



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

目录

01

项目背景与目标

构建一个图形化加密通信工具，实现 RSA/AES 加解密与 QQ 机器人通信联动。

02

拟解决的问题与工作内容说明

本项目解决通信过程缺乏安全机制的问题，集成消息加密图像加密和机器人问答。

03

测试与演示

通过本地与远程通信测试，验证系统在加密传输与界面交互中的稳定性与可用性。

04

未来展望

后续将拓展为支持多用户、实时通信与更高安全性的端到端加密平台。

01

项目背景与目标

You can enter a subtitle here if you need it

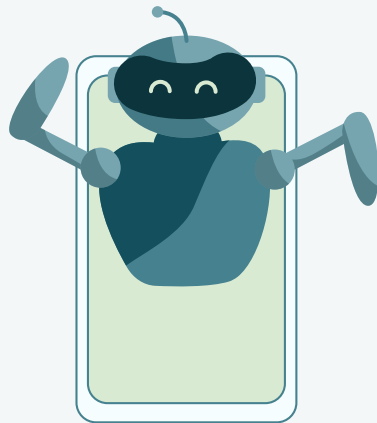
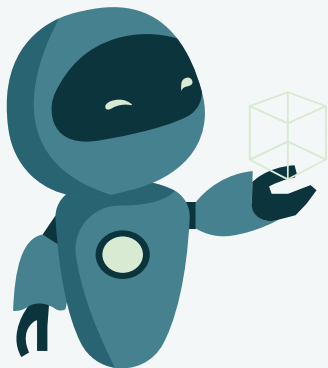


项目背景

在当前信息社会中，个人隐私和数据安全已成为网络通信中最为关键的问题之一。

随着人工智能聊天机器人的广泛应用，越来越多的用户在日常生活、工作和学习中通过聊天机器人获取服务或交流信息。

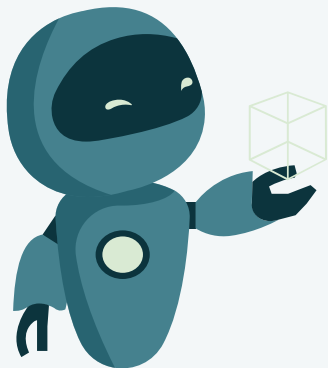
然而，传统的即时通讯平台（如QQ）在默认状态下并未为用户与聊天机器人之间的通信内容提供端到端加密保障，这使得用户的私密对话可能在传输过程中面临被拦截、监听或泄露的风险。



目标

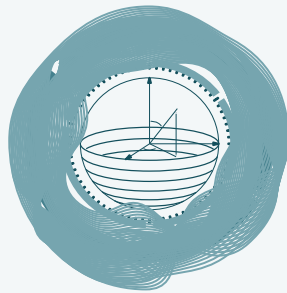
本项目的目标是构建一个融合“**AI聊天机器人**”与“**端到端加密通信机制**”的系统，使用户可以通过前端图形化界面与部署在QQ上的聊天机器人进行安全对话。

在该系统中，用户只需输入明文消息，系统会自动完成**AES对称加密 + RSA非对称加密**流程，并将加密后的密文发送至QQ平台。而机器人在接收到密文后，也能完成自动解密、生成回复，并重新加密返回给用户，确保整个通信链路上的信息内容均为密文，从而有效防止数据泄露与中间人攻击。



喵~ 抱抱你! (っ´▽`´)っ

我最近好辛苦，你可以安慰我一下吗



02

拟解决的问题与 工作内容说明

You can enter a subtitle here if you need it



拟解决的问题

本项目旨在解决用户在通过QQ聊天机器人进行交流时缺乏**隐私保护与消息加密机制**的问题。在当前实际应用中，用户与AI机器人的通信内容大多以明文形式通过QQ平台传输，这使得信息极易在客户端、服务器或网络传输过程中被恶意第三方窃听或篡改。

为此，我们设计并实现了一个**端到端加密通信系统**，该系统将加密与解密过程完全封装于用户界面与机器人逻辑中，用户无需掌握加密技术即可自动完成消息加解密，从而确保通信过程的机密性、完整性与抗监听性。

本项目主要解决以下几个问题：对应着具体的开发与技术任务：



构建完整的加密通信逻辑流程
(密钥协商 + 加解密机制)



实现前端用户界面与
自动加解密功能的集成



构建机器人端加密逻辑
与DeepSeek交互逻辑



实现QQ平台密文消息
的传输与解析支持



支持多轮加密对话
与新会话自动重新协商密钥

任务模块

本项目主要由以下几个模块组成，每个模块对应着具体的开发与技术任务：



前端用户界面模块

提供图形化操作界面，让用户能输入明文、查看回复，并自动执行加解密操作。



加密算法模块

实现 RSA 与 AES 混合加密逻辑，确保通信过程中信息机密性。



机器人服务器端模块

中间件接收密文消息、解密处理、生成AI回复，并加密后发回。



通信传输模块

作为中介传输层，实现机器人与用户之间的消息中转



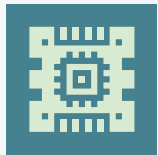
前端用户界面模块 与 机器人服务器端模块

- 功能：提供图形化操作界面，让用户能输入明文、查看回复，并自动执行加解密操作。
- 工作流：所有消息传输过程均采用**RSA + AES 混合加密**方案，保障通信隐私性。
整体架构遵循“密钥协商 + 会话加密”的安全设计思路，通信过程主要分为两个阶段：

1.密钥协商阶段 (Session Setup)

2.消息交互阶段 (Secure Chat)

用户通过界面启动加密通信，
前端自动生成 RSA 密钥对，
并将公钥发送给机器人。



用户输入明文问题，
前端使用当前 AES 密钥
进行 GCM 加密后通过 QQ 发送。



机器人生成临时 AES 密钥，
并使用用户公钥加密后发回用户。
用户使用私钥解密得到 AES 会话密钥，
从此用于加密对话内容。




机器人端接收密文并解密调用 API 生成回答。
回答内容重新使用新一轮 AES 密钥加密，
并使用用户 RSA 公钥加密该密钥，一并返回。
用户解密获取新 AES 密钥后，继续安全会话。









加密算法模块（加密工具层）

 **模块功能：**实现 RSA 与 AES 的混合加密逻辑，构建安全可靠的端到端加密通信机制，确保用户在 QQ 聊天过程中的信息**机密性、完整性与可验证性**。

工作内容与技术要点：

-  **RSA 密钥对生成与加解密（非对称加密）**
 - 使用 1024 位或 2048 位大素数生成公私钥
 - 公钥用于加密 AES 密钥并在公开网络中传输
 - 私钥用于解密收到的 AES 密钥，仅用户本地持有
-  **AES 密钥生成与消息加解密（对称加密）**
 - 使用 128/192/256 位 AES 密钥对用户输入内容加密
 - 采用 AES-GCM 或 AES-CBC 模式支持保密性与数据完整性
 - 每轮对话生成独立 AES 会话密钥，提升抗攻击能力
-  **Base64 编码支持文本传输**
 - 加密结果为二进制数据，需 Base64 编码后通过 QQ 发送
 - 解码过程在机器人与用户前端自动完成
-  **安全填充与分组模式设计**
 - 使用 **PKCS#7 填充机制** 补齐加密分组
 - 支持 **CBC 模式** 与 **GCM 模式**，根据用途选取
 - 每条消息附带随机 IV（初始化向量），防止密文重放攻击





通信传输模块（QQ 平台 + Napcat 框架）

模块功能：该模块是整个系统的“中介传输层”，负责用户前端与机器人后端之间的消息中转与通信控制。它基于 QQ 平台实现消息发送与接收，结合 Napcat 框架提供的事件监听与 API 调用，保障消息在客户端与机器人之间可靠、安全地传递。

工作内容与实现机制：

- ✓ **监听 QQ 消息事件**
 - 使用 Napcat 提供的 @bot.private_event 和 @bot.group_event 接口持续监听私聊和群聊消息
 - 实现对用户命令（如“加密通信模式on:”）的识别与解析
- ✓ **明文/密文通信自动切换**
 - 根据是否包含 AES 密钥判断通信模式是否开启
 - 在密文模式下自动调用解密模块解码内容；明文模式则直接处理 提供启用/关闭加密通信的动态控制逻辑
- ✓ **支持多种消息类型的中转**
 - 文本消息：支持明文对话、Base64 编码密文的传输 文件消息（如图片）：通过接口或文件结构传递
 - 密钥交换消息：将加密后的 AES 密钥通过 JSON 格式封装后传递结构化
 - 通信：使用 JSON 定义通信协议（如 {"type": "secure_msg", "enc": ..., "aes": ...}）



03

测试与演示

You can enter a subtitle here if you need it





本项目设计了一个融合聊天机器人与端到端加密通信机制的系统，用户通过前端界面向部署在 **QQ** 平台上的 **AI** 聊天机器人发送信息

✓ 设计方案概述

案例分析：当前系统实现状态

支持 **RSA** 密钥生成、公钥交换、**AES** 密钥生成与加密传输。

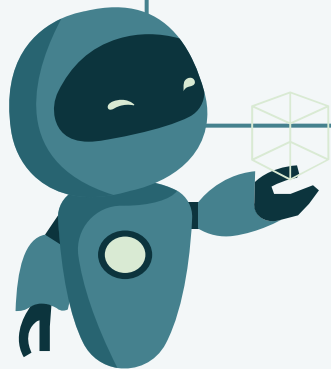
多层次信息加密工具

实现 **AES-GCM** 模式加密用户消息，机器人端能识别并自动解密

智能加密沟通

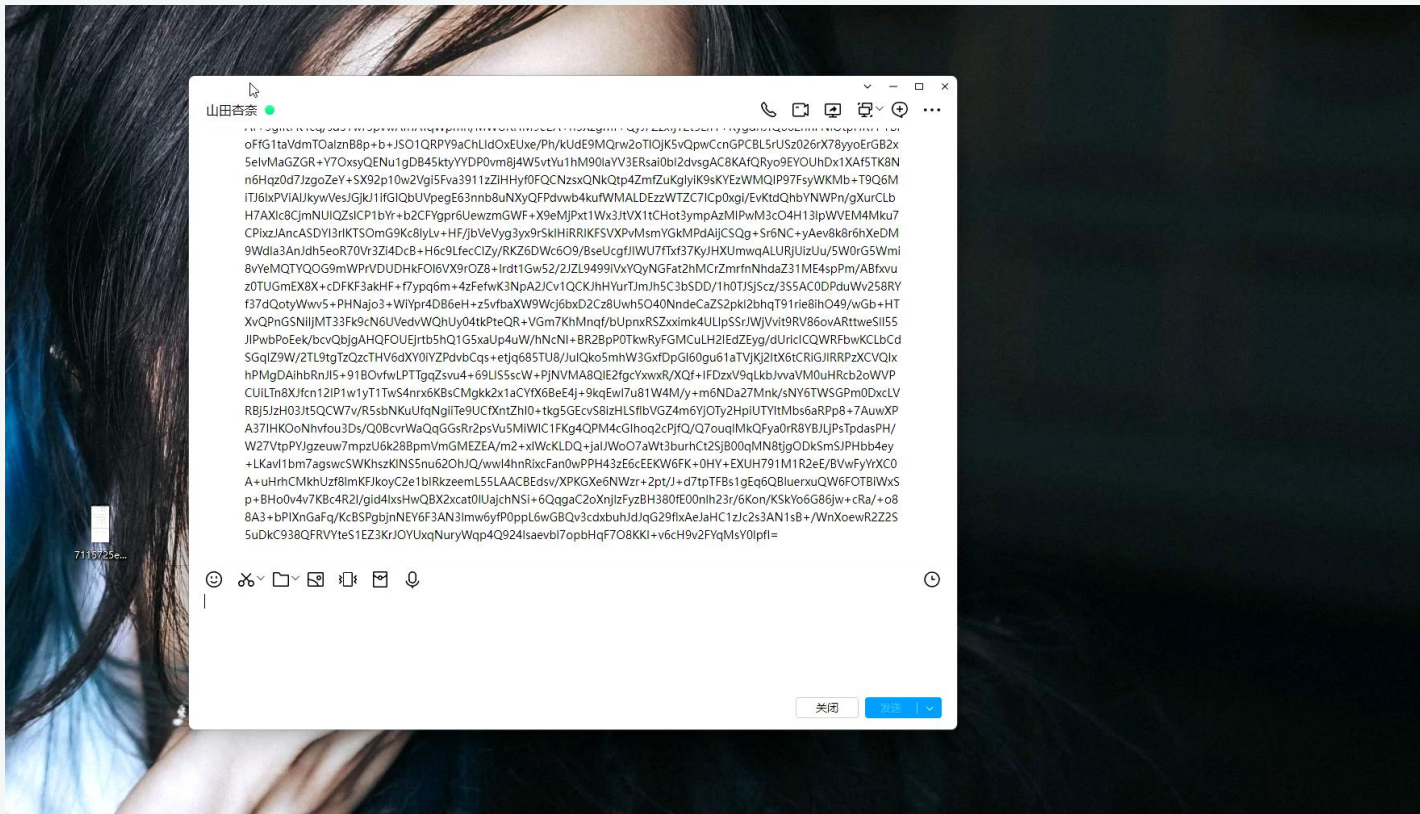
Napcat 能正确调用 **DeepSeek API** 生成对话内容。

智能QQ聊天机器人服务





未来技术学院
FUTURE TECH



🌟 项目未来展望

本项目以“端到端加密 + AI 聊天机器人”为核心，初步实现了基于 RSA 和 AES 的安全通信机制，以及自动化密钥协商与密文交互的初级框架。尽管目前系统已可完成部分加解密通信流程，但我们也清楚地意识到，在功能完备性、用户体验、安全鲁棒性等方面仍有较大提升空间。

目录

01 项目背景与目标

构建一个图形化加密通信

02 拟解决的问题与目标

本项目解决通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



改进建议

<u>自动化 AES 密钥接收与处理</u>	前端 GUI 中缺乏对机器人发来加密 AES 密钥的自动处理模块，建议加入监听机制，自动调用 <code>receive_and_decrypt_aes_key()</code> 完成解密与展示。
<u>密钥缓存与用户身份绑定</u>	增加 <code>user_id -> aes_key</code> 映射缓存（你已初步实现 <code>session_keys</code> ），确保每位用户的会话密钥独立存储与调用
<u>完善密文通信协议</u>	当前 <code>secure_msg</code> 结构中建议加入字段 <code>"iv"</code> （适用于 CBC），或 <code>"mode": "GCM"</code> 指示解密方式，避免解密失败。
<u>错误反馈与 UI 可视化</u>	为前端添加密钥状态、解密成功与否提示，提升用户体验。增设调试模式，可记录加密失败原因、密钥失效等信息。
<u>增强消息安全机制</u>	未来可集成消息签名、防篡改校验（如 HMAC 或 RSA 签名），提升安全等级。

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>

本文系统分析了混合加密体系的结构与优势，特别强调了将对称加密（如 AES）与非对称加密（如 RSA）结合的安全性和效率优势，为本项目设计加密通信架构提供理论支持。

[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>

该论文探讨了在嵌入式系统中快速实现 RSA 加密算法的优化策略，帮助本项目理解在客户端和机器人端使用 RSA 公私钥加解密 AES 密钥时的性能权衡问题。

[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>

本文详细介绍了 AES 算法的加解密流程、块加密结构及其适用于高效率数据加密的特性，为本项目选择 AES 用于对话数据加密提供算法依据。