Designing And Implementing a Chat System with Enhanced Security Via AES Encryption Methods

Heyang Liu

College of Computer Science and Technology, Harbin Institute of Technology, Weihai, Weihai, 264209, China

201026010137@stu.swmu.edu.cn

Abstract. As technology has advanced, we have ushered in the age of the Internet, where online communication tools have become the preferred method for information exchange. While this shift offers unprecedented convenience and efficiency, it also introduces a slew of security concerns. Messages traversing the vast expanse of the internet are vulnerable to interception, tampering, and even fabrication by malicious entities. Such actions can disrupt communications and lead to significant repercussions for all involved parties. Given these threats, the imperative for robust encryption becomes clear. Typically, to safeguard the contents of a message, the sender encrypts it using a specific key. In the absence of this key, adversaries find it challenging, if not impossible, to decipher the encrypted message. Upon receipt, the intended recipient, equipped with the necessary key, decrypts the message to access its original content. Beyond encryption, digital signatures are often employed to verify the authenticity of the sender and prevent message forgery. Such cryptographic measures are pivotal in maintaining the sanctity of digital communications. This article embarks on an exploration of prevalent encryption techniques employed by contemporary online communication tools. It delves deeper into the AES (Advanced Encryption Standard) method, spotlighting its mechanics and efficacy. Culminating in a tangible application, the article presents the design of a rudimentary chat system underpinned by AES encryption, showcasing its potential in real-world secure communication scenarios.

Keywords: Encryption algorithm; AES; Web chat.

1. Introduction

The primary objective of this paper is to architect a straightforward encrypted chat platform, ensuring the confidentiality and security of online dialogues for all participants. To this end, the paper delves into foundational concepts in cryptography, highlighting the HTTPS protocol and the AES encryption technique. Subsequent to this, the research examines the encryption methodologies and operational frameworks employed by prevailing online chat platforms. Drawing insights from this preliminary research, a streamlined and encrypted chat system is then crafted. This system encompasses core features such as user registration, login capabilities, and chat functionalities, all while maintaining a robust framework for the secure and efficient transmission of messages, courtesy of adept encryption practices.

2. Background and Relevant Theories

2.1. The Underpinnings of the HTTPS Protocol

Hypertext Transfer Protocol Secure is an extension of the HTTP protocol that incorporates the SSL protocol to ensure security through encryption and authentication during the transmission process. SSL is positioned between the application layer and the transport layer in computer networks. Instead of directly delivering application layer data to the transport layer, the sender first hands it over to the SSL layer [1]. The SSL layer encrypts the data using encryption keys, adds its own SSL header, and then passes it to the transport layer. Upon receiving the data, the recipient's transport layer forwards it to the SSL layer. The SSL layer removes the SSL header from the data, decrypts the

message using the corresponding key, and then delivers the message to the application layer [2]. The structure is depicted in Fig. 1.

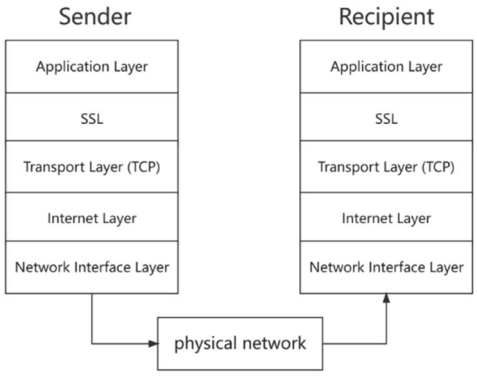

Fig. 1 The messaging path in the HTTPS protocol (Photo/Picture credit: Original).

2.2. Fundamentals of AES Encryption

AES is a block encryption algorithm proposed by NIST in 2001 to replace the insecure DES algorithm. In the AES algorithm, the input plaintext should be 128 bits (16 bytes), and the supported key lengths can be 128 bits, 192 bits, and 256 bits [3].

The complete process of the AES encryption algorithm is shown in Fig. 2. AES algorithm is an iterative algorithm over a finite field. When the initial key is 128 bits, it requires 10 rounds of iteration (12 rounds for a 192-bit key, and 14 rounds for a 256-bit key). Except for the last round, each iteration consists of four steps: SubBytes, ShiftRows, MixColumns, and AddRoundKey. In the last iteration, the step of MixColumns is omitted.

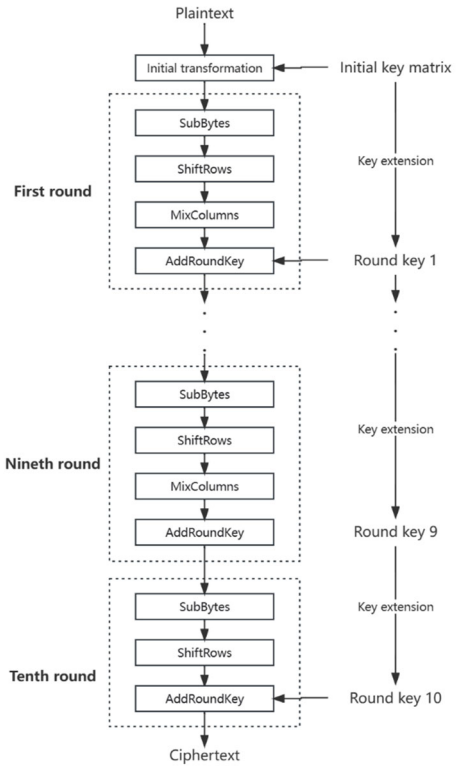

Fig. 2 AES algorithm flow (Photo/Picture credit: Original).

The plaintext of 16 bytes is first arranged into a 4x4 byte matrix Mp, and then undergoes an initial transformation. In the initial transformation, Mp is XORed with the initial key matrix Mk byte by byte, resulting in a new matrix M1, which serves as the input matrix for the subsequent iterations.

In each round of iteration, a byte substitution is performed. During the byte substitution process, each byte of the input matrix Mi (Mi being the input matrix for the i-th round of iteration) is substituted with the corresponding byte from the S-box, resulting in a new matrix Mi'.

Next is the ShiftRows. During the ShiftRows process, the first row of the matrix Mi' remains unchanged, the bytes in the second row are shifted one position to the right, the bytes in the third row are shifted two positions to the right, and the bytes in the fourth row are shifted three positions to the right, resulting in the matrix Mi".

In the column mixing, the matrix Mi" is multiplied on the left by a given matrix M, resulting in the matrix Mi'".

Finally, it is the AddRoundKey step. The matrix Mi'" is XORed with the round key matrix Mki (Mki being the round key matrix used in the i-th round of iteration, obtained from the initial key matrix through key expansion) byte by byte, resulting in the final output matrix Mi+1. If i is 10 (the tenth round of iteration), M11 is the final ciphertext matrix Mc.

The round key matrix is obtained by expanding the initial key matrix. The columns of the initial key matrix are numbered as M, M, M, and M. The round key matrix used in the i-th iteration is composed of M[x] to M[x+3], where x=4*i [4].

The calculation of M[x] follows the following formula:

$$\mathrm{M[x]} = \begin{cases} \mathrm{M[x-4]} \oplus \mathrm{M[x-1]}, & \mathrm{x\%4} \neq 0 \\ \mathrm{M[x-4]} \oplus \mathrm{T(M[x-1])}, & \mathrm{x\%4} = 0 \end{cases} \quad (1)$$

The operation of the T function is as follows:

Byte rotation: Each byte in the column is shifted up by one position, and the first byte is moved to the end.

Byte substitution: Each byte in the column is replaced with the corresponding byte in the S-box.

Round constant XOR: The round constant matrix is a 4x10 matrix W, and each column corresponds to the round constant for the corresponding iteration. The column obtained by byte substitution is XORed with the corresponding round constant.

2.3. Server-Based Communication in Chat Systems

A server-based chat system is a system that uses a server as an intermediary to transmit messages and enable real-time communication. Typically, this system consists of three main components: client, server, and database. The client displays the user interface, allowing users to search for other users, input messages, etc. The server receives messages from clients and then forwards them to the intended recipients. The database stores all user account information and chat records.[5] This type of chat system has several advantages:

By utilizing the server as an intermediary, messages can be checked for validity and integrity, ensuring the legitimacy of the messages and avoiding duplication.

It enables group messaging, saving time by sending a message to multiple recipients simultaneously.

The database on the server can store chat records, facilitating message backup and recovery.

Each client only needs to establish a connection with the server, avoiding the complex dependency relationships in a mesh-like structure.

However, this chat system also has its drawbacks:

The security of the server needs to be considered to prevent data stored in the server's database being stolen or tampered with. It requires a high-performance server. When there is a large number of concurrent users and from message volume, the stability and efficiency of the server need to be ensured [6].

3. Survey of Common Web Chat Applications

3.1. WeChat

WeChat is one of the most well-known chat applications. In order to ensure the safety of users, WeChat adopts a variety of encryption methods. WeChat uses symmetric encryption AES encryption algorithm; users use the same key when sending and receiving messages. To verify message integrity and authentication, WeChat uses digital signature technology, where the sender signs the message with a private key, while the receiver verifies the signature with a public key. In addition, WeChat uses SSL protocol and TLS protocol to ensure the security of the channel and uses encryption algorithms to store user data and chat records to protect user privacy [7].

MSN

MSN is a kind of communication application launched by Microsoft Corporation, which supports text chat, voice chat, video call and other functions. The encryption algorithm used by MSN is the RC4 algorithm, which is a stream cipher algorithm that XOR the message plaintext with the key to produce the ciphertext. Since it uses the same key for encryption and decryption, it is also a type of symmetric encryption. MSN uses RC4 algorithm to encrypt messages in the process of message transmission to protect user privacy and data security [8].

4. System Design and Implementation

4.1. User Registration Mechanism

To send messages using the chat system, users should first register an account. Users send their entered account information to the server, and upon receiving the information, the server processes it and saves it in the user information database. After this, users have successfully registered their account.

When transferring user account information, confidentiality is essential. This chat system intends to use AES encryption for message encryption. However, both the communicating parties should be aware of a shared AES key for encryption and decryption. But when users register an account, it is their first communication with the server, so AES encryption cannot be used to encrypt account information.

To solve the above problem, the account information provided by the user during registration is encrypted and transmitted using the HTTPS protocol. HTTPS is an asymmetric encryption method. The server only needs to publish its public key on the website, which can be used by anyone to encrypt messages. The server uses its private key to decrypt the messages. This allows secure message transmission between the user and the server without pre-agreed keys.

To ensure that a more secure AES encryption method can be used for subsequent message transmission, the user needs to add an AES key field when sending their account information. When the server receives the registration information sent by the user, it stores the various fields in the corresponding positions in the database (such as account, password, phone number, etc.), and also stores the AES key. This way, both the client and the server can use the AES key for subsequent communication.

Another issue is that the database in the server stores all user account information, which poses a security risk. If an attacker is able to infiltrate the server and steal user information, the impact would be devastating. Therefore, when storing user information in the database, measures are taken to protect it. One approach used is encrypting the information using a hash function. When storing the account information (especially passwords), the information is first input into the hash function for encryption, and then the resulting hash is stored in the database. If the attacker does not know the hash function used, it is almost impossible for them to reverse-engineer the user information from the encrypted data. These measures greatly ensure the security of user information.

4.2. User Authentication and Login

If the user already has an account, they can proceed with the login operation. Once the user enters their account and password in the corresponding fields and clicks the login button, the account and password information will be encrypted using AES encryption and sent to the server. Upon receiving the cipher, the server will decrypt it using the corresponding AES key to obtain the user's input information.

The server first inputs the user's password into the corresponding hash function, obtaining an output result H. Then, the server retrieves the encrypted password H0 from the database corresponding to the user's input account. If H=H0, it means that the user has successfully entered the password [9]. To ensure a more secure login process, a verification code mechanism can be added. By retrieving the user's mobile phone number from the database based on the entered account, the server sends the verification code Y0 to the corresponding mobile phone number. The user needs to fill in the received verification code and send it to the server using AES encryption. Upon receiving the user's filled-in verification code Y, the server compares it with Y0. If they are the same, the verification is successful.

4.3. Secure Chat Environment

When the user successfully logs in, they can start sending messages. All messages sent by the user will always be processed by the system into message blocks like the one shown in Fig. 3.

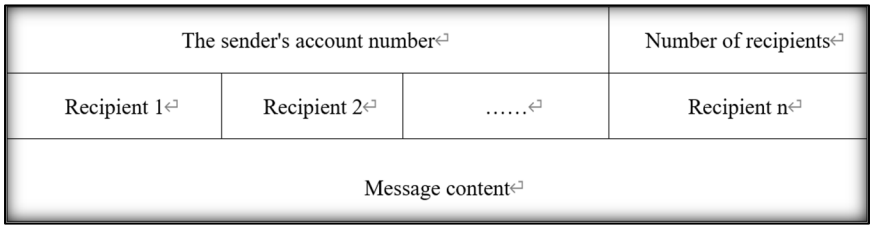

Fig. 3 Message block (Photo/Picture credit: Original).

The sender account location will automatically fill in the user's account. The recipient count location will fill in the number of accounts selected by the user, and the next n locations will fill in all the recipient account numbers selected by the user. The last location will fill in the information entered by the user. This is the unit when the user sends a message.

After the user edits the message and clicks the send button, the message block will be encrypted using AES encryption and sent to the server. The server decrypts it using the sender's AES key, obtaining n recipient account numbers. After necessary processing and checking of the message content, the server will generate n new message blocks, with the sending account location of each message block set to the initial sender's account, and the recipient location set to the n recipient account numbers specified by the sender. The server encrypts them using the corresponding account's AES key stored in the database, and then sends them to the corresponding accounts. The corresponding recipients can decrypt and access the information using the AES key, once they receive the message.

4.4. The overall flow of the system operation

The overall flow chart of users from registration to sending messages is shown in Fig. 4.

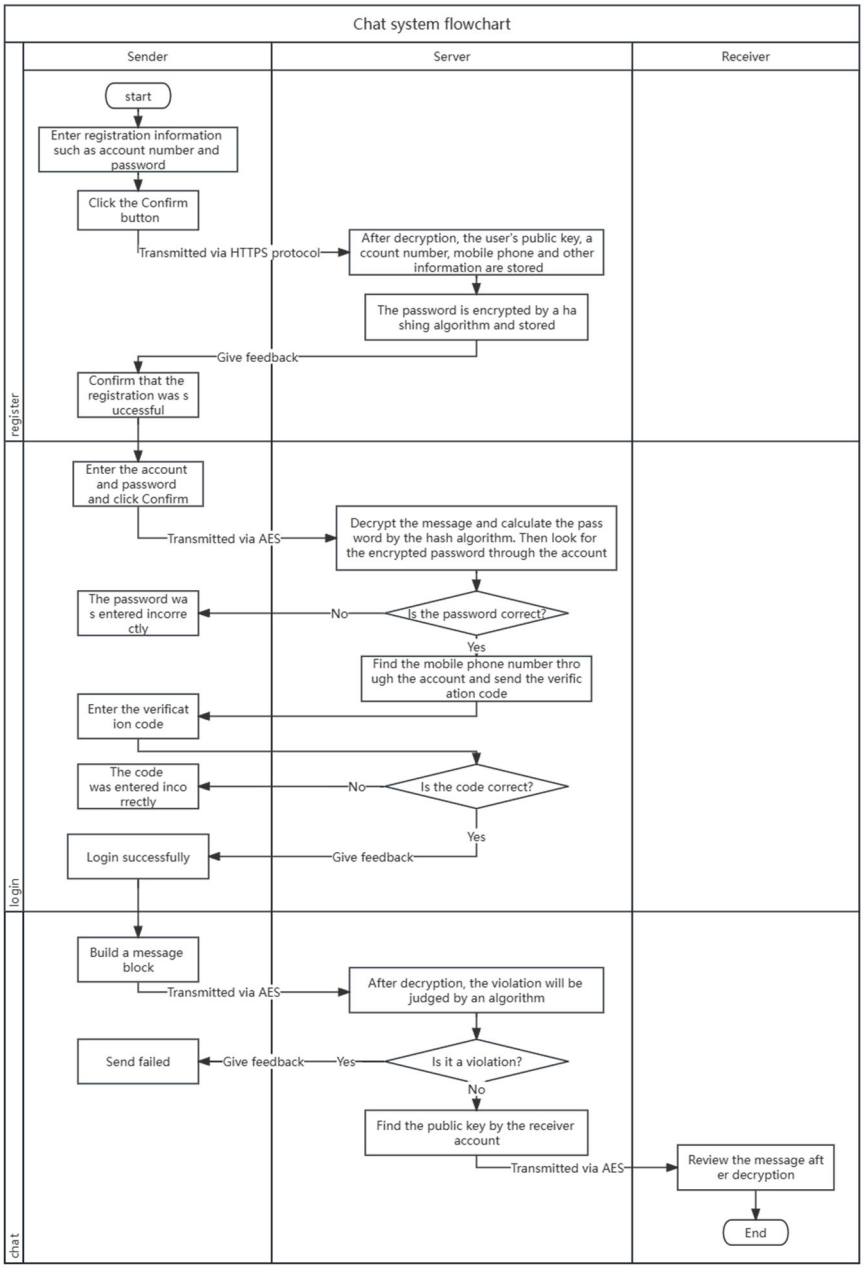

Fig. 4 Flow chart of users from registration to sending messages (Photo/Picture credit: Original).

5. Challenges and Limitations

The main problem is the load on the server. The system designed in this paper is a server as a medium to forward information system. When there are too many users registering, logging in, and sending messages at the same time, the server needs to process a lot of information, which leads to high requirements on the stability and efficiency of the server. Once the stability of the server does not meet the requirements, in this case, the server is likely to crash. If the efficiency of the server does not meet the requirements, the information processing time may be too long, which seriously affects the user experience. One solution is to turn the server side into a server farm, consisting of many servers that share database information. Using this method can greatly relieve the pressure of a single server and improve the communication efficiency [10].

6. Conclusion

This article studies the working principles of the HTTPS protocol, the working principles of AES encryption, and another cryptographic knowledge. It also explores the encryption methods and working principles used in existing chat applications and proposes improvements to design a simple network encrypted chat system. This system ensures secure and efficient transmission of information, thereby achieving the goal of safeguarding the information security of both parties involved in communication.

In the era of information technology, information security is becoming increasingly important as network development progresses rapidly. By designing a reasonable encrypted information transmission system, it can greatly assist in ensuring such security. Cryptography is continuously advancing, and in the future, more efficient and convenient encrypted chat systems will emerge.

References

[1] Wang, X.W., Wang, J.Z. (2007). Security Analysis and Improvement of SSL protocol. 2007 New Development of Communication Theory and Technology--Proceedings of the 12th National Conference on Communication for Youth (Volume 1).

[2] Huang, X.Q., Qiang, G. (2015). A Simulation Test Based on Improved SSL Protocol [Conference Paper]. International Symposium on Knowledge Acquisition and Modeling (KAM 2015).

[3] Zhang, D.L., Jiao, W.C. (2011). Research on AES and Analysis of Its Security [Conference Paper]. International Conference on Future Computer Science and Application, pp. 225-227.

[4] Kedia, R., Kumar, B., Banerjee, P., Jha, P., Kundu, T., &Dehury, M.K. (2023). Analysis and Implementation of Image Steganography by Using AES Algorithm. 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI).

[5] Kong, X.Y. (2019). End-to-End Encrypted Communication Optimization Design and Implementation. Master's Degree thesis, Central China Normal University, Wuhan.

[6] Zhang, C. (2017). Design and Implementation of Instant Messenger Based on DES and RSA Hybrid Encryption. Master's Degree thesis, Xiamen University, Xiamen.

[7] Liu, J., Li, X.Z. (2015). An Improved TLS handshake protocol. Proceedings of 2015 3rd International Conference on Machinery, Materials and Information Technology Applications (ICMMITA 2015).

[8] Gong, D.L., Huang, Y.H., Liu, F. (2010). Research and improvement of RC4 algorithm. Proceedings of the seventeenth Annual Conference on Information Theory of the Chinese Association of Electronics.

[9] Zhang, W.C., Li, H., &Cheng, G. (2018). Research and practice of password secure storage based on one-way salted slow hashing algorithm. Proceedings of 2018 China Hospital Information Network Conference (CHINC).

[10] Meng, G.P. (2009). High Availability Design of enterprise server group network. Proceedings of 2009 Annual Meeting of Metallurgical Branch of China Metrology Association, 2009.