TEXT-IT: A Secure Web Chat Application

Paras Kumar
(4th Year Student, CSIT)
KIET Group of Institutions,
Delhi - NCR, Ghaziabad
paras.2024csit1086@kiet.edu

Prabhat Saini
(4th Year Student, CSIT)
KIET Group of Institutions,
Delhi - NCR, Ghaziabad
prabhat.2024csit1095@kiet.edu

Ms. Garima Singh
(Assistant Professor, CSIT)
KIET Group of Institutions,
Delhi - NCR, Ghaziabad
garima.singh@kiet.edu

2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-7024-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCNT61001.2024.10724095

Abstract —— This paper introduces a secure web chat application that leverages steganography, a technique for concealing messages within images, to ensure secure and covert communication. The application integrates the MERN stack (MongoDB, Express.js, React.js, and Node.js) for robust development and employs the Least Significant Bit (LSB) algorithm for steganography. The paper delves into the intricate details of the application's architecture, encompassing modular design, user authentication, real-time chat functionality, steganography integration, user interface, back-end server operations, database management, and security considerations. Additionally, it explores potential future enhancements, including video chat, group chat, and more sophisticated steganographic methods. This research offers a comprehensive overview of a web chat application that prioritizes user privacy and data security through steganographic integration.

Keywords— Steganography, LSB, MERN, Socket.io, Cryptography

I. INTRODUCTION

The contemporary digital landscape necessitates robust security measures for safeguarding user privacy and data integrity. In this vein, we propose a novel web chat application that integrates steganography, a sophisticated technique for concealing covert messages within seemingly innocuous images. This unique solution transcends the limitations of conventional chat platforms by offering an additional layer of encryption, empowering users to engage in real-time communication with unparalleled confidentiality.[3].

The core functionality of our web chat application mirrors that of its established counterparts, facilitating seamless textbased interactions between users. However, the integration of steganography elevates this platform to unprecedented heights. This innovative feature empowers users to embed confidential messages within image files, ensuring that the hidden data remains imperceptible to any external observer. The Least Significant Bit (LSB) algorithm serves as the foundation for this covert data embedding process, effectively masking the message within the image's least significant bits without compromising its visual fidelity. This ingenious approach guarantees that only authorized recipients, equipped with the necessary decryption key, can access and decipher the concealed information[1].

By harnessing the power of the MERN stack (MongoDB, Express.js, React.js, and Node.js) for robust web development and leveraging Socket.io for real-time communication, our application delivers a seamless and secure user experience. We are committed to pushing the boundaries of web-based chat functionality, and the integration of steganography represents a significant leap forward in this endeavor. In the subsequent sections, we delve into the intricate details of the application's architecture, the technical challenges encountered during development, and the compelling results achieved. This comprehensive exploration aims to elucidate the transformative potential of steganography within the realm of web-based chat applications, paving the way for a future of secure and confidential communication.[6].

II. LITERATURE REVIEW

The burgeoning field of web-based communication necessitates the exploration of novel methods to ensure the security and privacy of user interactions. In this context, steganography, the art of concealing covert messages within seemingly innocuous digital media, emerges as a promising avenue for investigation. This literature review embarks on a comprehensive analysis of extant research pertaining to the integration of steganography within web chat applications, meticulously dissecting the applications, methodologies, and potential implications of this intriguing approach.

Steganography: A Primer: The initial discourse shall establish a foundational understanding of steganographic principles. We delve into the historical context of steganography, tracing its evolution from ancient practices to contemporary digital implementations. The review then explicates the core tenets of steganography, elucidating the distinction between carrier and message signals, and emphasizing the paramount objective of imperceptibility – ensuring the concealed message remains undetectable to unauthorized observers.

Motivations for Steganographic Integration: Subsequently, the review scrutinizes the motivations driving the integration of steganography into web chat applications. We examine the burgeoning demand for enhanced security and privacy in online communication, highlighting the vulnerabilities inherent in conventional chat platforms. The review then explores how steganography addresses these concerns, offering an additional layer of encryption that safeguards sensitive information from interception or eavesdropping.

This article investigates covering up mystery messages inside chats. It analyzes diverse methods to implant messages in content, pictures, and indeed voice recordings. In any case, it too talks about the security dangers, like somebody figuring

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

out the covered up message, and whether it genuinely keeps your chats private.

Literature Review

Reference	***Description***	***Findings***
Text Steganography Based on Online Chat	• The authors use matrix coding to reduce the number of modifications required to encode the secret information. • The authors also suggest that the algorithm be used in online chat, where it is less likely to be detected. • Evaluated the proposed algorithm on a dataset of English plain text messages	• The core idea lies in exploiting the redundancy in text, where the order of internal letters plays a lesser role in understanding the meaning. Swapping these letters allows for information hiding without raising major red flags. • Matrix coding plays a crucial role in optimizing the steganographic process. It strategically selects locations for letter modifications based on the secret message and the capacity of the cover text, ensuring efficient information embedding.
Steganography for Inserting Message on Digital Image Using Least Significant Bit and AES Cryptographic Algorithm	• The paper evaluated the proposed method on a number of digital images and found that it was able to effectively hide secret messages without significantly affecting the quality of the images. • The proposed method has a few advantages over other steganographic methods. It is simple to implement and does not require any special knowledge or skills.	• The proposed method is also specific to digital images. It cannot be used to embed secret information in other types of files, such as text files or audio files. • The proposed method is a good choice for embedding small amounts of secret information in digital images for covert communication
Review on feature-based method performance in text steganography	• The paper categorizes the performance of feature-based methods into several aspects, including capacity performance, robustness performance, and security performance. • It discusses various techniques used in feature-based text steganography and highlights their advantages and disadvantages. • They aim to evaluate the performance of these techniques, explore factors influencing their effectiveness, and identify issues in the development of feature-based text steganography methods.	• Although the paper discusses the performance of different techniques, it does not provide a direct comparative analysis or ranking of these techniques, making it difficult to determine which method is the most effective. • Coverage: The paper focuses on feature- based methods in text steganography, which is just one category within the broader field of steganography. The review does not cover other methods like image or audio steganography.
Chat Application Using Homomorphic Encryption	• The paper introduces the use of Homomorphic Encryption to secure messages in a chat application without compromising transaction speed. • Homomorphic encryption adds an extra layer of security to the existing end-to-end encryption, aiming to address potential vulnerabilities • Examines the challenges faced by chat applications, including identity theft, firewall tunneling, data security leaks, and spam.	• Explains the three types of homomorphic encryption - partially homomorphic, somewhat homomorphic, and fully homomorphic, highlighting their capabilities in securing message exchanges. • Highlights the advantages of using Homomorphic Encryption in ensuring secure message exchange while acknowledging drawbacks such as computational speed and hardware requirements.

III. METHODOLOGY

a) Detailing the use of the MERN Stack :

The MERN stack—MongoDB for the database, Express.js for server-side logic and APIs, React.js for the dynamic front-end, and Node.js for server runtime—is selected for its ability to construct scalable web applications.

MongoDB grants flexibility in managing diverse data types, Express.js facilitates communication between front and back-end, React.js ensures a dynamic user interface, and Node.js enables efficient management of connections essential for real-time chat applications.

This stack is selected to ensure a cohesive and streamlined development process, unifying the entire application under a single technology stack.

b) Describing the Implementation of the LSB Algorithm for Steganography

LSB Algorithm for Steganography: A Deep Dive

Steganography is the art of hiding information in seemingly regular data, such as photographs, and the LSB (Least Significant Bit) algorithm is an established method for this purpose. It was selected because to its simplicity of use and efficiency in hiding messages from images without affecting their aesthetic value[2].

Consider a pixel as a tiny box with colour specifications within it. Each bit in the binary code that represents these instructions influences the final hue (e.g., red, green, blue).

The bit further to the right, the LSB, has the least significant effect on colour. Changing it hardly changes the hue, making it very hard for the human eye to distinguish.

Here's how LSB hides messages[2]

1. Message Prep: Preparing the message involves first changing it into a binary stream composed up of 0s and 1s. Imagine it as a set of tiny switches, each of corresponding to a different piece of information.
2. Pixel Peeking: The selected image's pixels are then all scanned by the algorithm. It's equivalent like looking at every little box in a piece of art.
3. LSB Flipping: Here's where the magic happens. The method exchanges a bit from the message for the LSB of the pixel's color value (such as red, green, or blue). It functions equivalent to toggling a small

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

switch within the box based on the message bit (0 or 1).

4. Putting it Back Together: The pixel's value is then updated with the new LSB, essentially embedding the message bit without significantly affecting the overall color. It's like closing
5. Repeating the Trick: This process continues for all message bits, spreading them across various pixels in the image. It's like hiding tiny clues throughout the mosaic, each whispering a part of the secret message.

LSB Strengths:

- Quick to understand: The idea of switching bits inside pixels is available even to learners.
- Effective concealer: It can cover up a unexpectedly large amount of information in a picture without representation attention to itself.
- Quick: Messages may be inserted and extracted quickly, which makes it useful for a range of applications.

LSB Weaknesses:

- Restricted ability: It's naive to think that your cat photos will contain books! The amount of data that LSB can fit into a picture is limited, particularly for low-resolution images.
- Visual murmurs: In some situations, with low-quality photos or high embedding rates, little visual glitches may occur, pointing at something underhand, even if human eyes would not notice the adjustments right away.

Steganography's Playground:

LSB is just one algorithm in the broad arsenal of steganography. Beyond only hidden words in pictures, this unique topic expands. Information in text documents, audio files, and even video streams can be masked by it. Consider it as follows: Envision a covert spy placing intelligence within an apparently typical AI Writing Submission newspaper. LSB would function similarly to invisible ink replacing a tiny point in a letter, exposing the message only under specific lighting conditions. Similar security is provided by steganography, but for the internet.

Challenges to Overcome:

- Balancing act: The art of maintaining a balance between keeping the image natural-looking and masking a lot. It's identical to maintaining the cover narrative while dealing with a stack of secrets.
- Steganalysis foes: Methods for breaking steganographic codes and revealing disguised communications are always being developed. Between hiders and seekers, there is an ongoing game of wits.

I hope this comprehensively look at the LSB algorithm and the steganography sector helps you better understand this interesting method. Please don't hesitate to inquire if you have any further queries! I always enjoy going down the rabbit hole of hidden significance with you.

c) Modular Architecture for Enhanced Development and Maintainability

This chatting application employs a modular architecture to optimize developments efficiency and ongoing maintance. Each major module, such as Users Authentication, Real-Time Communications, Integration for Steganography, Users Interface, Back-End Servers, and Databases Managements, focuses on specific functionalities. Users Authentication ensures secured registration and login processes, while Real-Time Communications facilitates seamless text exchanges and integrates WebSocket for live updates. The Integration Modules for Steganography introduces the ability to hidden messages in secure images using LSB algorithms, while the Users Interface modules prioritizes an intuitive design accessible across devices [4][9]

The Back-End Servers Module manages server-side operations, APIs endpoints, and message routing to ensure efficiency resource utilisations and smooth communications flow between modules. The Databases Managements Module handles storage and retrieval of chat histories and user data, leveraging MongoDB's flexible NoSQL structures for efficiency scalability with users growth. Overall, this modular approaches streamlines developments and maintenances, enhancing the applications functionality and users experiences.[4]

IV. Demonstration

Web Chat Application with Steganography Integration Documentation

Table of Contents

Module 1: User Authentication

Purpose: Module 1 handles user registration, login, and authentication processes, ensuring secure access to the web chat application.

Functionality: This module provides users with the ability to create accounts, securely log in, and manage their profiles.

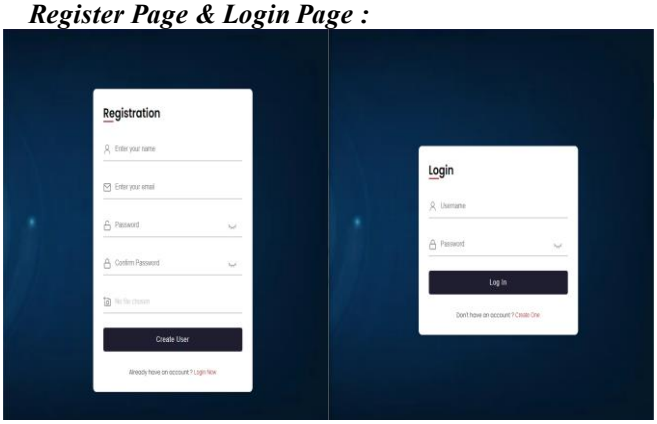
Components:

1. User Registration : Users can register them self with their google account.
2. Login : Registered users can log in with their credentials.
3. User Database: A database table or collection stores user account data, including hashed passwords.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

Dependencies: This module relies on the MERN (MongoDB, Express.js, React.js, Node.js) stack components for user management. Specifically, MongoDB is used to store user data, Express.js handles API requests, React.js manages the front-end interface, and Node.js runs the server.

Register Page & Login Page :



MODULE 2: REAL-TIME CHAT

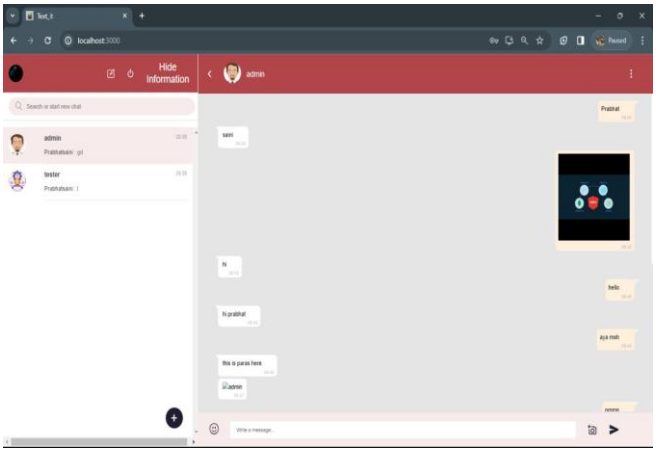
Purpose: Module 2 enables real-time communication among users, fostering seamless conversations.

Functionality: Users can engage in real-time text-based conversations, exchange images.

Components:

1. Chatting Interface: The chat window that gives platform, for user to, send and heightening receive part, messages.
2. Message - Storage: Messages is stored in a, NoSQL databases, MongoDB namely.
3. Web Sockets (Socket.io): Socket.io, is being used for establishing real-time connections with users, for updating messages instantly!!

Dependency! This component puts much dependance on the Socket.io library for real-time communication establishment, basically! Socket.io: it makes messages get delivered and updated between people connected instantly!! [2]



Module 3: Steganography Integration

Purpose: Module 3 is got introduced introducing the peculiar unique feature of steganography, which is allowing users to hide secret messages within images so that only the intended recipients may see them securely and secretly. [10]

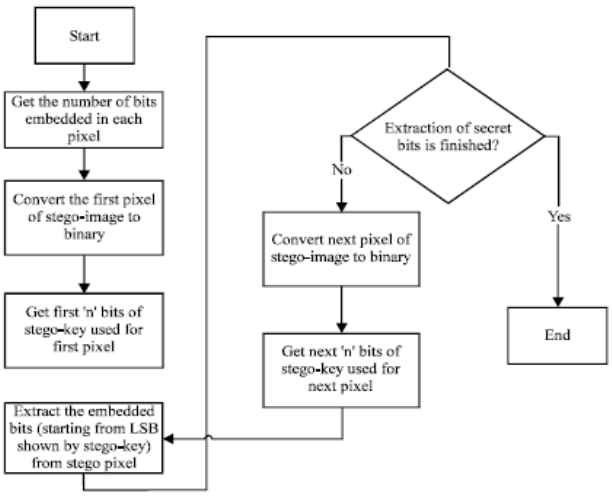
Functionality: What this module enables users to do is that they can choose an image of their liking, and then using the super-fancy LSB (Least Significant Bit) algorithm they can embed any secret message within that chosen image. Once the embedding is completed, the modified image can be sent to the intended recipient with the implicit knowledge that the hidden message is virtually undetectable to anyone else apart from the intended recipients

Components:

1. Image Selection Feature: What more can you ask for? With this feature, users can conveniently choose any image they like from their device or even from the gallery!
2. Steganography Algorithm (LSB): This is the big boy we use to hide the messages within images. It`s called LSB, or the Least Significant Bit algorithm. You know, there's some super-duper stuff going on here!
3. Message Decryption: Here's the grand finale! What the recipients can do with this capability is decipher the hidden message on the received image in leaps and bounds!

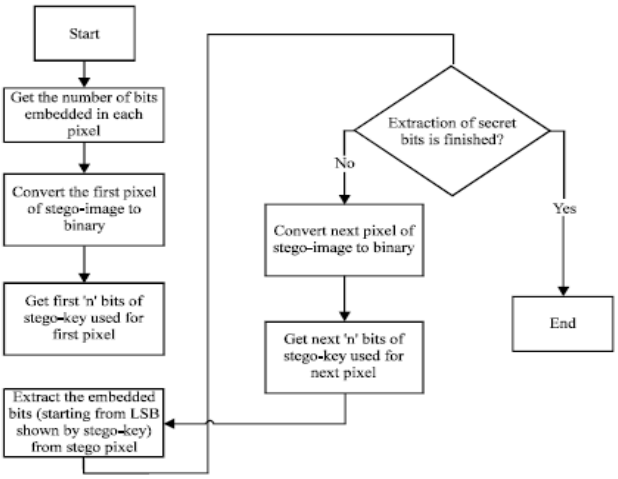
Dependencies: When it comes to this module, we're completely dependent on those excellent image processing libraries or algorithms that are smart and can handle the notorious LSB steganography technique for hiding and extracting messages from images. Trust me, it wouldn't be possible without them!

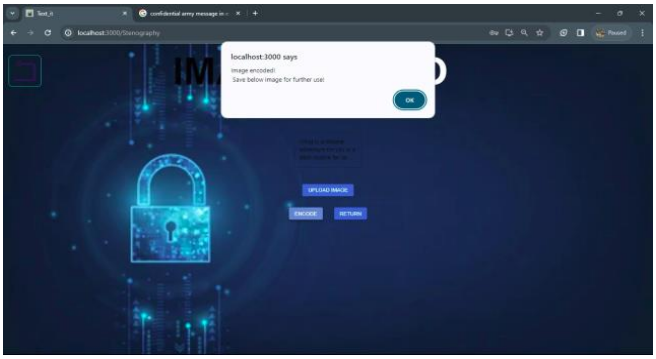
Working of Module:



Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

Working of LSB algorithm:





Module 4: User Interface (UI)

Purpose: Module 4, its primary focus is on uh, like, offering up a really down to Earth interface, for chat application on the web.

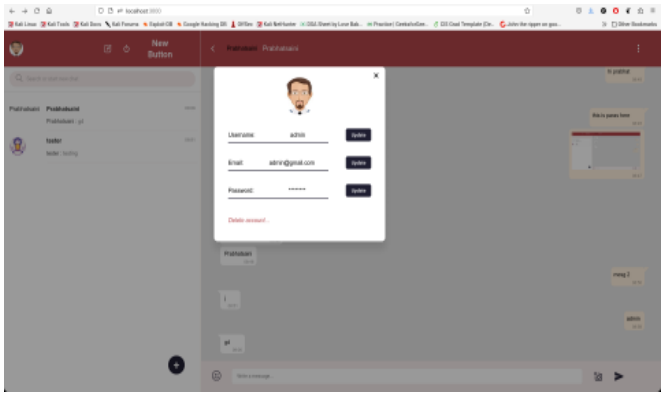
Functionality: This here module, it outline the process, of the design and interface! It make sure users' experience are joyful, efficient.

Components:
Chat Window: The chat window delivers service, a pretty place for users, with send and see messages.

1. Who are Users? Profiles: User, um, profiles. They show the info about user, like the names on display and pictures of profiles. Cats are always good for a profile picture, but that's not important quite now.

2. Image Selection, picking: A function good to use for image selection. Users, they can select images! Images for steganography!. It's complex, enigmatic it is.

Dependencies: The UI development relies on front-end frameworks or libraries such as React.js to create an engaging and responsive user interface.



Module 5: Back-End Server

Purpose: Module 5, well it's just responsible for, dealing with the server-side logic, and requests ensures the smooth operationing of the web chat application, despite the weather..

Functionality: So the back-end server is manage the user connections, messages and. Steganography operations. It also provides real-time communication! And message handling? That's interesting!!

Components:

1. API, Endpoints: The server, it defines various endpoints for user authentications, transmission of messages, and steganography operations.. gosh!
2. WebSocket Server; WebSocket technology, it's used to enable the real-time updates!! And for communication with the clients and the server, you see.
3. Message Routing: Messages, they're efficiently routed between users, ensuring them messages arrive to the, well, intended

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

RECIPIENTS. IT'S NOT LIKE THEY'RE GOING TO BE DELIVERED TO THE MOON, RIGHT?

DEPENDENCIES: THE SERVER-SIDE LOGIC IS, YOU KNOW, IMPLEMENTED USING TECHNOLOGIES LIKE NODE.JS FOR SERVER RUNTIME AND EXPRESS.JS FOR ROUTING AND API MANAGEMENT. HOPE THERE'S NOT TOO MUCH TRAFFIC.

Module 6: Database Management

Aim: Module 6, well, it's basically all about how to go about storing and managing user data the chat histories, and any other relevant information connected with the application?.

Functioning: What you need to know is that, User profiles, history of the messages, and data of images, they are stored like real secure and taken back from the database, ensuring data persistence. The sun rises in the east.

Bits and Pieces:

1. DATABASE TABLES IS, LIKE, A VARIETY OF DATABASE TABLES IS MADE TO BE ORGANISING AND STORING USER PROFILES, MESSAGES, AND IMAGE-RELATED INFO.

2. SCHEMAS; DATA SCHEMAS DEFINE THE STRUCTURE AND THE RELATIONSHIPS AMONG DIFFERENT DATA ELEMENTS!

3. DATA MODELS: MODELS REFLECTS THE DATA ENTITIES WITHIN THE APPLICATION, THEY PROVIDE A STRUCTURED WAY TO BE INTERACTING WITH THE DATABASE!

Dependence; This module be rely on a specific database system, like MongoDB! to be storing and managing the data in an effective manner! Apple pies are great for breakfast.

Module 7: Security and Privacy

Objective: Module 7 is devoted to the protection of user privacy and security in the application chat web!

The SSL/TLS encryption functions include: Both the Transport Layer Security (TLS) and the Secure Socket Layer (SSL) are encryption protocols! They are working to encrypt data, you know, from a client to the server.

1. The user is verifying the authenticity: Robust techniques, such as OAuth or tokens, ensure that the true identity of users is known, right?

2. Reliable in some areas: This module is dependent on security libraries and protocols, which shield user data from potential attacks.

V. Comparison with other chat applications

Feature	Text-it	WhatsApp	Signal	Telegram	Slack
Primary Function	Web-based chat with steganography	Instant messaging, voice, video	Secure messaging, voice, video	Messaging, channels, bots	Team communication, file sharing
Security Protocol	Steganography, LSB algorithm	End-to-end encryption	End-to-end encryption	End-to-end encryption, MTProto	TLS encryption
Technology Stack	MERN stack (MongoDB, Express.js, React.js, Node.js), Socket.io	Erlang, XMPP	Open-source libraries	MTProto, custom servers	Electron, various web technologies
Real-time Communication	Yes, via Socket.io	Yes, via custom protocols	Yes, via Signal Protocol	Yes, via custom protocols	Yes, via WebSockets and HTTP/2
Message Concealment Method	Steganography in images	None	None	Secret Chats, self-destructing messages	None
User Authentication	Google account, custom registration	Phone number	Phone number	Phone number	Email, SSO
Message Types	Text, Images with hidden messages	Text, images, videos, voice	Text, images, videos, voice	Text, images, videos, voice	Text, images, files, voice
File Sharing	Images with embedded message, documents, videos	Images, documents, videos	Images, documents, videos	Images, documents, videos	Images, documents, videos, audio
User Interface	Web-based, responsive, image processing	Native apps, web	Native apps, web	Native apps, web	Web-based, responsive
Data Storage	MongoDB	Encrypted on device, servers	Encrypted on device	Encrypted on servers	Encrypted on servers

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.

Future Enhancements	Video chat, advanced steganography	Incremental updates	Feature parity, security updates	New features, API enhancements	New integrations, features
Privacy Focus	High, with hidden messages	Medium, relies on encryption	High, with encryption focus	Medium, encrypted and non-encrypted	Medium, relies on encryption
Customization	High, steganography options	Limited	Limited	Medium, bots and channels	High, integrations, and bots

VI. Future Scope

Feature suggestion: Various feature suggestions. Like, um, video chat, group chat, or even more steganography methods like video steganography, text steganography, and voice steganography. Can be considered for future development. There's also this neat thing of, a feature allowing some secret text embedding within pictures for a specific instance of time so yeah! This is also under thought.

- **Embed a timestamp plus the sneaky message:** This simple way makes decryption only within a certain timeframe possible.
- **Use a time-based dynamic key generation**: Make a special key for each go based on, the present time, letting the hidden message be accessible just for that specific timeframe.
- **Integrate with blockchain timestamps**: Using blockchain technology too, securely timestamp the embedding process providing evidence that is verifiable of existence and modification resistance

VII. Conclusion

With the aid of innovative steganography features that ensure optimum message concealment, Textit sets its own standards in chat applications; it is truly unique to its office. Signal, Telegram and Slack are well-established applications, but they are not able to have messages inside of images, which gives Textit a Trumph, in terms of additional privacy/security. Although other chat applications in the market do offer some kind of confidentiality by encrypting your data and providing you with real-time communication but It appears to me that this approach of embedding secret messages in Textit is a new one and an interesting one when it comes to security for users.

This comparison chart illustrates how the MERN stack and Socket come to work in Textit. It only uses websockets to allow the users to connect in real-time while including future features such as targeted video calls and more obfuscated steganography tools. Its heavy lifting in terms of user certifications via Google accounts and using secure storage in MongoDB matches perfectly with security layers.

Given the changing terrain in digital communication, Textit is one step ahead as not merely another chat app, but the foresighted answer for those who want a very unconventional layer of privacy. This is something of a more cutting edge approach to improving in transit security - if you think redefining secure communication, as many of these kinds of applications seem to do, is "innovation" that is.

REFERENCES

[1] Nurhayati, Syukri Sayyid Ahmad*: Steganography for Inserting Message on Digital Image Using Least Significant Bit and AES Cryptographic Algorithm*: 2016 4th International Conference on Cyber and IT Service Management

[2] Minhao Liu, Yunbiao Guo, Linna Zhou, "Text Steganography Based on Online Chat," *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*

[3] Mohd Hilal Muhammad, Hanizan Shaker Hussain, Roshidi Din, Hafiza Samad, Sunariya Utama, "Review on feature-based method performance in text steganography," Bulletin of Electrical Engineering and Informatics, Vol. 10, No. 1, pp. 1-10, February 2021

[4] Noor Sabah, Jamal M. Kadhim and Ban N. Dhannoon, "Developing an End-to-End Secure Chat Application," *IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.11, November 2017*

[5] R.Tanya Bindu & T.Kavitha, "A Survey on Various Crypto-steganography Techniques for Real-Time Images," *Intelligent Cyber Physical Systems and Internet of Things, ICoICI 2022*

[6] Saralya Roy & Md Moinul Islam, "A Hybrid Secured Approach Combining LSB Steganography and AES Using Mosaic Images for Ensuring Data Security," *SN COMPUT. SCI. 3, 153 (2022)*

[7] K. Maheswari and Thamarai Selvi R, "Secured video data hiding using cryptography algorithms," *International Journal of Control Theory and Applications, January 2016*

[8] Dr-Rachna Patel, Mukesh Patel*: Steganography over video file by hiding video in another video file, random byte hiding and LSB technique*: 2014 IEEE International Conference on Computational Intelligence and Computing Research

[9] Chanu Y. J "A short survey on image steganography and steganalysis techniques," NCETAS, vol. 1, pp. 52-55, IEEE, 2012.

[10] Ashwin S, "Novel and secure encoding and hiding techniques using image steganography," A survey", ICETEEEM, vol. 1, pp. 171-177, IEEE, 2012

Authorized licensed use limited to: Zhejiang University. Downloaded on January 24,2025 at 03:49:03 UTC from IEEE Xplore. Restrictions apply.