未来技术学院
FUTURE TECH

AI Chatbot Based on RSA and AES Encryption Process

杨腾越 202364820861
罗景楠 202364870981
张越 202364820921
陈思涵 202330420212

未 来 技 术 学 院
F U T U R E T E C H

Contents

未来技术学院
FUTURE TECH

01

Project Background and Objectives

Build a visualized encrypted communication tool to realize RSA/AES encryption & decryption and interact with QQ chatbot.

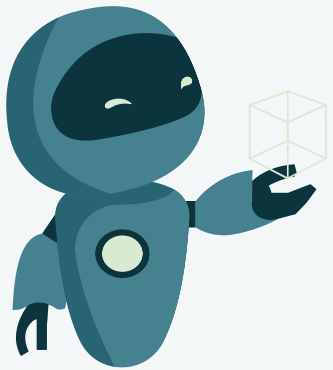
Project Background

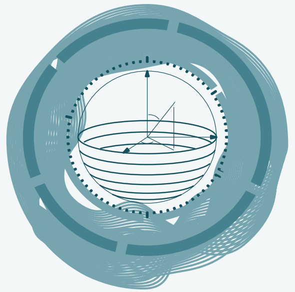
未来技术学院
FUTURE TECH

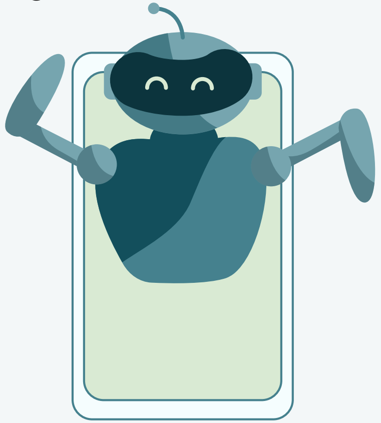
In today's information society, personal privacy and data security have become critical concerns in network communication.

With the widespread use of AI chatbots, more and more users are relying on chatbots in their daily lives, work, and studies to obtain services or exchange information.

However, traditional instant messaging platforms (such as QQ) do not, by default, provide end-to-end encryption for communications between users and chatbots. This creates the risk that private conversations may be intercepted, monitored, or exposed during transmission.





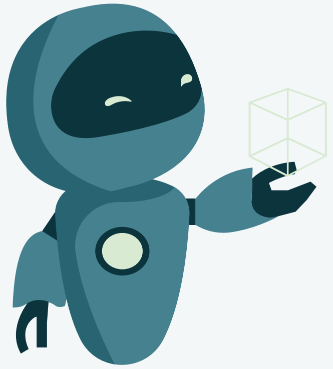


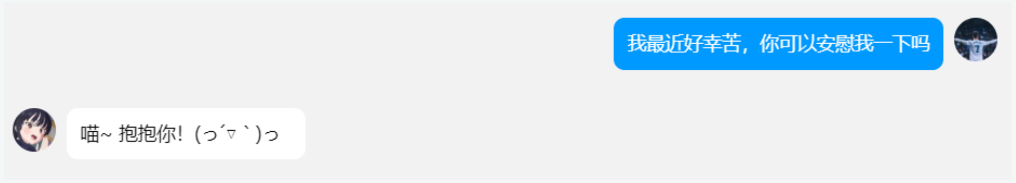
未来技术学院 FUTURE TECH

Objective

The goal of this project is to build a system that integrates **AI chatbots** with an **end-to-end encrypted communication mechanism**, allowing users to securely converse with a chatbot deployed on the QQ platform via a visualized front-end interface.

In this system, users only need to input plaintext messages, and the system will automatically complete **AES symmetric encryption** and **RSA asymmetric encryption**, then send the encrypted ciphertext to the QQ platform. Upon receiving the message, the chatbot will automatically decrypt it, generate a reply, re-encrypt the response, and return it to the user—ensuring that all content in the communication process remains encrypted, effectively preventing data leakage and man-in-the-middle attacks.





我最近好辛苦，你可以安慰我一下吗

喵~ 抱抱你！(っ´▽`)っ

未来技术学院 FUTURE TECH

Problems 02 Addressed and Work Description

This project addresses the lack of security mechanisms in communication processes, integrating encrypted image messaging and chatbot Q&A.

Problems to Be Solved

This project aims to address the lack of **privacy protection and message encryption mechanisms** when users interact with QQ-based AI chatbots. In current real-world applications, most communication content between users and chatbots is transmitted in plaintext via the QQ platform, posing a high risk of being intercepted or exploited during transmission on client devices, servers, or networks.

To tackle this, we designed and implemented an **end-to-end encrypted communication system**, in which the encryption and decryption processes are fully embedded within the user interface and the chatbot logic. This allows users to complete secure message exchanges without any knowledge of encryption technologies, ensuring **confidentiality**, **integrity**, and **resistance to eavesdropping** throughout the communication.

🛡️ **Construct a complete encrypted communication logic**

💻 **Integrate decryption functionality into the front-end interface**

🤖 **Develop encryption logic on the chatbot side**

📩 **Enable secure ciphertext message transmission and parsing**

🔄 **Support multi-round encrypted conversations**

Functional Modules

This project consists of the following main modules, each corresponding to specific development and technical tasks:

🔷 Front-End User Interface Module

Provides a visual operation interface that allows users to input plaintext, view responses, and automatically perform encryption/decryption operations.

🔷 Encryption Algorithm Module

Implements hybrid encryption logic combining RSA and AES to ensure data confidentiality during communication.

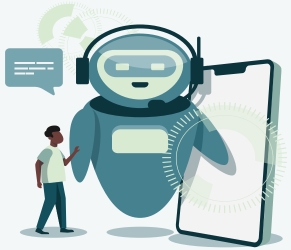
🔷 Chatbot Server -Side Module

Acts as a middleware to receive encrypted messages, handle decryption, generate AI responses, and send back encrypted replies.

🔷 Communication Transmission Module

Serves as the transmission layer to relay messages between the chatbot and the user.

Front-End User Interface Module & Chatbot Server-Side Module



•**Function**:
Provides a visualized interface that allows users to input plaintext, view responses, and automatically perform encryption and decryption.

•**Workflow**:
All message transmissions adopt a **hybrid RSA + AES encryption scheme** to ensure privacy throughout communication.
The entire architecture follows a secure design pattern of **"Key Exchange + Encrypted Chat"**, with two main stages:

1.Session Setup → **2. Secure Chat**

The user starts encrypted communication. The front-end generates an RSA key pair and sends the public key to the chatbot.

The chatbot generates a temporary AES key, encrypts it with the user's public key, and sends it back. The user decrypts it with the private key.

The user encrypts their message using the AES key (via GCM) and sends it through QQ.

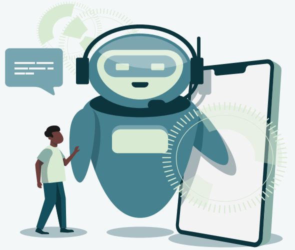
The chatbot decrypts the message, generates a response, encrypts it with a new AES key, encrypts that key with RSA, and returns both to the user.

Encryption Algorithm Module (Encryption Utility Layer)

🔐 **Module Function**:

Implements hybrid encryption using **RSA and AES**, building a secure and reliable end-to-end encryption mechanism.

Ensures **confidentiality**, **integrity**, and **verifiability** of user messages during QQ-based chatbot communication.



🔧 Technical Implementation & Key Tasks:

✅ **RSA Key Pair Generation and Decryption (Asymmetric Encryption)**

•Uses 1024 or 2048-bit primes to generate RSA key pairs

•Public key encrypts AES keys and is transmitted over the network

•Private key decrypts the received AES keys and is stored only on the user's device

✅ **AES Key Generation & Message Encryption (Symmetric Encryption)**

•Uses 128/192/256-bit AES keys to encrypt user input

•Supports **AES-GCM** or **AES-CBC** modes for both confidentiality and data integrity

•A unique AES session key is generated for each round of conversation to enhance security

✅ **Base64 Encoding for Text Transmission**

•Encrypted results are binary, so **Base64 encoding** is required for QQ transmission

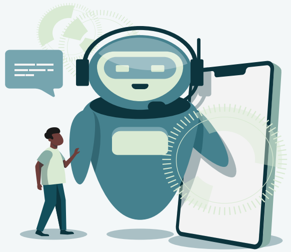
•The decoding process is automatically handled at both chatbot and user front-end

✅ **Secure Padding & Block Mode Design**

•Implements **PKCS#7 padding** for uniform block alignment

•Supports both **CBC mode** and **GCM mode**, chosen based on usage scenarios

Communication Transmission Module



🧩 **Module Function**:

This module serves as the **transmission layer** of the entire system, responsible for message routing and communication control between the front-end user and the chatbot backend. It uses the QQ platform for message sending and receiving, combined with event listening and API invocation provided by the Napcat framework to ensure secure and reliable message delivery between client and bot.

🔧 Implementation Mechanisms:

✅ **Monitoring QQ Message Events**

•Continuously listens to private and group messages via Napcat's @bot.private_event and @bot.group_event interfaces

•Supports command recognition and parsing (e.g., "enable encryption mode on:")

✅ **Automatic Switching Between Plaintext and Ciphertext**

•Determines mode based on whether an AES key is present

•In encrypted mode, the decryption module is triggered; in plaintext mode, messages are forwarded directly

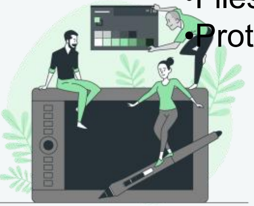
•Enables dynamic encryption control logic (on/off)

✅ **Multi-Type Message Relay**

•Text: Supports plaintext & Base64-encoded ciphertext. Encrypted content: Wrapped in JSON for secure transmission.

•Files: Encrypted files sent via structured JSON or file APIs.

•Protocol: Follows JSON schema like:{ "type": "secure_msg", "enc": "...", "aes": "..." }



未来技术学院
FUTURE TECH

03

Testing and Demonstration

Perform local and remote communication tests to verify the stability and usability of the system in encrypted transmission and interface interaction

This project designs a system that integrates a chatbot with an end-to-end encrypted communication mechanism.
Users can send messages to an AI chatbot deployed on the QQ platform through a front-end interface.

✅ Design Overview

Case Analysis: Current System Implementation Status

Supports RSA key generation, public key exchange, and AES key generation with encrypted transmission.

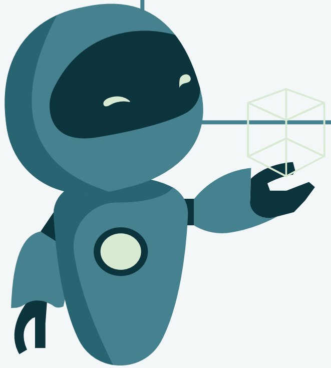
Multi-layer encrypted communication tool

Implements AES-GCM mode to encrypt user messages; the chatbot can detect and automatically decrypt them.

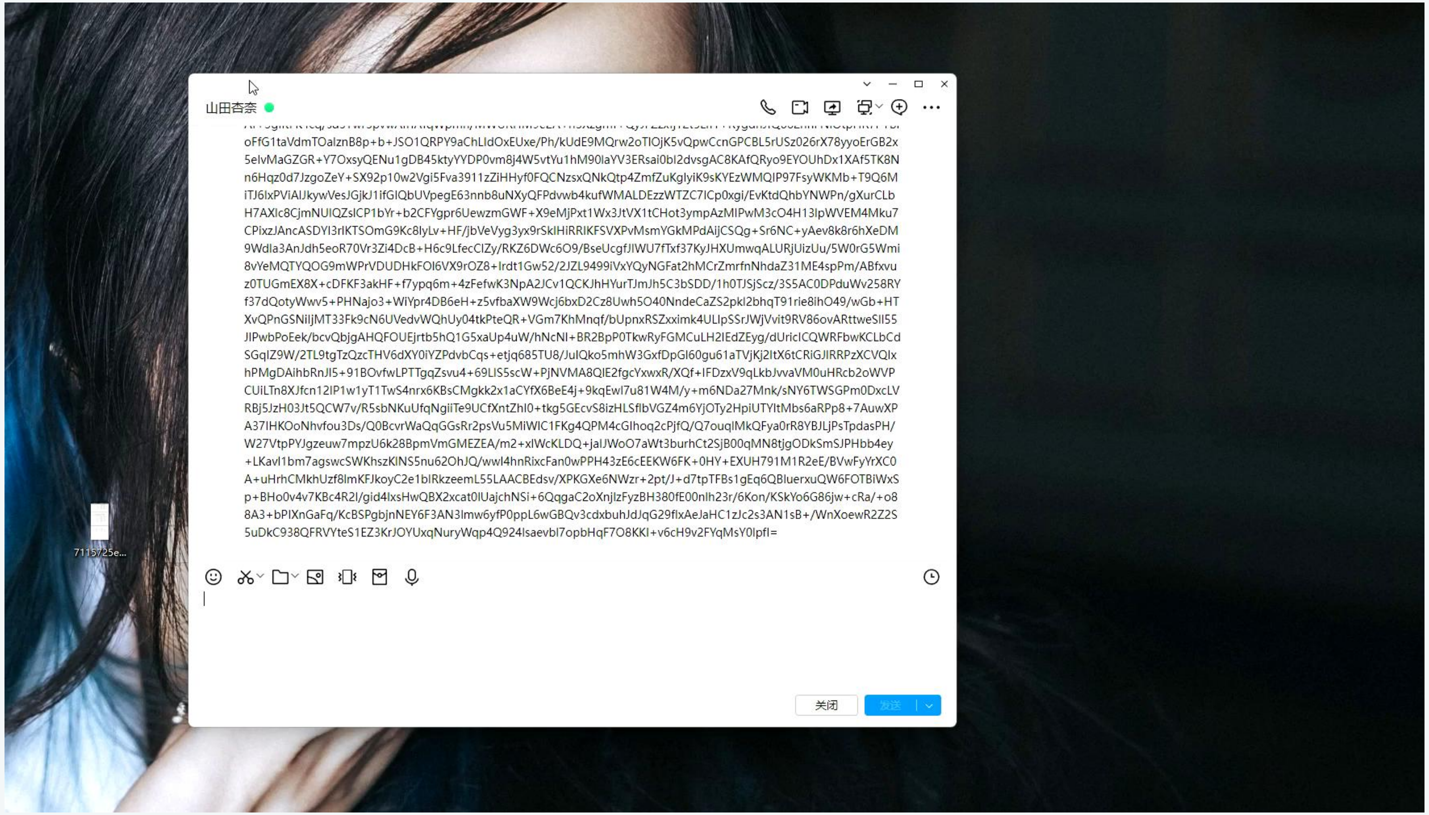
Intelligent encrypted interaction

Napcat successfully invokes the DeepSeek API to generate chatbot responses.

Smart QQ chatbot service



未来技术学院
FUTURE TECH



oFfG1taVdmTOalznB8p+b+JSO1QRPY9aChLIdOxEUxe/Ph/kUdE9MQrw2oTIOjK5vQpwCcnGPCBL5rUSz026rX78yyoErGB2x
5elvMaGZGR+Y7OxsyQENu1gDB45ktyYYDP0vm8j4W5vtYu1hM90laYV3ERsai0bl2dvsgAC8KAfQRyo9EYOUhDx1XAf5TK8N
n6Hqz0d7JzgoZeY+SX92p10w2Vgi5Fva3911zZiHHyf0FQCNzsxQNkQtp4ZmfZuKglyiK9sKYEzWMQIP97FsyWKMb+T9Q6M
iTJ6IxPViAIJkywVesJGjkJ1ifGIQbUVpegE63nnb8uNXyQFPdvwb4kufWMALDEzzWTZC7ICp0xgi/EvKtdQhbYNWPn/gXurCLb
H7AXIc8CjmNUIQZsICP1bYr+b2CFYgpr6UewzmGWF+X9eMjPxt1Wx3JtVX1tCHot3ympAzMIPwM3cO4H13IpWVEM4Mku7
CPixzJAncASDYI3rIKTSOmG9Kc8lyLv+HF/jbVeVyg3yx9rSkIHiRRIKFSVXPvMsmYGkMPdAijCSQg+Sr6NC+yAev8k8r6hXeDM
9Wdla3AnJdh5eoR70Vr3Zi4DcB+H6c9LfecClZy/RKZ6DWc6O9/BseUcgfJIWU7fTxf37KyJHXUmwqALURjUizUu/5W0rG5Wmi
8vYeMQTYQOG9mWPrVDUDHkFOI6VX9rOZ8+Irdt1Gw52/2JZL9499iVxYQyNGFat2hMCrZmrfnNhdaZ31ME4spPm/ABfxvu
z0TUGmEX8X+cDFKF3akHF+f7ypq6m+4zFefwK3NpA2JCv1QCKJhHYurTJmJh5C3bSDD/1h0TJSjScz/3S5AC0DPduWv258RY
f37dQotyWwv5+PHNajo3+WiYpr4DB6eH+z5vfbaXW9Wcj6bxD2Cz8Uwh5O40NndeCaZS2pkI2bhqT91rie8ihO49/wGb+HT
XvQPnGSNiIjMT33Fk9cN6UVedvWQhUy04tkPteQR+VGm7KhMnqf/bUpnxRSZxximk4ULIpSSrJWjVvit9RV86ovARttweSII55
JIPwbPoEek/bcvQbjgAHQFOUEjrtb5hQ1G5xaUp4uW/hNcNI+BR2BpP0TkwRyFGMCuLH2IEdZEyg/dUricICQWRFbwKCLbCd
SGqIZ9W/2TL9tgTzQzcTHV6dXY0iYZPdvbCqs+etjq685TU8/JuIQko5mhW3GxfDpGI60gu61aTVjKj2ItX6tCRiGJIRRPzXCVQIx
hPMgDAihbRnJI5+91BOvfwLPTTgqZsvu4+69LIS5scW+PjNVMA8QIE2fgcYxwxR/XQf+IFDzxV9qLkbJvvaVM0uHRcb2oWVP
CUiLTn8XJfcn12IP1w1yT1TwS4nrx6KBsCMgkk2x1aCYfX6BeE4j+9kqEwI7u81W4M/y+m6NDa27Mnk/sNY6TWSGPm0DxcLV
RBj5JzH03Jt5QCW7v/R5sbNKuUfqNgiiTe9UCfXntZhI0+tkg5GEcvS8izHLSfIbVGZ4m6YjOTy2HpiUTYItMbs6aRPp8+7AuwXP
A37IHKOoNhvfou3Ds/Q0BcvrWaQqGGsRr2psVu5MiWIC1FKg4QPM4cGIhoq2cPjfQ/Q7ouqIMkQFya0rR8YBJLjPsTpdasPH/
W27VtpPYJgzeuw7mpzU6k28BpmVmGMEZEA/m2+xIWcKLDQ+jaIJWoO7aWt3burhCt2SjB00qMN8tjgODkSmSJPHbb4ey
+LKavl1bm7agswcSWKhszKINS5nu62OhJQ/wwI4hnRixcFan0wPPH43zE6cEEKW6FK+0HY+EXUH791M1R2eE/BVwFyYrXC0
A+uHrhCMkhUzf8ImKFJkoyC2e1bIRkzeemL55LAACBEdsv/XPKGXe6NWzr+2pt/J+d7tpTFBs1gEq6QBIuerxuQW6FOTBiWxS
p+BHo0v4v7KBc4R2I/gid4IxsHwQBX2xcat0IUajchNSi+6QqgaC2oXnjIzFyzBH380fE00nIh23r/6Kon/KSkYo6G86jw+cRa/+o8
8A3+bPIXnGaFq/KcBSPgbjnNEY6F3AN3Imw6yfP0ppL6wGBQv3cdxbuhJdJqG29fIxAeJaHC1zJc2s3AN1sB+/WnXoewR2Z2S
5uDkC938QFRVYteS1EZ3KrJOYUxqNuryWqp4Q924Isaevbl7opbHqF7O8KKI+v6cH9v2FYqMsY0IpfI=

Future Prospects

This project is centered on the concept of "end-to-end encryption + AI chatbot." We have preliminarily implemented a secure communication mechanism based on RSA and AES, along with a basic framework for automated key exchange and encrypted message interaction. Although the current system can complete parts of the encryption and decryption communication process, we are clearly aware that there is still significant room for improvement in terms of functional completeness, user experience, and overall security robustness.

目录

01 项目背景与目标

构建一个图形化加密通信

02 拟解决的问题与工

本项目解决通信过程缺乏安全机制的问题，集

03 测试与演示

通过本地与远程通信测试，验证系统在加密传

04 未来展望

后续将拓展为支持多用户、实时通信与更高安

未来技术学院 FUTURE TECH

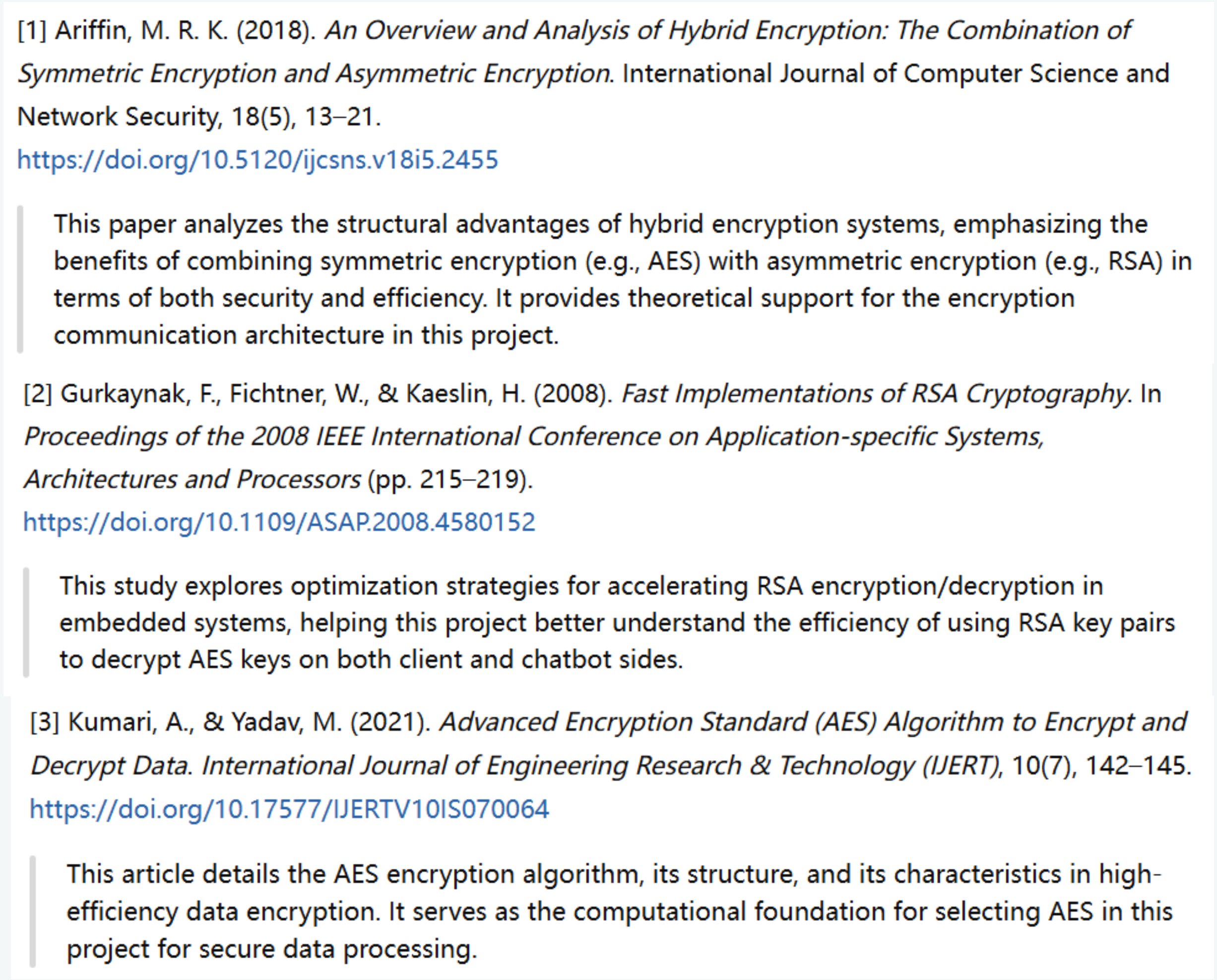
基于RSA与AES加密过程的AI聊天机器人

杨腾越	202364820861
罗景楠	202364870981
张越	202364820921
陈思涵	202330420212

🔧 Suggestions for Improvement

Automated AES Key Reception and Handling	The front-end GUI lacks a module for automatically processing AES keys sent by the chatbot. Suggest adding a listener mechanism to auto-call receive_and_decrypt_aes_key() for decryption and display.
Key Caching and User Identity Binding	Add a user_id -> aes_key mapping for key caching (you've begun session_keys), ensuring per-user session key storage and retrieval.
Improved Encrypted Communication Protocol	Suggest adding "iv" (for CBC) or "mode": "GCM" fields to the current secure_msg structure to indicate the decryption method and prevent failure.
Error Feedback and UI Visualization	Add encryption status, success/failure prompts, and debugging mode to help users understand decryption failures and key validity issues.
Enhanced Message Security Mechanisms	In future, add message signing to prevent tampering (e.g., HMAC or RSA signatures) and improve overall security level.

Reference



[1] Ariffin, M. R. K. (2018). *An Overview and Analysis of Hybrid Encryption: The Combination of Symmetric Encryption and Asymmetric Encryption*. International Journal of Computer Science and Network Security, 18(5), 13–21. https://doi.org/10.5120/ijcsns.v18i5.2455

> This paper analyzes the structural advantages of hybrid encryption systems, emphasizing the benefits of combining symmetric encryption (e.g., AES) with asymmetric encryption (e.g., RSA) in terms of both security and efficiency. It provides theoretical support for the encryption communication architecture in this project.

[2] Gurkaynak, F., Fichtner, W., & Kaeslin, H. (2008). *Fast Implementations of RSA Cryptography*. In *Proceedings of the 2008 IEEE International Conference on Application-specific Systems, Architectures and Processors* (pp. 215–219). https://doi.org/10.1109/ASAP.2008.4580152

> This study explores optimization strategies for accelerating RSA encryption/decryption in embedded systems, helping this project better understand the efficiency of using RSA key pairs to decrypt AES keys on both client and chatbot sides.

[3] Kumari, A., & Yadav, M. (2021). *Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data. International Journal of Engineering Research & Technology (IJERT)*, 10(7), 142–145. https://doi.org/10.17577/IJERTV10IS070064

> This article details the AES encryption algorithm, its structure, and its characteristics in high-efficiency data encryption. It serves as the computational foundation for selecting AES in this project for secure data processing.