Encrypted Chat Box - README

Encrypted Chat Box

Overview

This is a simple **Encrypted Chat Box** built in Python using:

- **Sockets** for network communication

- **RSA encryption** for secure message exchange

- **Threading** for simultaneous sending and receiving

The application allows two users to connect over a local network (LAN) and exchange messages securely using public-private key encryption.

Features

- **RSA Key Pair Generation**: Automatically generates a 1024-bit key pair for each user.

- **Secure Key Exchange**: Public keys are exchanged between participants for encryption.

- **End-to-End Encryption**: Messages are encrypted before sending and decrypted after receiving.

- **Real-time Chat**: Sending and receiving messages occur simultaneously.

How It Works

1. When starting, the user can choose to:

   - **Host (1)**: Start the server and wait for a connection.

   - **Connect (2)**: Connect to the server.

2. RSA keys are generated for both parties.

3. Public keys are exchanged between users.

4. Messages are encrypted with the recipient's public key and decrypted with the private key.

5. The chat runs in two threads:

   - **Sending Thread**: Reads input from the user and sends the encrypted message.

   - **Receiving Thread**: Receives encrypted data and decrypts it.

Requirements

- Python 3.x

- `rsa` library

- `socket` (built-in)

- `threading` (built-in)

Install dependencies:

```bash
pip install rsa
```

Usage

1. **Run the script** on both devices:

```bash
```

python encrypted_chat.py

```

2. **Choose role**:

   - Enter `1` to host.

   - Enter `2` to connect.

3. **Start chatting** securely!

---

## Security Notes

- RSA 1024-bit encryption ensures secure communication.

- This example is for educational purposes; for production, use stronger keys (2048-bit or higher) and proper authentication.

- Works best over local network (LAN).

---

## License

This project is released under the MIT License.
```