantage in an environment of uncertainty and variability.

(2) Adaptability and self-adjustment: Dynamic encryption can dynamically select suitable encryption algorithms, key lengths, or encryption modes according to different security requirements

DOI: 10.54254/2755-2721/135/2025.21089

and network environments. For example, in IoT scenarios, different devices have different data importance and computing power, and dynamic encryption can adapt to these differences and balance security and performance through self-adjustment.

(3) Attack resistance and enhanced security: Dynamic encryption makes it difficult for attackers to crack through fixed-mode analysis by frequently changing keys, updating algorithms, or randomizing encryption strategies. For example, a dynamic key management system regularly updates keys or dynamically distributes keys according to the amount of data, reducing the risk of key interception or cracking. In addition, dynamic encryption can be combined with pseudo-random number generators and multiple encryption techniques to further enhance attack resistance.

3. Framework for dynamic encryption

The encoding and decoding process of dynamic encryption is similar to traditional encryption in that it requires specific encryption algorithms and keys to encrypt and decrypt data. The difference is that dynamic encryption dynamically adjusts the algorithm, key or encryption strategy according to specific scenarios or security requirements, thus increasing the flexibility and security of the system.

3.1. Encoding process

(1) Data preprocessing: the data to be encrypted is preprocessed according to an agreed format

(2) Dynamic key generation: one of the core of dynamic encryption is dynamic key generation.

Time-based: the key can be dynamically generated based on the current timestamp and updated every certain period of time.

Based on function: the key is generated in real time based on a non-linear function.

Based on random number generator: use high quality pseudo-random number generator or true random number generator to generate dynamic key.

(3) Dynamic Algorithm Selection: Dynamically select encryption algorithms based on preset policies or environmental changes.

(4) Encrypting Data Blocks: Use dynamically generated keys to encrypt each data block.

(5) Encryption Metadata Generation: Attach metadata required for encryption (e.g., dynamic key, encryption algorithm identifier, timestamp, or checksum value) to the header or tail of the encrypted data for use in decryption. These metadata usually also need to be encrypted or signed to prevent tampering.

(6) Output encrypted data: outputs the final encrypted data ready for transmission or storage.

3.2. Decoding process

(1) Data reception and parsing: After receiving the encrypted data, the encrypted metadata is first extracted. If the metadata is encrypted, the metadata is decrypted using the agreed decryption method.

(2) Dynamic key reduction: According to the metadata or dynamic key generation rules, the dynamic key used in encryption is regenerated.

(3) Dynamic Algorithm Matching: Select the correct decryption algorithm according to the algorithm identification in the metadata.

(4) Data block decryption: decrypt encrypted data block by block using the restored dynamic key.

(5) Data Integrity Verification: Perform integrity verification on the decrypted data to ensure that the data has not been tampered with during transmission or storage. Verification methods include checksum, hash value comparison or digital signature verification.

(6) Data reorganization and output: reassembles the decrypted data blocks into original data. If the data has been pre-processed before encryption, such as compression or encoding, the reverse process is performed, and the data is eventually restored to usable plaintext.

DOI: 10.54254/2755-2721/135/2025.21089

4. Types and Applications of Dynamic Encryption

The types of dynamic encryption mainly include dynamic keys and dynamic algorithms. Dynamic key refers to generating, updating and replacing the key periodically or on-demand during the encryption process, so as to enhance the security of data and avoid the risk of key leakage. Dynamic algorithms, on the other hand, refer to the flexible adjustment of encryption algorithms used according to specific application scenarios or environmental conditions to optimize encryption performance and maintain data security. These two techniques enhance the efficiency and security of encryption systems in variable environments by adaptively adjusting encryption strategies. This paper focuses on the application of dynamic keys.

Wenliang Xie et al. have mentioned the concept of dynamic key, where the server updates the key stored in the database after each login [11]. During password transmission, the client uses a one-time password which is invalidated once the login is completed, preventing the attacker from logging in again using the invalidated password. For the generation of one-time key, different researchers have proposed various optimization algorithms, which mainly contain time-based, function-based and dynamic key generation based on random number generator.

4.1. Time-based dynamic key

Time-based dynamic key generation is a widely used key generation method in information security and cryptography. The method aids in the dynamic generation of authentication keys by utilizing timestamps or time information, and the keys are usually updated at certain intervals. Highly dynamic and flexible, time-based key generation is one of the most important means to ensure the security and privacy protection of data during transmission, especially playing a crucial role in scenarios such as authentication, digital signature and data encryption.

One of the most widely known applications of time-based dynamic key generation is One-Time Password (OTP) algorithms, especially Time-One-Time-Password (TOTP). TOTP is one of the algorithms that use timestamps and shared keys to generate one-time passwords. TOTP works as follows:

The TOTP system first assigns a shared randomly generated key to the user, which is stored on the user's device and in the authentication server. The system obtains the current UTC time, which is usually split in fixed time segments. Using the timestamp and the shared key, the system computes a hash value of a specific length by means of a hashing algorithm, and part or the entire value of the hash is output as a one-time password. The server generates the corresponding TOTP based on the key and the current timestamp and compares it with the password entered by the user. If it matches, the authentication passes. At fixed intervals, new one-time passwords are generated to ensure that each password is valid for a short period of time and to avoid using the same password for a long period of time [12].

Xie Wenliang et al. mentioned a method of using timestamps for verifying keys, the main idea is to take the timestamps and keys for calculations in both the registration and login phases when accessing resources and send the results to another party, when the other party receives the post data, they take out part of it for calculations and compare it with the accepted information, and if it meets the requirements, then it will be able to prove the authenticity of the identity as well as the integrity of the data [11]. Based on this, they also improve the possible attacks in the original scheme.

TOTP is widely used in several domains, in two-factor authentication, it enhances the security of authentication by providing a combination of dynamic and traditional passwords, common applications are Google Authenticator, Microsoft Authenticator, Authy, etc., and in the financial domain as well [13].

Time-based dynamic key generation ensures that each key is unique and has a very short validity

DOI: 10.54254/2755-2721/135/2025.21089

period, greatly reducing the risk of key leakage or reuse. Each key has a strict time limit, and an attacker cannot reuse the key even if it is intercepted. Moreover, compared with other key management mechanisms, the time-based dynamic key generation mechanism is relatively simple and easy to implement, balancing security and low cost [14].

4.2. Function-based dynamic key

Among encryption systems, One-Time Pad (OTP) is theoretically the most secure encryption method, which requires that a unique and randomized key be used for each communication and that the key be used only once. This type of encryption theoretically provides complete confidentiality because each key is unique and equal in length to the data, and it is not possible to restore the encrypted data through brute force decryption. However, one-at-a-time encryption encounters a major challenge in practice: how to efficiently and dynamically generate and update the key required for each communication.

Usually, encryption algorithms already come with a key update mechanism for generating new keys according to specific rules, but most of the existing key update functions are linear, i.e., the process of key update follows a predictable pattern. Although linear updating can effectively improve the efficiency in some cases, it lacks sufficient randomness and complexity to meet the requirement of "one secret at a time". In order to realize the real "one time encryption", it is necessary to avoid the predictability of the key pattern through a more complex, non-linear key update function mechanism, so as to enhance the security of the encryption system.

To solve this problem, researchers have proposed a nonlinear key updating mechanism, which mainly consists of two methods: recursive method and counting method. These two methods ensure the uniqueness and unpredictability of the key in each communication through different mechanisms and algorithms, thus improving the attack resistance of the encryption system. Next, the article will discuss the principles of these two nonlinear key update methods and their applications in detail [15].

4.2.1. Recursive method

As shown in the following equation: $Ki = EK(Ki - 1)$; in this equation, Ki is the current key used for encryption, $Ki - 1$ is the previous key used for encryption, K and is the basic key, $EK(Ki - 1)$ indicating that the encryption algorithm uses the key K to encrypt $Ki - 1$. The recursive method is suitable for key updates of different packets in the same communication using packet cipher algorithms; key updates for different frames are not easily synchronized.

Recursive method has a wide range of applications in the field of encryption as an algorithm that generates the next state from the previous state. Recursion ensures uniqueness and unpredictability of each encryption operation by generating different keys, key streams, or encryption blocks. The application of recursive methods plays a central role in various encryption modes (e.g., stream encryption mode, block encryption mode, and in some advanced encryption schemes), especially in enhancing key management and improving encryption security.

The recursive method was earlier applied in the output-Feedback mode (OFB), one of the feedback modes, which generates a keystream by recursion, and then encrypts it block-by-block with the plaintext on a dissimilarity basis. The key of the OFB mode is to form a recursive process by using the output of the previous round of encryption as the input for the next round of encryption. In this way, OFB mode ensures that the keystream is not duplicated during the encryption process, thus improving security.

With the continuous development of the network society, the information security problem becomes more and more important, and the cyber attacks become more and more intense. As a result, the packet cipher algorithm encryption mode Output Feedback with Nonlinear Functions (OFBNLF)

DOI: 10.54254/2755-2721/135/2025.21089

has made important improvements on the basis of OFB, which mainly enhances the complexity and unpredictability of the key flow by int roducing nonlinear functions, and it is a typical mode adopting recursive method.

The traditional OFB mode feeds the encrypted output of the previous round directly as the input of the next round, which is easy to be predicted by attackers through analysis. On the contrary, OFBNLF mode makes each generated keystream more complex by adding nonlinear functions (e.g., hash functions, S-boxes, etc.) in the feedback process, which greatly enhances the security and resistance to analysis of the keystream. The OFBNLF model is described in detail in the book Applied Cryptography [16].

OFBNLF mode combines OFB mode and ECB mode, the upper layer adopts OFB mode, which uses the master key to generate a series of required subkeys; the lower layer is similar to ECB mode, which utilizes the subkeys to encrypt the packet cipher. The OFBNLF mode is divided into two processes, encryption and decryption. The process is mainly as follows: first define an initial vector and key and start generating the feedback value by encryption function. At each time a new feedback is generated, the encryption result is processed through a nonlinear function. Where the nonlinearity can be a complex permutation, S-box, hash operation or other mathematical nonlinear mapping. And then the result obtained is retained and isomorphized with the plaintext to obtain the ciphertext. The next feedback uses the retained result from the previous one and continues to generate a new keystream by encryption and nonlinear processing. This process is continued until all plaintext blocks are processed.

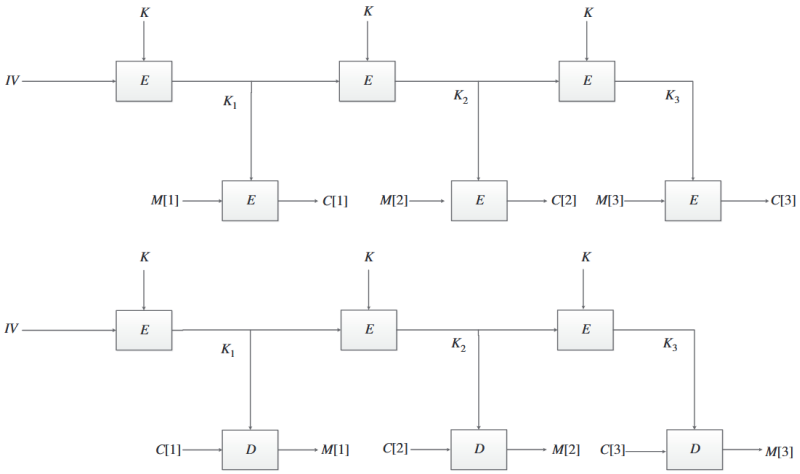

Figure 1: Output Feedback Operating Mode with Nonlinear Function (OFBNLF) [14]

Figure 1 above depicts the encryption process in OFBNLF mode.

Zhelei Sun et al. conducted an analysis of OFBNLF encryption mode of operation, and based on giving the online encryption description of OFBNLF mode, they used the technique of making games to prove the security of OFBNLF mode under the indistinguishable model of packet-by-packet selective plaintext attack for the first time [17].

In addition to this, the recursive method has been used in a variety of other algorithms such as the RC4 algorithm and the ChaCha20 algorithm [18,19]. These algorithms have also been progressively improved in later studies to be more secure.

4.2.2. Counting method

As shown in the following equation: $Ki = \mathrm{re}key(K; IV)$; The counting method, also known as counter mode (CTR), generates a stream cipher for the key stream by encrypting the counters that are

DOI: 10.54254/2755-2721/135/2025.21089

accumulated one by one. The basic key K is usually already set, and the receiver only needs to synchronize according to the count value IV transmitted from the sender. In CTR mode, each packet corresponds to a counter that is accumulated one by one, and the key stream is generated by encrypting the counters. In other words, the final ciphertext packet is obtained by XORing the sequence of bits obtained from the counter, with the plaintext packet, as shown in Figure 2. The basic algorithm is as follows:

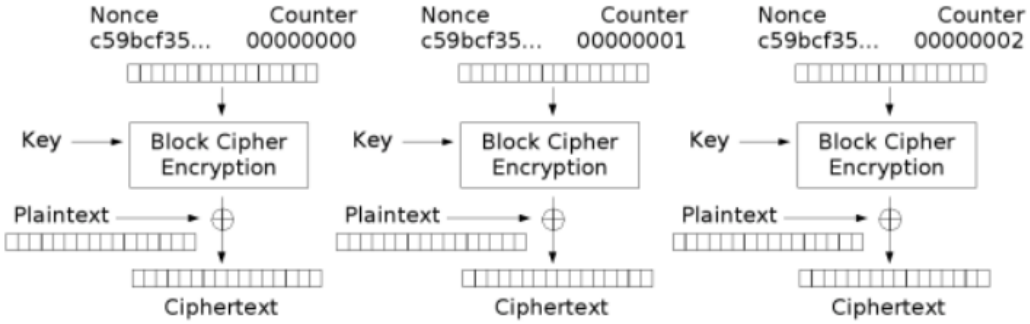

Figure 2: Counter mode [15]

An initial counter value VI is first selected as the initial vector. This counter is usually used in conjunction with a cryptographic key and can be an incremental value, a timestamp, or some other value that guarantees uniqueness. The structure of the counter usually consists of two parts: a higher-bit part that is used to guarantee the uniqueness of the counter, and a lower-bit part that is incremented. Each time the keystream is computed, the counter value is first passed into a packet encryption algorithm (e.g., AES) along with the encryption key

$$K_{\text{ey}}Streamn = EK(Countern) \tag{1}$$

Where EK denotes the encryption operation, K is the encryption key, and Countern is the current counter value. After each generated keystream block, the counter is incremented, usually by the bottom part of the bits, to ensure the uniqueness and continuity of each counter value in the encryption process. The generated keystream is differentiated (XOR) from the plaintext to obtain the encrypted ciphertext:

$$Cn = Pn \oplus KeyStreamn \tag{2}$$

There are several features of CTR mode, the first one is parallelism, because the generation of each keystream block depends only on the counter, and the counter value is deterministic, so CTR mode allows parallel encryption and decryption. This means that multiple blocks of data can be encrypted at the same time without having to wait for the encryption result of the previous block, which is particularly suitable for efficient hardware and large-scale data encryption. Second, unlike other modes (e.g., CBC, OFB), CTR mode does not need to maintain state information. Only counters and keys are needed each time a keystream is generated, and the counter increment is independent of the data block. Therefore, the decryption process does not need to rely on the encryption result of the previous block.

Currently, most key update mechanisms use counting methods, for example, the key update proposal by Albert Young of 3Com, the Temporary Key Integrity Protocol (TKIP) scheme proposed by Jesse Walker of Intel's Microsoft and Tim Moore et al [15].

DOI: 10.54254/2755-2721/135/2025.21089

4.3. Dynamic key based on random number generator

Keys generated based on Random Number Generator (RNG) are very common in cryptosystems, especially in symmetric encryption and stream cipher applications. Random number generators are used to generate encryption keys, initialization vectors, key streams, or random data required in their cryptographic algorithms.

Random number generators can be divided into two categories: true random number generators (TRNG), which derive randomness from physical processes (e.g., thermal noise, radioactive decay, etc.), and pseudo-random number generators (PRNG). This type of random number generator does not depend on algorithms or the internal state of the computer, so the random numbers generated are unpredictable; the latter generates a series of pseudo-random numbers through a mathematical algorithm based on an initial seed (SEED) value. Although these random numbers appear to be random, they are generated by a deterministic algorithm and are therefore predictable. Both approaches generate dynamic keys in roughly the same way, and the following pseudo-random number generator is used as an example to describe the generation process:

First, a random source (e.g., system time, hardware RNG, user input, etc.) is selected as the seed value. Initialize the PRNG using the generated seed. A common initialization algorithm may be to hash the PRNG using the seed value or initialize the state of the PRNG by other means. Use the PRNG to generate a random number of the desired length as the encryption key. For example, if AES-128 encryption is used, a 128-bit key needs to be generated. The PRNG outputs a 128-bit random number that serves as the key for the AES algorithm. Finally, the key is used to perform encryption and decryption operations.

Xu Shuang et al. designed a two-way diffusion mechanism and pseudo-random number synchronous generator, and proposed an image encryption algorithm based on the two-way diffusion mechanism and pseudo-random number synchronous generator, which uses an improved pseudo-random synchronous generator to process the key and optimize the encryption process of the image, which not only dramatically improves the computational efficiency, but also increases the security of the system [20].

5. Challenges and prospects

Although dynamic encryption technology has made remarkable progress in the field of information security, it still faces a series of challenges in its practical application. First, the complexity of key management and distribution is high, and how to ensure the security and timely update of keys without increasing the computation and communication overhead is an urgent problem. Secondly, dynamic encryption systems need to find a balance between security and performance, and complex encryption algorithms may bring high computational overhead, affecting the real-time performance and efficiency of the system. Finally, dynamic encryption technology also faces cross-platform compatibility issues. With the popularization of cloud computing, IoT, blockchain and other technologies, cross-platform transmission of data is more and more frequent, and how to ensure seamless encryption and decryption between different platforms remains a challenge.

At the same time, algorithms that can dynamically adjust their behaviors and strategies according to changes in input data or environmental conditions are also a way to enhance the adaptability of security, such as in the application of SSL protocols for electronic transactions, when the customer and the merchant carry out in the first time communication, the two sides through the handshake agreement in the data encryption algorithms, key exchange algorithms, hash algorithms, and the version number of the consultation, and ultimately reach an agreement. However, its popularity in practical applications is relatively low, and the flexibility of the algorithms means that more computational resources are required, which also increases the complexity of key management, and

DOI: 10.54254/2755-2721/135/2025.21089

in environments where high security is required, key generation, distribution, and storage become more complex. The lack of standardized protocols and compatibility issues in different systems and platforms may hinder its wide application. Especially when communicating across platforms, ensuring that different platforms support the same dynamic encryption algorithms and key exchange protocols may encounter technical and implementation challenges. The reason for the lag in the development of this technology will also be a major issue to be addressed in the future.

Looking ahead, with the rapid development of artificial intelligence and quantum computing technology, dynamic encryption technology is expected to make breakthroughs in security and intelligence.AI can help dynamic encryption systems more intelligently cope with complex cyberattacks, while quantum encryption technology provides stronger security for dynamic encryption. In addition, with the development of privacy protection and cross-platform support technology, dynamic encryption will be widely deployed in more application scenarios and play an increasingly important role in the future network security system.

6. Conclusion

This paper discusses in detail the principle, research status and application of dynamic encryption technology, and analyzes its advantages and challenges in information security. Compared with traditional static encryption, dynamic encryption can effectively improve security and adaptability by adjusting encryption strategies and keys in real time. However, issues such as key management, algorithm performance, and cross-platform compatibility remain key challenges in its popularization and application. In the future, with the development of artificial intelligence, quantum computing and privacy protection technology, dynamic encryption will usher in new development opportunities and is expected to provide more reliable security in a more complex and volatile network environment. Therefore, dynamic encryption technology will play a crucial role in the future network security system and promote the development of information security protection technology to a higher level.

References

[1] *Wan, L. Y. (2007). The Frigate of Information Security-Information Hiding and Digital Watermarking Technology. Proceedings of the 2007 National AECC Symposium. Taiyuan, 30-32.*

[2] *Yegireddi, R., Kumar, R. K. (2016). A survey on conventional encryption algorithms of Cryptography. 2016 International Conference on ICT in Business Industry & Government (ICTBIG). 1-4.*

[3] *Lv, J. Y., Zhu, Z. R., Yao, Z. Q. (2020). A dynamic encryption strategy for dual-channel data in cloud computing environment. Computer Applications, 40(8): 2268-2273.*

[4] *Fu, H. Y. (2010). Design and realization of secure data storage system based on cloud computing. Coal Technology, 29(12): 169-171.*

[5] *Hu, X. Y. (2017). Research and simulation of dynamic access control method for cloud computing storage data. Computer Simulation, 34(3): 365-368.*

[6] *Li, L., Tan, Z. W., Zhu, J. W., et al. (2024). Dynamic searchable encryption scheme with fuzzy multi-keyword support. Information Security Research, 10(11): 1064-1073.*

[7] *Xu, W. S., Zhang, J. B., Yuan, Y. L. (2023). A dynamically verifiable symmetric searchable encryption scheme based on blockchain. Journal of Software, 34(11): 5392-5407.*

[8] *Sun, X. L., Wang, D. H., Li, S. S. (2024). A dynamic ciphertext sorting retrieval scheme based on blockchain. Computer Applications, 44(8): 2500-2505.*

[9] *Komori, T., Saito, T. (2004). A secure wireless LAN system retaining privacy. 18th International Conference on Advanced Information Networking and Applications, AINA 2004.: Vol. 2. 2004: 370-375 Vol.2.*

[10] *Zhou, Y. Y., Cheng, G., Guo, C. S., et al. (2018). Research review on attack surface dynamic transfer techniques for mobile target defense. Journal of Software, 29(9): 2799-2820.*

[11] *Xie, W. L., Liao, W. M., Yang, C. I. (2009). One-time key authentication scheme and its improvement. Information Network Security, 22-24.*

[12] *M'Raihi, D., Machani, S., Pei, M., et al. (2011). TOTP: Time-Based One-Time Password Algorithm: RFC6238. RFC Editor, RFC6238.*

DOI: 10.54254/2755-2721/135/2025.21089

[13] Yin, X., He, J., Guo, Y., et al. (2020). An Efficient Two-Factor Authentication Scheme Based on the Merkle Tree. SENSORS, 20(20): 5735.

[14] Bao, Z. T., Ke, J. M., Yang. Z., et al. (2021). A time-based one-time password scheme for PLC. Computer Engineering, 47(8): 149-156.

[15] Huang, Y. Z., Hu, A. Q., Song, Y. B. (2003). Research and implementation of key update algorithm in network security. Computer Engineering and Applications, 39(35): 27-29.

[16] Schneier, B. (2015). Applied Cryptography_Protocols, Algorithms and C Source Programs.

[17] Sun, Z. L., Wang, P. (2016). Analysis of OFBNLF encryption working mode. SCIENCE CHINA: INFORMATION SCIENCE, 14.

[18] Cai, W., Chen, H., Wang, Z., et al. (2022). Implementation and optimization of ChaCha20 stream cipher on sunway taihuLight supercomputer. JOURNAL OF SUPERCOMPUTING, 78(3): 4199-4216.

[19] Hu, L., Chi, L., Yuan, W., et al. (2012). Cryptanalysis and improvement of RC4 algorithm. Journal of Jilin University (Science Edition), 50(3): 511-516.

[20] Xu, S., Wang, W., Su, Y. (2014). Research on fast image encryption algorithm based on two-way diffusion mechanism fused with pseudo-random number synchronization generator. Science, Technology and Engineering, 14(7): 45-50.