PROGRESS THROUGH KNOWLEDGE

Secure Chat API: Encrypted Chat Application Program Interface with Block-chain Authentication

BATCH MEMBER

G. GOKUL	(812420104027)
M. PREMSRIDEV	(812420104071)
A. RAHMATHULLAH	(812420104073)
T. SHAIK MOHAMED FAHAD	(812420104083)

GUIDE NAME

Mr. P. MANIKANDAN M.E

HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering

DOMAIN

- **Block-Chain Technology**

Block-Chain technology is a decentralized and distributed ledger system that enables secure and transparent record-keeping of transactions across a network of computers. It gained prominence as the underlying technology for cryptocurrencies like Bitcoin, but its applications extend far beyond digital currencies. Block-Chain has the potential to revolutionize various industries by providing a tamper-resistant and efficient way to manage and verify transactions.

INTRODUCTION

In an era where digital communication plays a pivotal role in our daily lives, ensuring the confidentiality and integrity of our conversations is paramount. Introducing the "Secure Chat API," a revolutionary development in the realm of encrypted communication. This cutting-edge Chat Application Program Interface (API) not only prioritizes user privacy through robust encryption but also leverages the power of blockchain authentication for an unprecedented level of security.

OBJECTIVE

- ➢ To develop a highly secure and reliable Chat API, designed to facilitate encrypted communication within a Chat Application.
- ➢ To implement robust encryption protocols among the various platforms without implementing the actual cryptographic technique.
- ➢ To integrate Blockchain Authentication mechanisms to enhance user identity verification and authentication, thereby fortifying the overall security posture of the Chat API.

LITERATURE SURVEY

Serial No:	Author name, Title and date	Concept	Technique (Algorithm)	Advantages	Disadvantages
1	Mirza K. B. Shuhan1, Tariqul Islam. A Secure and Decentralized Block-chain Based Messaging Network 5th Aug 2023	A Proof of Concept (PoC) of the Quarks system leveraging Distributed Ledger Technology (DLT), and conducted load testing on that.	Smart Contract	Decentralized Block-chain Based Messaging Network	Platform Dependent, Need to create a Network to make Communication
2	Samira Prabhune, Sonal Sharma. End-to-End Encryption for Chat App with Dynamic Encryption Key 09 March 2022	how dissimilar concerns in an app can add up to the probable for severe vectors adjacent to end-to-end solitude despite their creature several layers of security.	To overcome the app is anticipated in the chats are secured by the dynamic key encryption by **md5** algorithm	Dynamic Key Encryption	Need to save Dynamic Keys

LITERATURE SURVEY

Serial No:	Author name, Title and date	Concept	Technique (Algorithm)	Advantages	Disadvantages
3	Soman Nayak, Surajit Dass An application for end to end secure messaging service on Android supported device 03-05 October 2017	Proposing a secure messaging protocol that utilizes SHA-2 hash generation for key generation and AES-256 encryption for securing messages during transmission.	SHA-2, AES-256 encryption	Two step Verification	Complicate Process, Cannot Implement in Multiple Platform
4	Noor Sabah, Jamal M. Kadhim and Ban N. Dhannoon Developing an End-to-End Secure Chat Application November 2017	The proposed architecture is designed to be Client-Server chat application. In client side, when a user sets up the application, the user either selects registration or log-in.	XSalsa20	Firebase Cloud Messaging	Static Key encryption, Platform Dependent

EXISTING SYSTEM

❖ Basic Encrypted Chat:

Typically, existing chat applications use encryption protocols (e.g., TLS/SSL) to secure data in transit. However, end-to-end encryption may not be universal, leaving potential vulnerabilities.

❖ User Authentication:

Commonly, user authentication relies on usernames and passwords.

Centralized authentication systems may pose risks if compromised.

❖ Platform Dependent:

Most chat applications are created based on a particular platform.

This introduces a single point of failure and cannot integrate the other platforms.

PROPOSED SYSTEM

- ❖ End-to-End Encryption:

  Implements robust end-to-end encryption algorithms for secure communication.

  Ensures that only the intended recipients can decrypt and access messages.

- ❖ Implementation of several platforms:

  To creating an integrated application programming interface (API) that can be used with a variety of platforms, including mobile, software, and Web applications

- ❖ Decentralized Message Integrity:

  Uses block-chain as a distributed ledger for storing message metadata.

  Ensures the integrity and tamper-resistance of messages with timestamping on the block-chain.

METHODOLOGY

The project methodology encompassed several key phases to ensure the successful development of the Chat API. We began with extensive research and analysis, delving into encryption protocols and Block-chain Authentication methods to determine the most suitable options for integration. Following this, we designed the architecture of the Chat API, focusing on seamless integration of encryption and Block-chain Authentication modules while prioritizing scalability and cross-platform compatibility. The development phase involved the execution of our architectural design, implementing encryption protocols and Block-chain Authentication mechanisms into the Chat API. Rigorous testing and validation procedures were then undertaken to verify functionality and security, ensuring compliance with specified requirements.

KEY FEATURES

- End-to-End Encryption: The Chat API ensures that all communication within the Chat Application is encrypted using advanced encryption protocols, guaranteeing privacy and confidentiality for users.
- Seamless Cross-Platform Compatibility: The developed Chat API is designed to work seamlessly across different devices and platforms, providing a consistent user experience regardless of the device being used.
- Robust Security Measures: The project prioritizes security, implementing robust encryption protocols and Block-chain Authentication mechanisms to safeguard user data and privacy against potential threats.

ENCRYPTION PROTOCOLS

Fernet is a symmetric encryption algorithm and protocol for securing data. It was developed as part of the cryptography library in Python. Fernet provides a straightforward and secure way to encrypt and decrypt data, making it suitable for various applications where data confidentiality is essential.

```
Key = base64.b64encode((timestamp+user_identity).encode('utf-8'))
```

Fernet required Key to encrypt data. Create a dynamic key type that does not need to store or send along with the encrypted data. Not like asymmetric encryption methods, it does not require memorizing either a private or public key.

BLOCK-CHAIN AUTHENTICATION

The messages are encrypted with a series of dynamic keys that are generated by block-chain technology (base64). The float data timestamp and string data user identity are combined and converted to a base64 value, which is called a hash value in block chains.

The hash value of the non-encrypted data is generated and sent along with the encrypted message. The hash value is used to ensure the decrypt function confirms the decrypted value is correct.

More specifically, block chains are used to ensure the correct user is connected to the server.

IMPLEMENTATION

During the implementation phase, we translated design specifications into a fully functional Chat API with integrated encryption and Block-chain Authentication. Following best practices, we executed the architectural design, implementing Fernet for communication encryption. Block-chain Authentication was seamlessly integrated, utilizing smart contracts for automated user verification. Emphasis was placed on code quality and efficiency, ensuring maintainability. Through iterative development and rigorous testing, we optimized the Chat API to meet project objectives. The implementation phase showcased meticulous attention to detail, resulting in a secure and reliable communication platform.

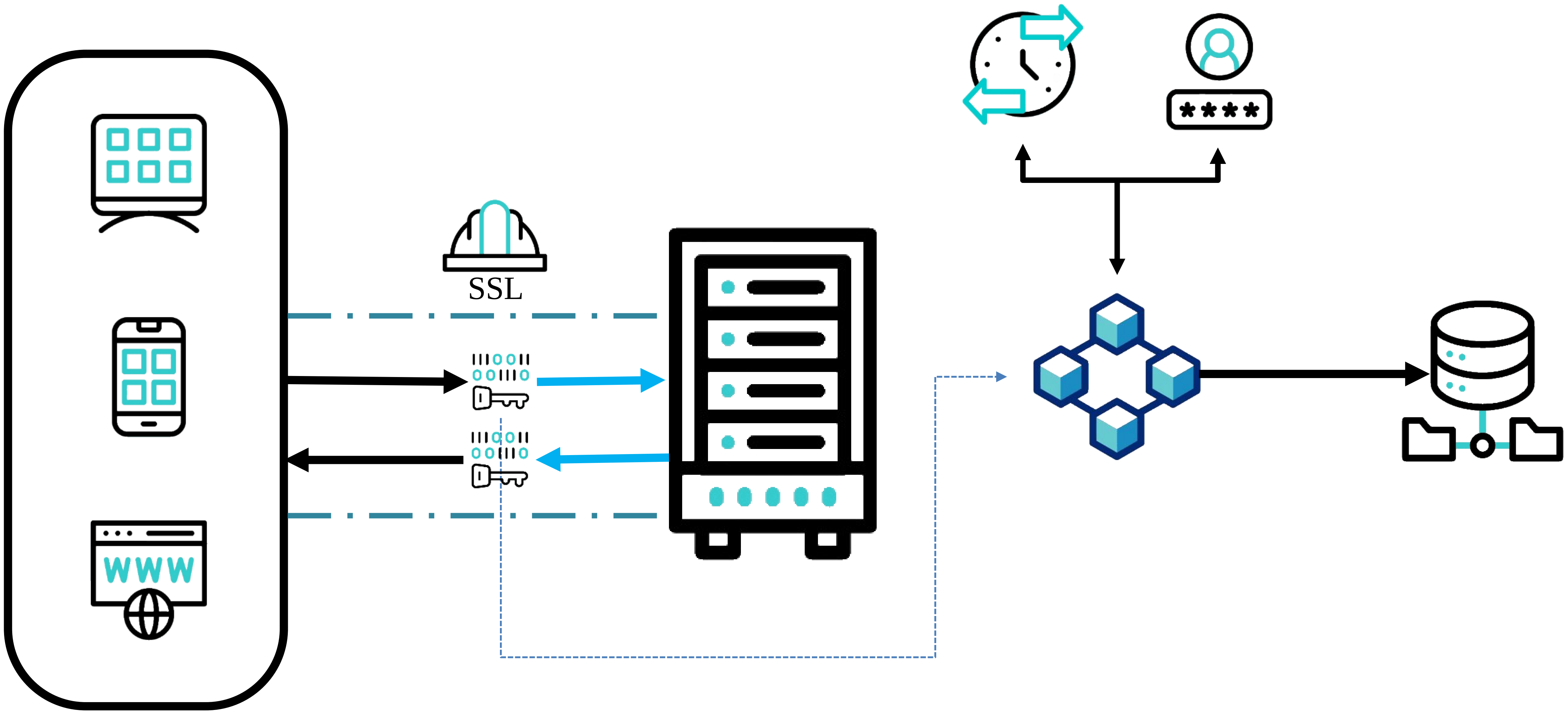
MODULES

The system is completely deployed server side as backend.

Demo Application:

- Login Page
- Chatting Page

ACTIVITY DIAGRAM



SOFTWARE AND HARDWARE REQUIREMENTS

FRONT END

- Python (Toolkit Interface - tkinter)

BACK END

- Python (threading, socket, cryptography.fernet, hashlib, base64)

HARDWARE

- Basic requirements for a system

SPECIFICATION

- Minimum 4 GB ram
- Window(8,10 and 11 versions), Linux, Ubuntu

TESTING AND VALIDATION

The testing and validation phase of the project played a crucial role in ensuring the functionality, security, and reliability of the developed Chat API. This phase encompassed various testing methodologies and validation procedures to assess the performance and effectiveness of the system.

- ❖ Functional Testing
- ❖ Security Testing
- ❖ Performance Testing
- ❖ User Acceptance Testing
- ❖ Validation

CONCLUSION

In conclusion, the development of the secure Chat API has been a significant achievement, providing a robust platform for encrypted communication while integrating Block-chain Authentication for enhanced security. Throughout the project, we have achieved several key milestones and addressed various challenges, resulting in a successful outcome. Overall, the secure Chat API project represents a significant step forward in the development of secure communication platforms, offering users a reliable and privacy-focused solution for encrypted communication.

FEATURE ENHANCEMENT

- **Multimedia Messaging:** Enable users to exchange multimedia content such as images, videos, and audio files securely within the Chat API.
- **Voice and Video Calling:** Integrate voice and video calling functionalities into the Chat API, enabling users to initiate secure audio and video calls with their contacts.
- **Advanced Authentication Mechanisms:** Enhance user authentication by introducing advanced biometric authentication methods such as fingerprint recognition or facial recognition.

QUESTIONS

THANK YOU