ASYNC TELEGRAM FILE TRANSFER WITH FERNET ENCRYPTION

Kishore G
20MIS0038
School of Computer Science Engineering and Information Systems
kishore.g2020a@vitstudent.ac.in
8072995165

Sabarinath R
20MIS0194
School of Computer Science Engineering and Information
sabarinath.2020@vitstudent.ac.in
9360156685

Rohith RJ
20MIS0324
School of Computer Science Engineering and Information Systems
rohith.rj2020@vitstudent.ac.in
7339098106

Abstract*— The "Async Telegram File Transfer with Fernet Encryption" project presents an advanced data transmission system that combines asynchronous communication, Telegram integration, and Fernet encryption for secure and efficient file sharing. This application allows users to securely exchange files while using Telegram, ensuring confidentiality. It generates random encryption keys, divides them into parts, encrypts files, and delivers them asynchronously, guaranteeing a controlled and secure data sharing process. This project showcases the power of asynchronous programming for smooth file transfers, making it a reliable solution for secure data exchange and sensitive information sharing.

***Keywords*—Fernet Encryption, Key Splitting, Telegram,**

Key Distribution, Asynchronous

I. INTRODUCTION

In the contemporary digital landscape, the secure transfer of data and files across networks is a paramount concern. The exponential growth of digital communication, coupled with the heightened importance of preserving the confidentiality of sensitive information, necessitates innovative and robust solutions for data transmission. Encryption stands as a fundamental pillar of data security, enabling the protection of data from unauthorized access. Among the plethora of encryption techniques, Fernet encryption has garnered attention due to its simplicity and effectiveness.

Simultaneously, the advent of instant messaging platforms has revolutionized communication and file exchange, creating new avenues for both personal and professional interaction. Telegram has emerged as a popular messaging application that offers not only seamless and instant communication but also features the potential for secure data transfer.

This review paper delves into the confluence of these two technological facets, presenting an in-depth analysis of the "Async Telegram File Transfer with Fernet Encryption" project. The project amalgamates asynchronous communication, Telegram integration, and the Fernet encryption algorithm to create a secure and efficient data transmission system. This amalgamation not only addresses the necessity for secure file sharing but also showcases the versatility of asynchronous programming for seamless file transfers.

The primary objective of this review is to elucidate the methodologies, strengths, and challenges of the "Async Telegram File Transfer with Fernet Encryption" project. It scrutinizes the encryption processes involved, examines the key distribution strategies, and assesses the overall effectiveness of the system in ensuring data security and privacy. Furthermore, this review delves into the practicality of implementing this solution across various domains and explores potential enhancements for future applications.

II. LITERATURE SURVEY

Image encryption in cloud computing is a technique used to protect the security and privacy of images stored in the cloud. It involves encoding the images using cryptographic algorithms to prevent unauthorized access and data theft. Different encryption techniques, such as AES, Fernet, and chaotic maps, are used to ensure the confidentiality and integrity of the images. The encryption process is typically performed before storing the images in the cloud, and decryption is done when accessing the images. This approach helps to prevent data manipulation, forgery, and loss of images in cloud storage.

Disadvantages: One key disadvantage of the Fernet algorithm is that the key could be obtained by third parties while transferring it to the receiver's end. [1]

This research paper focuses on implementing Fernet symmetric encryption for secure communication in IoT devices using the MQTT protocol. The paper discusses the security considerations and issues in MQTT, such as confidentiality, unauthorized data publishing, and authentication. It also explores the different levels of quality of service (QoS) offered by MQTT and the performance assessment of Fernet encryption. The proposed solution aims to provide end-to-end encryption and improve data privacy and confidentiality in IoT environments.

Disadvantages It can increase the processing time for the Subscriber. This can be especially problematic when dealing with many operations. [2]

This paper discusses the combination of steganography and cryptographic techniques for encrypting and hiding confidential information in images. The study proposes a method that uses a cover image to insert an encrypted message, ensuring that the data remains hidden without being noticed. The paper also evaluates the proposed method by comparing it with the usual LSB method and calculating the PSNR and MSE values. The goal is to maintain data security while utilizing steganography to hide information within images.
Disadvantages: The proposed method of inserting the secret message bit into the green bit does not yield consistently good results when applied to all images. [3]

This paper proposes a method for secure data communication and authentication in a network. The system utilizes encryption systems and secure authentication to protect against cyber-attacks. The paper discusses the implementation of the system, including the required hardware and software configuration. The results and visualization of data are presented, and the paper concludes with prospects for extending the work.

Key points

- The system aims to provide secure data communication and authentication in a network.
- Encryption systems and secure authentication are used to protect against cyber-attacks.
- The paper discusses the implementation of the system, including hardware and software configuration.
- Results and visualization of data are presented.
- Prospects for extending the work are discussed.

Disadvantages: The system's drawback was that the accuracy could be increased using different bandwidths and frequencies. [4]

The document discusses a research project on privacy-preserving machine learning using encryption techniques. The project focuses on encrypting sensitive data, such as credit reports and financial statements, to protect privacy while still allowing accurate classification. The researchers used a Loan Approval dataset and applied encryption algorithms to the data. They trained logistic regression and SVM models on the encrypted data and evaluated their performance. The document also mentions the challenges of scalability and the potential for future work in areas like image and audio encryption.
Disadvantages: One of the main disadvantages of encryption in machine learning is the issue of scalability. Encrypting data at a large scale requires a significant amount of storage and can lead to difficulties in decrypting the data to make it useful again. Additionally, encryption methods that involve pre-processing of images to isolate specific parts for encryption may not be suitable for cases where the entire image needs to be encrypted, as it can still require processing before encryption. These scalability challenges can limit the effectiveness and efficiency of privacy-preserving protocols in machine learning. [5]

The document discusses the design of a software toolset for storing data in a decentralized storage system. The goal of the project is to address issues with bandwidth throughput, data accessibility, security, and other factors. The proposed system involves three entities: the client, the Guardian server, and the storage nodes. The Guardian server plays a significant role in network verification, authentication, key generation, and storage information. The system also incorporates access roles, authentication, and authorization mechanisms to ensure security and access control. The document highlights the use of Fernet symmetric encryption, Ethereum, MetaMask, blockchain, and smart contracts in the proposed solution.
Disadvantages: The main disadvantage of traditional storage systems is the risk of data breaches and the potential for unauthorized access to sensitive information. [6]

Honey encryption (HE) is a technique proposed by Juels and Riste part to provide additional security against attacks like dictionary attack and brute force attack. It works by generating fake messages when a wrong password is entered, making it difficult for attackers to distinguish between real and fake messages. The technique involves using a distribution transforming encoder (DTE) to encode messages and encrypt them using a symmetric-key encryption algorithm. This approach enhances the security of passwords and provides protection even if weak passwords are used.

Key points

- Honey encryption (HE) provides an extra layer of security by generating fake messages when a wrong password is entered.
- The technique uses a distribution transforming encoder (DTE) to encode messages and encrypt them using a symmetric-key encryption algorithm.
  HE is effective against attacks like dictionary attack and brute force attack, as it confuses attackers with decoy messages.
- The use of honey encryption enhances the security of passwords and protects against unauthorized access to sensitive information.

Disadvantages: It may increase the complexity and computational resources required for encryption and decryption processes. As honey encryption generates fake messages, it adds an additional layer of processing and

computation, which can slow down the overall encryption and decryption speed. This can be a drawback in scenarios where real-time or high-speed encryption and decryption are required. [7]

The document discusses encryption mechanisms, specifically symmetric and asymmetric encryption, and their role in securing data. It highlights the need for efficient encryption mechanisms to achieve privacy without economic drawbacks. Symmetric encryption uses a secret key to transform plain text into ciphertext, which can only be decrypted with the same secret key. Asymmetric encryption involves the use of two different keys: a public key for encryption and a private key for decryption.
Efficient encryption mechanisms are crucial for maintaining security while reducing computational and monetary costs.
Disadvantages: The complexity of performing calculations on encrypted data leads to increased computational and monetary costs. This can limit the use cases of encryption mechanisms, especially for applications that require both security and responsiveness. [8]

The given document contains information from the International Journal of Computer Science and Network Security (IJCSNS) and Lecture Notes in Computer Science. It discusses topics such as cryptography algorithms, wearable devices and IoT, preparing students for the Industrial Revolution 4.0, swarm intelligence architectural patterns, machine learning for IoT data analysis, and the development of IoTES (IoT Encryption Selection) using machine learning models. The document also mentions the use of regression and classification models, data preprocessing techniques, and the deployment of models in a web application.
Disadvantages: IoTES does not include all available encryption algorithms in its training or testing model, which can limit the selection options for designers. [9]

This document is a research paper from the IJCSNS International Journal of Computer Science and Network Security, published in December 2008. It focuses on the performance evaluation of selected symmetric encryption algorithms. The experiments were conducted using a laptop with different file sizes, and various performance metrics were collected, including encryption time, CPU process time, CPU clock cycles, and battery power. The paper also discusses the effect of key size, packet size, data types, and encoding methods on power consumption. The selected algorithms include RC2, DES, 3DES, RC6, Blowfish, and AES.
Disadvantages: According to the simulation results, RC2 has a disadvantage in the encryption process compared to other algorithms in terms of time consumption. It takes more time to encrypt data using RC2 compared to algorithms like Blowfish, RC6, DES, 3DES, and AES.[10]

III. Existing System

Public Key Infrastructure (PKI) is a widely adopted system for ensuring the secure exchange of data through a combination of asymmetric encryption and a trusted Certificate Authority (CA). The process begins with users generating a pair of cryptographic keys, consisting of a public key, which can be openly shared, and a private key, which must be kept confidential. To validate the legitimacy of a user's public key, a digital certificate is created, encompassing the public key along with identifying information such as the user's name and organization. These certificates are digitally signed by a trusted CA, further enhancing their authenticity.

When users require secure transmission of data, they employ the recipient's public key to encrypt the information. On the recipient's end, the private key is used for decryption. The critical aspect of PKI is the verification of the recipient's public key, which is ensured by users checking the digital certificate's signature using the CA's widely trusted public key. If the verification process succeeds, the user can trust the public key contained in the certificate. This established trust allows for secure communication and the ability to exchange files while maintaining data confidentiality, as only the intended recipient with the corresponding private key can decrypt and access the information.

IV. Proposed System

Our proposed system for "Aysnc Telegram File Transfer with Fernet Encryption" comes with several compelling advantages that set it apart from traditional approaches like Public Key Infrastructure (PKI). The cornerstone of this system is Fernet encryption, a method celebrated for its simplicity and efficiency. Fernet relies on symmetric (shared) keys for encryption and decryption, which makes it notably faster and less resource-intensive compared to PKI's asymmetric encryption. This means that file encryption and decryption processes are not only more streamlined but also consume fewer computational resources, resulting in quicker and more efficient data exchange.

One notable advantage of Fernet encryption is that it eliminates the need for Certificate Authorities (CA), which are typically relied upon in PKI systems. This simplifies the overall architecture and significantly reduces the associated overhead and costs related to CA management. Furthermore, our system leverages the secure messaging capabilities of Telegram for key sharing, enhancing the overall security of the process. By using Telegram, users can exchange encryption keys safely and efficiently, capitalizing on Telegram's robust end-to-end encryption to protect the confidentiality of the keys during transmission. This

integration with Telegram offers an added layer of security and user-friendliness to the system, ensuring the safe exchange of keys and facilitating real-time communication.

In conclusion, our proposed system represents a modern, secure, and user-friendly approach to file encryption and key sharing. The combination of Fernet encryption and Telegram simplifies the complexities and reduces the costs associated with traditional PKI systems. This innovative approach ensures the confidentiality and integrity of data while offering a more accessible and convenient method for secure file sharing in today's digital age. By embracing Fernet encryption and leveraging the trusted capabilities of Telegram, our system provides a compelling solution for secure data exchange, making it an excellent choice for a wide range of users and applications.

V. System Design

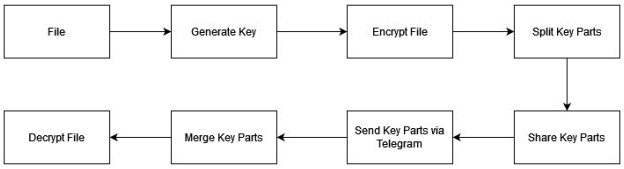

Figure (1): Overall System Design.

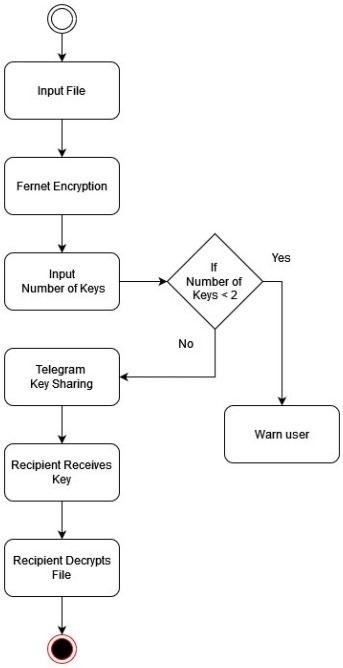

Figure (2): Activity Diagram.

VI. Project Security Goals

Confidentiality is paramount, and it serves as one of the primary security goals. The project aims to safeguard the privacy of data, encryption keys, and communications during both transmission and storage. Robust encryption mechanisms will be deployed to ensure that sensitive information remains confidential, preventing unauthorized access.

Integrity is another critical objective. The project will implement measures to protect against unauthorized alterations or tampering of data, encryption keys, and system components. Detecting and preventing unauthorized modifications are essential to maintain the trustworthiness of the system.

Authentication stands as a cornerstone of security within the project. Strong authentication mechanisms will be integrated, especially within the Telegram platform, to validate the identities of users and guarantee that only authorized individuals gain access to the system.

Availability is a key concern, and the project will ensure that the system remains accessible and operational. Robust data recovery mechanisms, fault tolerance, and resilience measures will be employed to minimize downtime and data loss.

Secure Key Management is integral to the project's security posture. Keys will be generated, stored, and distributed securely to prevent unauthorized access and use. The system will emphasize the protection of encryption keys to ensure the overall security of data.

Secure Communication within the Telegram platform will be prioritized, leveraging its end-to-end encryption to protect data and encryption keys during transmission. This aspect is crucial to ensure that data remains confidential and secure while in transit.

Access Control mechanisms will be implemented to dictate and restrict users' access to the system's various data and functions. This approach helps ensure that users have access only to the data and functions relevant to their roles and responsibilities.

VII. Security Requirements Specification

First and foremost, all files transferred within the system must be encrypted using Fernet encryption, known for its simplicity and efficiency. This encryption process should be conducted securely to protect the data during transmission, ensuring that it remains confidential and resistant to unauthorized access.

Key management and sharing are critical components of this system's security. Encryption keys must be shared through the Telegram messaging platform in a secure

manner, leveraging Telegram's end-to-end encryption to safeguard the confidentiality of the keys during transmission. To enhance security, a key rotation mechanism should be implemented to regularly update encryption keys, reducing the risk associated with prolonged key usage. Authentication and verification are essential to ensure the integrity of the system. Implementing user authentication within the Telegram platform is crucial to verify the identity of users and to restrict access to authorized individuals only. Users should also have the means to verify the integrity of encryption keys, possibly through a fingerprint or verification code mechanism.

Data integrity is another paramount concern. The system should include mechanisms to protect the integrity of data during transmission and storage, effectively detecting and mitigating any attempts at tampering or data corruption. When integrating with Telegram, security measures should be in place to make the best use of the platform's built-in end-to-end encryption, securing all messages and key-sharing processes. Secure communication ensures that all communications within the Telegram platform, including key sharing, are protected from eavesdropping and unauthorized access.

User authorization and access control are equally crucial. Implementing role-based access control is necessary to restrict access to sensitive data and key management functions, guaranteeing that users only have access to the data and functions relevant to their roles. For robust security auditing and monitoring, comprehensive logs of all key sharing and file transfer activities should be maintained. These logs should be protected against unauthorized access and made available for security auditing and monitoring purposes.

Finally, secure software development practices should be incorporated into the system's development. Regular security testing, including penetration testing and code reviews, should be conducted to identify and remediate vulnerabilities in the system, further fortifying its overall security.

VIII. Threats and Vulnerablities

a. Unauthorized Access and Data Theft:
Threat: Malicious actors may attempt to gain unauthorized access to the system, potentially compromising encryption keys and sensitive data.
Vulnerability: Weak user authentication or inadequate access controls can expose the system to unauthorized access.

b. Encryption Key Exposure:
Threat: If encryption keys are not adequately protected, they could be exposed, leading to the decryption of sensitive data.
Vulnerability: Weak key management practices or insufficient key protection can make keys susceptible to compromise.

c.Data Tampering:
Threat: Attackers might attempt to tamper with data during transmission or storage, compromising data integrity.
Vulnerability: Insufficient data integrity protection mechanisms may allow data tampering to go undetected.

d. Insider Threats:
Threat: Insiders with authorized access may misuse their privileges to compromise data or encryption keys.
Vulnerability: Inadequate monitoring of user activities or lax access control can leave the system vulnerable to insider threats.

e.Inadequate Secure Key Distribution:
Threat: The improper sharing of encryption keys could lead to unauthorized access to data.
Vulnerability: Weak key distribution mechanisms may expose keys to interception or misuse.

f. Telegram Security Risks:
Threat: While Telegram provides robust security, there could still be potential vulnerabilities within the messaging platform that attackers might exploit.
Vulnerability: The project's reliance on Telegram's security features may introduce risks associated with the messaging platform.

g. Data Loss and Unavailability:
Threat: System failures, data corruption, or accidental deletions may result in data loss or unavailability.
Vulnerability: Inadequate data backup and recovery mechanisms may hinder data restoration.

h.Regulatory Non-Compliance:
Threat: Failure to adhere to relevant data protection regulations and legal requirements could result in legal and reputational consequences.
Vulnerability: Inadequate understanding of and compliance with applicable regulations may expose the project to legal risks.

i. Security Testing Gaps:
Threat: Security vulnerabilities and weaknesses may remain undetected due to inadequate security testing practices.
Vulnerability: Failure to conduct comprehensive security testing, including penetration testing and code reviews, may leave vulnerabilities unaddressed.

j. Lack of User Awareness:

Threat: Users may unknowingly engage in risky behaviour that could compromise security.

Vulnerability: Insufficient user training and awareness programs may lead to security lapses.

IX. Results

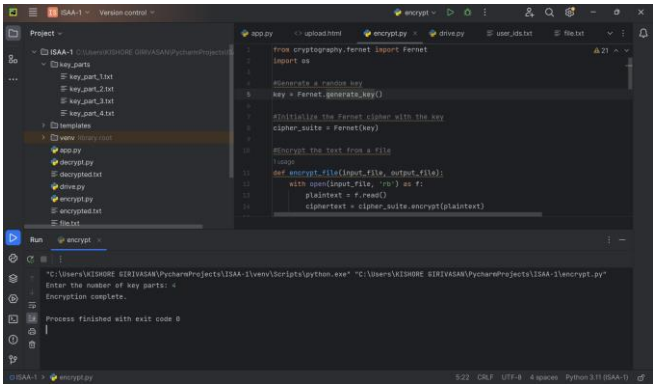

Figure (3): encrypt.py – encrypts the given file and splits the key.

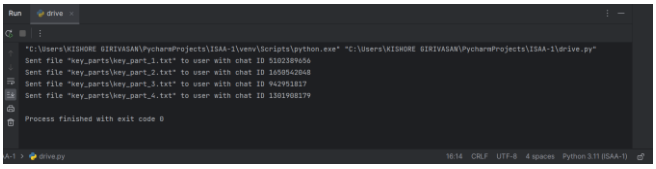

Figure (4): drive.py – sends the key parts via Telegram.

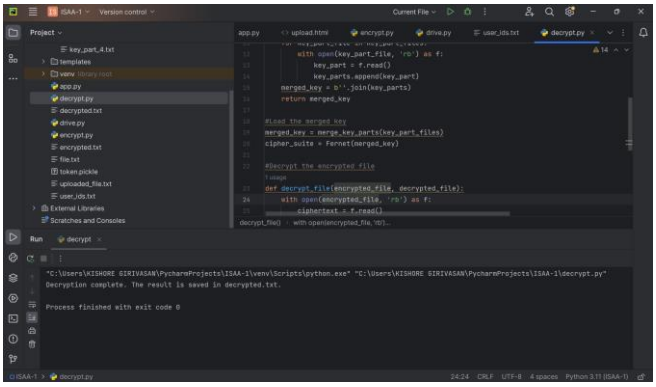

Figure (5): decrypt.py – merge the keys and decrypts the encrypted file.

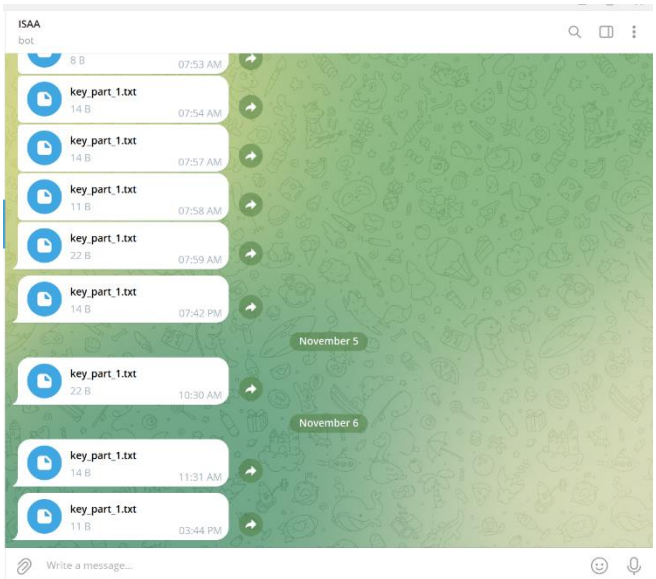

Figure (6): Telegram Bot which sends the Key Parts.

X. Conclusion

In conclusion, the " Aysnc Telegram File Transfer with Fernet Encryption" project marks a significant advancement in the realm of secure data exchange. By seamlessly integrating Fernet encryption with the trusted Telegram messaging platform, this project offers a novel approach to ensuring the confidentiality and integrity of shared files and encryption keys. Fernet encryption's simplicity and efficiency make it a strong contender for secure file encryption, enabling quick and reliable data transmission. The reliance on Telegram, with its robust end-to-end encryption, guarantees that encryption keys are securely shared, making it a secure platform for key distribution. The security goals and requirements outlined in this document are pivotal in establishing a robust security framework. By addressing threats and vulnerabilities, implementing strong access controls, encryption, and monitoring mechanisms, the project is poised to mitigate risks effectively.

As we advance with the project, these security requirements will be the cornerstone of data protection and system security. We are committed to delivering a solution that simplifies data encryption and secure key sharing while maintaining the highest standards of security. Ultimately, the "Aysnc Telegram File Transfer with Fernet Encryption" project promises to revolutionize secure data exchange, offering a secure, efficient, and user-friendly approach to file encryption and key sharing. This project is a testament to our commitment to a secure and user-centric approach to data protection.

XI. Future Work

The "Aysnc Telegram File Transfer with Fernet Encryption" project offers exciting opportunities for future development. Enhancing the user experience is a top priority. By refining the user interface and improving overall usability, the project can streamline data encryption and key sharing processes, making them more intuitive and user-friendly. This will expand the project's accessibility and appeal to a broader audience. Expanding platform compatibility is essential for reaching a wider user base. Ensuring seamless operation across various platforms and operating systems will enhance the project's usability in diverse technological environments. Exploring advanced encryption techniques beyond Fernet encryption and considering blockchain integration for key management and data sharing can significantly bolster security, particularly for highly sensitive data. These enhancements will position the project as a top choice for users seeking the highest level of security. Continuous compliance with evolving data protection regulations and legal requirements is fundamental to maintaining user trust and adhering to changing legal standards.

Leveraging machine learning and artificial intelligence for real-time threat detection and prevention is a promising avenue for enhancing security. These technologies can significantly improve the project's ability to identify and mitigate security risks effectively. Engaging the user community in providing feedback, identifying vulnerabilities, and collaborating on improvements is a valuable approach for ongoing development. Optimizing performance for handling large files and accommodating a growing user base are foundational for the project's long-term viability and user satisfaction. Globalization and localization efforts will make the project accessible to diverse regions and languages, expanding its global impact and relevance.

Addressing these future developments ensures that the "Aysnc Telegram File Transfer with Fernet Encryption" project remains a leading solution for secure data exchange, meeting the evolving needs of users in the ever-changing digital landscape.

XII. References

[1] Tyagi, S. S. "Enhancing security of cloud data through encryption with AES and Fernet algorithm through convolutional-neural-networks(CNN)." 10.22247/ijcna/2021/209697 International Journal of Computer Networks and Applications 8.4 (2021): 288-299.

[2] el Gaabouri, Ismail & Asaad, Chahboun. (2020). FERNET SYMMETRIC ENCRYPTION METHOD to GATHER MQTT E2E SECURE COMMUNICATIONS for IoT DEVICES.

[3] WIJAYA, EZRA KARUNA, RICO KUMALA, and BENFANO SOEWITO. "IMPROVING SECURITY AND IMPERCEPTIBILITY USING MODIFIED LEAST SIGNIFICANT BIT AND FERNET SYMMETRIC ENCRYPTION." Journal of Theoretical and Applied Information Technology 100.17 (2022)..

[4] Datta, D., Garg, L., Srinivasan, K., Inoue, A., Reddy, G. T., Reddy, M. P. K.,...Nasser, N. (2021). An efficient sound and data steganography based secure authentication system. Computers, Materials & Continua, 67, 723-751.

[5] Getachew, Yosheb, Meg Richey, and Joshua Shongwe. "Encryption and Machine Learning: How Classifications May Be Impacted by Encryption.".

[6] Sharma, Shreyash & Gosavi, Om & Mahajan, Ritu & Rathod, Nikita. (2023). DESIGNING A SOFTWARE TOOL SET FOR STORING DATA IN DECENTRALIZED STORAGE SYSTEM. 10.13140/RG.2.2.20394.18883.

[7] Singh, Apoorv, Chandraket Singh, and Samridhi Mishra. "Enhanced Honey Encryption Algorithm on e-mail with Increased Message Space." Int. J. Res. Eng. Sci. Manag 3 (2020): 453-456.

[8] Singh, Apoorv, Chandraket Singh, and Samridhi Mishra. "Enhanced Honey Encryption Algorithm on e-mail with Increased Message Space." Int. J. Res. Eng. Sci. Manag 3 (2020): 453-456.

[9] Imfeld, Raphael. "Increasing Privacy in Smart Contracts: Exploring Encryption Mechanisms."

[10] Elminaam, Diaa Salama Abdul, Hatem Mohamed Abdul Kader, and Mohie Mohamed Hadhoud. "Performance evaluation of symmetric encryption algorithms." IJCSNS International Journal of Computer Science and Network Security 8.12 (2008): 280-286.