**Frontiers in Computing and Intelligent Systems**
ISSN: 2832-6024 | Vol. 3, No. 2, 2023

# Kerberos authentication protocol implementation based on DHR atypical construct

**Tingan Zhang**

Cyberspace Security Academy, Chengdu University of Information Technology, Chengdu 610225, China.

**Abstract:** As a commonly used identity authentication protocol, the Kerberos authentication protocol uses a symmetric password system, which has significant efficiency advantages but also brings risks such as password guessing and replay attacks. Some existing research combines public key cryptography with Kerberos to greatly enhance its security, but the large amount of computation required for public-private key cryptography leads to lower efficiency. This paper proposes a solution that uses the DHR non-typical construction to provide dynamism and randomness to the Kerberos authentication protocol to improve its security while minimizing efficiency losses. Experimental results show that the improved protocol can resist attacks while ensuring efficiency.

**Keywords:** Kerberos; DHR atypical construction; Symmetric password; Password guessing.

## 1. Introduction

Kerberos authentication protocol is considered as a mature and well-performing authentication system[1], which is based on ticketing mechanism and key distribution to complete the authentication process through the interaction between Kerberos TGS and Kerberos Client. However, the existing Kerberos authentication protocol has some security vulnerabilities in key management and distribution, which can be exploited by hackers and lead to security problems such as network attacks and data leakage.

To solve these security problems, this paper proposes an improved Kerberos authentication scheme, which enhances the security and reliability of the key by providing a key scheme for Kerberos authentication protocol through the idea of DHR atypical construction to improve the security and reliability of Kerberos authentication protocol.

The main contribution of this paper is to propose a new Kerberos authentication scheme that implements the DHR atypical construction by introducing a key generator for sequence ciphers, and enhances the security and reliability of the key by introducing randomness and dynamism to the key. Also, this paper analyzes the advantages and disadvantages of the scheme and compares the existing Kerberos authentication protocols with other secure authentication schemes to verify the superiority and feasibility of the scheme in this paper.

This paper is organized as follows: Chapter 1 introduces the related work and research background; Chapter 2 introduces the atypical construction of DHR to be used in this paper; Chapter 3 details the Kerberos improvement scheme proposed in this paper; Chapter 4 compares the improvement scheme proposed in this paper with other schemes.

## 2. DHR Atypical Structure

The Kerberos authentication protocol is based on a trusted third-party KDC, which has three parties: the user, the KDC, and the application server. The ticket carries a timestamp and expiration time, and with the ticket, the communicating parties can communicate without authentication until the ticket expires under the premise of machine time synchronization. The KDC in turn contains two parts, AS and TGS, for authentication and ticket authorization respectively.

There are several versions of Kerberos, the common ones are v4 and v5. The version studied in this paper is v4, and the encryption algorithm uses the Des algorithm.

Table 1 lists the symbols that will be used later and their meanings, among which User Info contains information such as user ID, domain to which the user belongs, and timestamp.

**Table 1.** Symbol Definition

| Symbol | Meaning |
|---|---|
| **AS** | Authentication Server |
| **TGS** | Ticket Granting Server |
| **SS** | Application Server |
| **Kc** | User Key |
| **Kt** | TGS Key |
| **Ks** | SS Key |
| **Kst** | Session key for communication with TGS |
| **Kss** | Session key for communication with SS |
| **EKc** | Encryption with Kc |
| **TGT** | TGS Tickets |
| **ST** | Tickets for communication with SS |
| **IDx** | X's ID |
| **Ke** | Ke used for encryption |

### 2.1. kerberos Certification Process

The Kerberos authentication process can be divided into three steps, the first step is to communicate with AS for authentication, the second step is to communicate with TGS to obtain a ticket, and the third step is to communicate with SS. The user key is used to communicate in the first step, and the last two steps both use the session key generated earlier to communicate. The specific process is as follows:

Step 1:

Client->AS: UserInfo

User sends identity information in clear text to AS

AS->Client: EKc[ UserInfo || Kst ] || TGT

Select the corresponding key according to the user's identity information to encrypt and transmit the session key Kst used in the second step, and together transmit the TGT that only TGS can decrypt, where TGT=EKt[ UserInfo || Kst ]. The user obtains it and decrypts it with his own key to obtain

Kst.

Step 2:

Client->TGS: EKst[ UserInfo || IDs ] || TGT

TGS->Client: EKst[ UserInfo || |Kss ] || ST, ST=EKs[ UserInfo || Kss]

TGS decrypts TGT with its own key to get Kst and verifies the identity, and issues ST if it is consistent.

Step 3:

Client->SS: EKss[ UserInfo ] || ST

SS Use your own key to decrypt the ST, get the Kss and authenticate the identity information, then you can start communication if it is consistent.

Until the ST expires, the Client can always use the ST for authentication and does not need to repeat the first two steps of communication.

## 2.2. Attack method

(1) Password guessing attack

In the three steps of this authentication process, symmetric passwords are used, where the key used in the first step is the key used in user registration, and the user key is usually constant for a long time, so it is subject to password guessing attack. The user key usually carries some information about the user or simple laws, so it becomes more effective to use dictionary attacks to crack the key.

(2) Replay attack

Assuming that the first step is secure, in the second and third step of communication, the session key generated by KDC is used, which is time-sensitive and secure for password guessing. It may be subject to replay attack. Although the ticket has an expiration time, it is required that the machines on both sides of the communication are clock synchronized, and the attacker can just use this to perform replay attack.

(3) Others

In addition to the above two relatively effective attacks against Kerberos, Kerberos authentication systems have the same security issues as other security systems: key storage issues, system program security issues, and so on.

## 2.3. Improvement Program

The literature[2] replaces the symmetric cryptosystem used in the original Kerberos authentication process with a public-key cryptosystem that uses the RSA algorithm for communication and requires the user to sign the timestamp, which on the one hand avoids the problem of the lower security of the symmetric cryptosystem itself, and on the other hand is also resistant to replay attacks by signing the timestamp, but due to the use of the RSA algorithm for multiple communications, the efficiency decreases a lot compared to the original symmetric cipher. Similarly, the literature [3] also proposes the use of public key encryption and decryption algorithms for communication. In the literature [4], the ECC algorithm is used to exchange keys, allowing the initial authentication process (i.e., the first step above) to use a different key for each authentication, thus improving the security of the system. And it is slightly more efficient compared to the previous scheme because only the keys are exchanged by this algorithm and it is not used for subsequent multi-step communication.

Some of the improved schemes replace the timestamp with a random number or sequence number to avoid replay attacks. In the literature [5], the security problem of replacing the timestamp with a random number is analyzed, and a scheme combining a random number and a sequence number is proposed to identify the messages sent by the user to improve the security of the system, while the communication process also uses a public key cipher, which has the same efficiency problem as the above scheme. The literature [6] stipulates that the user and the KDC store multiple random keys and random numbers respectively, and each communication then randomly selects a random key and random number to communicate, and the communication process still uses DES, and the authentication efficiency of this scheme is almost no different from the original version, but how to synchronize between various random numbers in this scheme should be the biggest security problem, which is not mentioned in the original text, and the frequent data updates can also cause considerable stress on the system program. The literature [7] proposes a dynamic cryptosystem using logistic mapping to generate the keys used by the DES algorithm, allowing the keys to have dynamic randomness while ensuring the efficiency of authentication, but the scheme is still difficult to apply in concrete implementations.

In this paper, we propose a simpler scheme that uses the DHR atypical construction to provide dynamicity and randomness to the keys, and achieves the level of security in the above scheme while guaranteeing authentication efficiency.

# 3. DHR Atypical Structure

In the existing cyberspace, the network system has a high risk due to its lack of defense against unknown attacks, the system itself having known or unknown vulnerabilities, and its inability to cope with the rapidly evolving cyberattack techniques, which are called endogenous security problems in cyberspace. In order to solve this problem, the domestic team has proposed a mimetic defense with dynamic heterogeneous redundancy construct (DHR construct) as the key core in several years of research and practice, which is intended to be used to solve the above endogenous security problem[8].

DHR constructs are proposed to provide dynamic, random, and heterogeneous nature to network systems, and to cope with different security problems, DHR constructs are subdivided into two categories: typical and atypical constructs, as shown in the following figure.

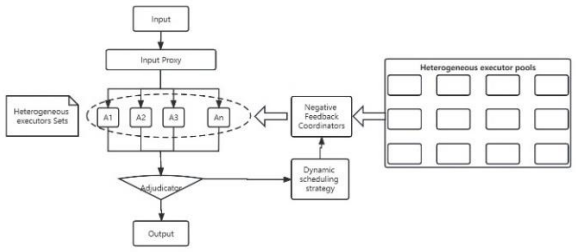


**Figure 1.** DHR Typical construction

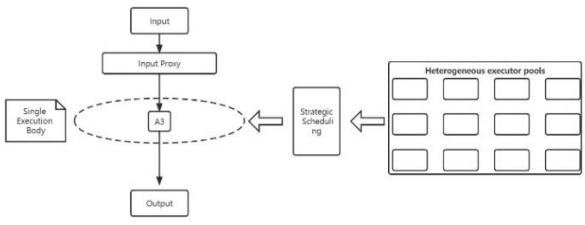


**Figure 2.** DHR Atypical construction

The difference between these two types of models is that in the typical construct, several heterogeneous executors are

selected to execute together for each input, while in the atypical construct, only one executor is selected to execute. The feedback and voting mechanisms in the typical construct of DHR make it impossible for the result of an attack to pass the voter even if one of the executors in the system is under attack or has a vulnerability, and the executor with the vulnerability will be fed back and cleaned. DHR atypical constructs are more efficient and take less space. The atypical construct is based on the fact that the target system is functionally recoverable and allows a certain degree of information leakage within a certain time window, using a dynamic, randomized approach that affects the reproducibility of the effect of the attack, thus making the attack itself less or less valuable for sustainable exploitation.

The kerberos authentication protocol to be improved in this paper operates in an open network, where both security and efficiency are key considerations for the authentication protocol. The attack on this protocol is mainly to analyze and crack the data captured in the communication, and once the user key is cracked, the attack will be continuous and reproducible, while the authentication system itself generally does not have functional problems such as vulnerabilities. the complexity of the typical construction of DHR and multiple constructs do not meet the needs of the Kerberos authentication protocol, so we try to implement one based on the atypical construction of DHR Kerberos authentication protocol based on the DHR atypical construct to solve the above problems.

The key of the DHR atypical construct is that the information system needs to reserve multiple executors and select one of them to execute when the system is running, so as to provide the information system with dynamism and randomness by dynamically selecting different executors at each run.

In Kerberos authentication protocol, both the encryption and decryption algorithms and the keys used in the algorithms can be regarded as executors, but there are fewer encryption and decryption algorithms to choose from, and the dynamism and randomness are not guaranteed. Therefore, in this paper, the keys under fixed algorithms are regarded as executors in the information system, and the pool of heterogeneous executors is regarded as a key pool from which different keys are selected to encrypt and decrypt the transmitted data, thus providing dynamic randomness to the information system.

# 4. Kerberos Authentication Protocol Combined with Zuc Algorithm

## 4.1. Zuc Key Generator

Sequence ciphers, also known as stream ciphers, have the advantages of fast operation, simple implementation, and no or limited error propagation. The keys used for sequence cipher encryption and decryption are generated by a keystream generator, and the higher the randomness of its keys, the closer the sequence cipher system is to one-at-a-time, so the security of a sequence cipher system is largely determined by the keystream generator. However, sequence ciphers also have fatal drawbacks compared to other symmetric ciphers: low diffusivity, insensitivity to insertion and modification, means that sequence cryptosystems may make the plaintext statistically specific and thus subject to specific attacks, such as differential attacks [9].

The function of a keystream generator is to obtain a shorter seed key on the basis of which a longer sequence of keys is generated, which can also be seen as multiple short keys of fixed length. How to make the generated key stream with higher randomness is the core problem of constructing a key stream generator, and the most researched key stream generators are those consisting of multiple shift registers.

The key used for encryption and decryption in the kerberos authentication process is obtained through a keystream generator for serial ciphers, depending on the security and reliability of the key generator. If the keystream generator is capable of generating high-strength keys, then it is theoretically reliable to obtain the keys used for encryption and decryption through this keystream generator.

In summary, this paper will use the keystream generator as described above to generate a number of keys to form a key pool, which will be used as the key for Des encryption and decryption in the Kerberos authentication process. This will enhance the security of the ciphertext by using a different key for each transmission, and the diffusion and obfuscation properties of Des will overcome the low diffusion problem inherent in sequential cryptosystems.

The Zuc algorithm (Zuc), originally designed for security problems in communication networks, is a Chinese self-designed, word-based synchronous sequence cryptographic algorithm, published as national standard GB/T 33133-2016 in October 2016.The algorithm structure is divided into three layers, namely, linear feedback shift register, bit reorganization, and nonlinear function.The algorithm in the key output phase each cycle for the output of a 32-bit keyword. The linear feedback shift registers used in common stream cryptosystems all use m-sequences on a binary domain or some diffusion domain in the binary domain, which have obvious multiple linear relationships, making them vulnerable to correlation attacks, while the linear feedback shift registers in the Zuc algorithm use m-sequences in the prime domain GF(2^31-1), which have long periods, good statistical properties, and weak linear structure , the bitwise relationship compliance is low [10]. The Zuc algorithm has been studied to resist weak cryptanalysis, guess-deterministic analysis, algebraic analysis, and other analysis methods, and has very high security properties [11]. Therefore, it is chosen in this paper as the key generation scheme in this topic.

In the original Kerberos authentication protocol, the key used for encryption during transmission is the md5 value of the password set by the user, and the Zuc cipher accepts a 128-bit initial key seed, which can be used directly, minimizing the modification of the original Kerberos protocol. the reliable security of the Zuc algorithm ensures that the key sequence provided by it also has a high randomness. to confer dynamic randomness to the ciphertext transmitted during the authentication process.

## 4.2. Certification Process

### 4.2.1. Registration Stage

The user registers his userId and password to the AS, which stores the userId and the user password Kc after the md5 operation, a step consistent with the original kerberos version. In addition, both parties store an additional authentication serial number i with an initial value of 0, which is used to select the key from the key pool during the authentication phase.

### 4.2.2. Certification Stage

Similar to the original authentication process, only part of the process of encryption and decryption in the authentication process is changed. Among the three steps in the original

authentication process, the risk lies in the ciphertext security during the first step of transmission, and the last two steps all use the random session secret key for encryption and decryption transmission, and the latter two steps of the authentication process are not modified in this paper.

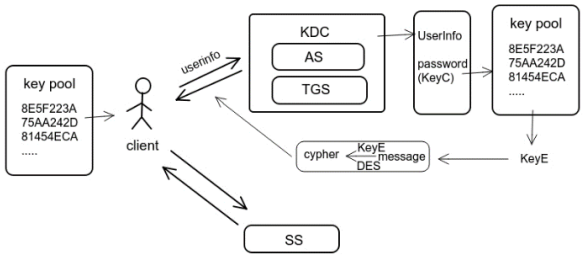


**Figure 3.** Certification Process

As in Figure 3, authentication starts: Client->AS: UserInfo / AS->Client: EKe[ UserInfo || Kst ] || TGT

The user also sends the user information to AS in plaintext only. At this time, the AS side selects the corresponding user key and serial number according to the user identity information, uses the user key as the initial key seed of Zuc's key generator, generates several sub-keys, and then selects the key key by serial number i. The key is used as the key of des algorithm to encrypt TGS credentials, session key and other information to send back to the user. The user obtains the key and decrypts the ciphertext in the same way, and obtains the session key and TGS credentials for subsequent communication in the ciphertext.

Client->TGS: EKst[ UserInfo || IDs ] || TGT / TGS->Client: EKst[ UserInfo || Kss ] || ST

The TGS then encrypts the identity information with the newly obtained session key and sends it back to the TGS with the TGS credentials. The TGS decrypts the credentials to obtain the session key and uses the session key to decrypt the cipher text to compare the identity information in the cipher text with that in the credentials, and if it is consistent, it generates a new key as the next session key to communicate with the server.

After a complete authentication, the user side and the server side will add 1 to the authentication number to ensure that the next authentication can use a different subkey. If the authentication fails or is interrupted in the middle, neither side will add 1 to the authentication number to ensure the synchronization of the authentication numbers of both sides.

### 4.3. Security Analysis

The kerberos authentication studied in this topic converts the security of the 128-bit key, which is the riskiest part of the original authentication process, into the security of several subkeys generated by the 128-bit key.

For an attacker, the ciphertext obtained from the communication channel is one of several subkeys generated by the unknown user key and des encrypted, and the ciphertext in the data obtained during each authentication is encrypted by a different key, even though all these keys are generated by the same key seed and selected one by one from the abstract key pool, but Zuchon's key generator guarantees the weak correlation between the keys, so that even if an attacker obtains the keys used for several consecutive authentications by some means, he cannot get the keys used for subsequent authentications, i.e., he cannot pass the authentication.

In terms of attack methods, the original version of authentication is targeted by brute force cracking and dictionary guessing of the 128-bit user key, which is no longer secure with today's computing power, while the improved authentication scheme targets not only the 128-bit key, but also a key pool with several keys generated by a hidden 128-bit secret key, which appears from the outside as if each authentication uses a The authentication process becomes random and dynamic, and the difficulty of the attack rises exponentially.

Moreover, due to the existence of the authentication number, even if an attacker obtains data from a certain authentication process and attempts to replay the attack, the data will no longer be valid due to the change of the authentication number, which can effectively prevent replay attacks.

## 5. Program Comparison

### 5.1. Safety Comparison

For different Kerberos improvement schemes, the core goal is to ensure the security of the session secret key transmitted during the first step of communication in the authentication process, abstracting its communication process into a simple communication between the authentication server side and the client side, as shown in Figure 4.

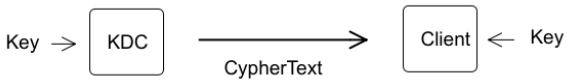


**Figure 4.** General Communication Process

For the original authentication, once the user key is cracked, the ciphertext of any subsequent communication can be decrypted by the user key as long as it is intercepted by the attacker to get the session secret key of subsequent communication, and the authentication security cannot be guaranteed. Some of the ways to enhance the key strength by using public key cryptosystem aim to enhance the key strength and ensure that the key cannot be obtained by brute force cracking.

The improved scheme based on the atypical construction of DHR proposed in this thesis improves the above communication process into the way shown in Figure 5. For kerberos authentication, the cipher text is encrypted by the session secret key, which has a valid time (generally 8 hours), and the session secret key will be invalidated after this limited time. In this way, even if an attacker gets the ciphertext of a certain authentication and wants to intercept and attack the communication process, the session secret key contained in the ciphertext will have exceeded the limited time and lost its validity by the time the attacker cracks the key, and the authentication will fail. Even if the key is cracked, it cannot be applied to the decryption of the subsequently obtained ciphertext because the key used in each encryption is dynamically changing.

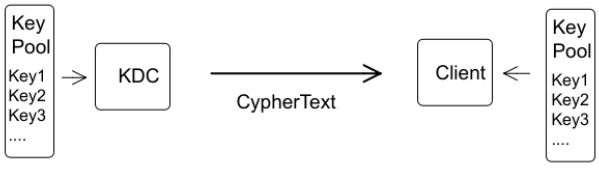


**Figure 5.** Improved communication process

The main difference between the scheme in this thesis and others is that it provides endogenous security for the communication process, accepting a certain level of attacks, but using the dynamic randomness assigned to it by the DHR model so that its attacks cannot be applied to the authentication process.

## 5.2. Efficiency Comparison

Efficiency Comparison The purpose of the scheme proposed in this topic compared with the schemes proposed by other scholars is to change the way of encryption and decryption to improve the security of the ciphertext with a more secure algorithm, while the purpose of the scheme proposed in this paper is to provide dynamic randomness to the key to make the ciphertext secure.

Through the above theoretical analysis, the improved scheme proposed in this topic can already effectively prevent the common attacks against the kerberos authentication protocol. In order to further analyze the authentication efficiency of different authentication schemes, the author conducted experiments on the encryption and decryption algorithms used in the authentication process and the authentication process.

First, experiments were conducted on the speed of the cryptographic algorithms used in authentication, including the Des encryption algorithm, the speed of generating different numbers of Zuc algorithms, and the Rsa algorithm, and the experimental results are shown in Figure 6 and Figure 7, respectively.

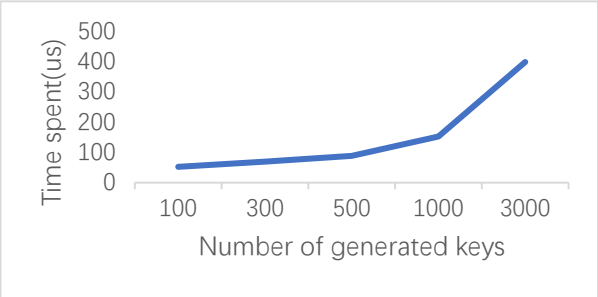


**Figure 6.** Zuc generation efficiency experiments

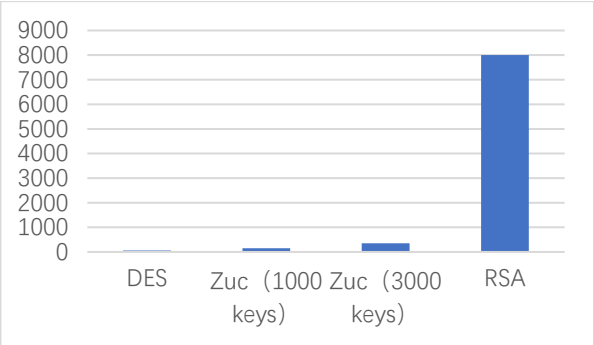


**Figure 7.** Algorithm Comparison

Then compare the original kerberos, the improved scheme 2 proposed in this paper, scheme 3 proposed in literature[12], scheme 4 proposed in literature [4], because the key flow generator used in this paper is dynamically generated key pool, the key pool size is not fixed, so in the experimental process to take the key pool size of 100 and 1000 when different cases for testing, the experimental environment is a dual-core, running memory of 4G centos7 cloud server, the time spent on the original authentication process is set as a unit time, the experimental results are shown in Figure 8.

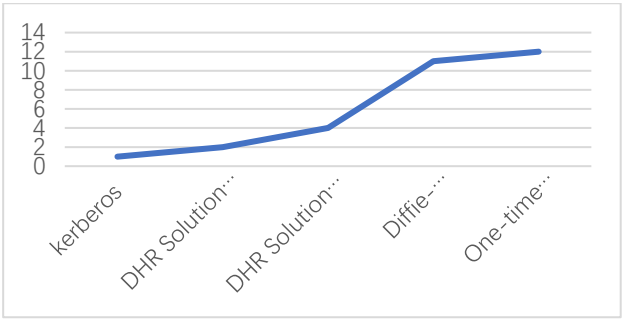


**Figure 8.** Comparison of different solutions Kerberos

From this, it can be seen that since schemes 3 and 4 use public-key cryptosystems, the efficiency of addition and resolution is significantly reduced, and even when the key pool size in this topic is 1000, the authentication efficiency of the scheme proposed in this paper is better than that of schemes 3 and 4.

# 6. Summary

In this thesis, we propose a new and improved scheme for the kerberos authentication protocol, where the keys used for encryption and decryption are obtained through a key generator for sequence ciphers. Through detailed analysis and experimental verification of the scheme, we demonstrate that the scheme can improve the performance and reliability of the system while ensuring authentication security.

In this paper, we mainly improve the traditional kerberos authentication protocol to solve the security problems in key management and transmission, and also optimize the shortcomings of other improved schemes that significantly reduce the authentication efficiency. Experimental results show that the improved protocol outperforms the traditional kerberos authentication protocol and other improved schemes in terms of security, performance and reliability to a certain extent.

Although the improved scheme proposed in this paper has achieved certain results, there are still many aspects that need further research and improvement, for example, the key pool can be further optimized by a caching scheme to obtain the speed of keys. It is hoped that the scheme can provide a reference for researchers in related fields and provide ideas for solving authentication security problems in practical applications.

# References

[1] Neuman C,Steiner J. Authentication of Unknown Entities on an Insecure Network of Untrusted Workstations[J]. Proceedings of the Usenix Workshop on Workstation Security, 1988.

[2] Improvement of Kerberos Protocol Based on Dynamic Password Regime_ Huang [J].

[3] Ganesan R. Yaksha: Augmenting Kerberos with Public Key Cryptography[C]//Symposium on Network & Distributed System Security, 1995.

[4] Ma Limin,Zhang Wei,Song Ying. An enhanced Kerberos protocol approach based on one-time password and its formal analysis_ Ma Limin [J]. Information Network Security, 2019(10): 57-64.

[5] Security Analysis and Countermeasure Research of Kerberos Protocol_Ping Yang [J].

[6] Wang Ruifang. Research on Kerberos password recovery and security hardening technology_Wang Ruifang [D]. Zhengzhou University, 2020.

[7] A method to improve the Kerberos protocol using dynamic cryptosystem_Hu Hanping [J].

[8] Mimetic Security Defense in Cyberspace_Wu Jiangxing [J].

[9] Shi Zhen. Research on LFSR-based Sequential Cryptographic Correlation Attack Method_Shizhen [D]. Strategic Support Force Information Engineering University, 2022.3

[10] GPP_LTE International Encryption Standard ZUC Algorithm_Feng Xiutao [J].

[11] Zuchongzhi Sequence Cipher Algorithm_Feng Xiutao [J]..

[12] A Method for Using Diffie-Hellm... Key Negotiation Improved Kerberos Protocol_Chen Feng [J].