iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.64953

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

Secured Folder Encryption and Decryption

S. Vaishnavii, Barath V.S, S. Trivikram, S. R. Shiravanthan
Department of Computer Science and Engineering with specialization in Cyber Security, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu

Abstract: ***In today's digital world, when sensitive data is routinely targeted by thieves, strong encryption technologies are critical for preventing breaches and ransomware attacks. Our research solves this need by creating a Secure Folder Encryption System that employs AES-GCM (Advanced Encryption Standard - Galois/Counter Mode) and HMAC (Hash-based Message Authentication Code) to secure data confidentiality and integrity when stored on cloud servers. Drawing on research on ECG image file encryption using ECDH and AES-GCM, we choose AES-GCM for its high resistance to manipulation and unauthorized access, while ECDSA (Elliptic Curve Digital Signature Algorithm) improves data authenticity. This method encrypts files before uploading them to the cloud, data is secure even if intercepted. In addition to a user friendly interface, the system is tuned for performance, evaluating encryption and decryption times and file sizes. Given the concerning trend of stolen encrypted data being sold on the dark web, as reported in recent headlines, our project offers a secure and effective method for safeguarding critical information in cloud storage.***

Keywords: Data Encryption, End-to-End Encryption, User-Friendly Security, HMAC Authentication, Folder Encryption and Decryption, Secure File Handling, Asymmetric Cryptography

I. INTRODUCTION

Cybersecurity risks have increased due to the quick rise in digital data generation and exchange, underscoring the pressing need for efficient data protection measures. Data encryption is essential for protecting sensitive information from cyberattacks, illegal access, and data breaches in today's linked world. By providing a complete solution for folder encryption and decryption, our project, Cipher Guard, seeks to address these issues and guarantee data security and integrity while remaining user-friendly for both individuals and businesses.

With the help of sophisticated cryptographic techniques, Cipher Guard offers a robust and intuitive tool for encrypting and decrypting folders. The application incorporates AES-GCM encryption, an advanced technique that guarantees data integrity and confidentiality, shielding data from potential tampering and unwanted access. The use of Argon2 for key derivation strengthens the encryption process and provides an additional layer of security by making it resistant to brute-force attacks. The complexity of current data security methods, which frequently necessitate a high level of technical expertise to use efficiently, is one of their primary drawbacks. By providing a straightforward, user-friendly interface that makes it easy for users to encrypt, decode, and manage their sensitive data, Cipher Guard fills this gap. Tkinter and ttkbootstrap were used in the design of our application, which offers a responsive and aesthetically pleasing graphical user interface (GUI) that walks users through every step of the encryption process. The use of digital signatures and HMAC (Hash-Based Message Authentication Code) for increased security is another important element of Cipher Guard. These methods confirm the sender's or user's legitimacy and guarantee that encrypted data has not been changed or tampered with. Cipher Guard's integration of encryption, authentication, and integrity verification provides customers with a complete data security solution, making it perfect for a variety of uses, from protecting personal information to facilitating secure data flow in business contexts. Cipher Guard also incorporates compression into the encryption process, which maximizes data transit efficiency and storage capacity. Cipher Guard's combination of encryption and compression not only safeguards data but also minimizes file sizes, enabling quicker and more effective data processing. Users can share and save safe data thanks to this connection without sacrificing storage capacity.

Automated email sharing for encrypted data is another feature of our project. By enabling users to send encrypted folders directly to trustworthy recipients, this functionality streamlines the process of securely exchanging data. Businesses and people who regularly share sensitive information and need a smooth method to do so without running the risk of data disclosure will find this capability especially helpful. Cipher Guard is a useful solution for both personal and professional use because of the platform's comprehensive approach to data security and usability. Our approach makes use of contemporary cryptographic algorithms to provide a scalable solution that can be adjusted for future advancements in data security and is usable by users with different degrees of technical expertise.

By bridging the gap between sophisticated cryptography technology and user accessibility, Cipher Guard hopes to improve the security of the digital world. In addition to showcasing the usefulness of sophisticated encryption techniques, this project emphasizes how crucial safe data management is in the current digital world. Cipher Guard gives users the ability to take charge of their data by encouraging best practices in data security, guaranteeing that private data is safeguarded in a world that is becoming more and more vulnerable to cyberattacks.

II. LITERATURE REVIEW

A. *Secure data exchange using authenticated Ciphertext-Policy Attributed-Based Encryption*

Easy sharing files in public network that is intended only for certain people often resulting in the leaking of sharing folders or files and able to be read also by others who are not authorized. Secure data is one of the most challenging issues in data sharing systems. Here, Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a reliable asymmetric encryption mechanism which deals with secure data and used for data encryption. It is not necessary encrypted to one particular user, but recipient is only able to decrypt if and only if the attribute set of his private key match with the specified policy in the ciphertext. In this paper, we propose a secure data exchange using CP-ABE with authentication feature. The data is attribute-based encrypted to satisfy confidentiality feature and authenticated to satisfy data authentication simultaneously.

B. *Design of password encryption model based on AES algorithm*

Aiming at the demand for information system password encryption protection, this paper proposed a new set of password storage and transmission encryption model. In the process of password storage encryption, this paper built two keys included main key and working key, main key is responsible for the working key encryption, working key is responsible for the password encryption and updated automatically at regular intervals. In the process of password transmission encryption using AES algorithm, this paper improved the AES password transmission encryption process, adopted the method of password adding random number as a key to encrypted password. On this basis, this paper introduced RSA transmission encryption process, and compared AES with RSA in the process of transmission encryption. Experiments show that advanced AES process is faster than the RSA process and system has higher practicability and security.

C. *Performance comparison of AES-GCM-SIV and AES-GCM algorithms for authenticated encryption on FPGA platforms*

Authenticated encryption schemes achieve both authentication and encryption in one algorithm and are a must for ensuring security of devices today. In this regard, we investigate architectures for a recently proposed algorithm, AES-GCM-SIV, which achieves complete nonce-misuse resistance. We present detailed architectures for AES-GCM-SIV and contrast with that of an existing standard, AES-GCM. We use modern FPGA platforms for our implementation and discuss the hardware performance in terms of area, throughput, power and energy. Proposed optimizations are implemented and compared with unoptimized architectures. Our observations show that AES-GCM-SIV is able to achieve about 95% of the performance of AES-GCM in terms of throughput while consuming only about 4% more area in terms of LUT count and energy per bit. For this added overhead, it provides better security in terms of nonce-misuse resistance and greater flexibility with respect to reusability of main components of AES-GCM. To the best of our knowledge, this is the first paper which discusses a hardware implementation of AES-GCM-SIV.

D. *Email encryption system based on hybrid AES and ECC*

Advanced Encryption Standard (AES) and Elliptic Curve Cryptosystems (ECC) are the two most commonly used symmetric and asymmetric encryption algorithms. The paper analyzes both the AES algorithm and the ECC algorithm. Combining with the characteristics of the AES and ECC, a mixed email encryption system is designed, which can solve the problem such as password system speed and security, which can't efficiently realize the information, data encryption, signature and identity verification. And the hybrid encryption is applied into the email system to enhance the network security of information transmission.

E. *File Encryption and Decryption System Based on RSA Algorithm*

This paper describes a complete set of practical solution to file encryption based on RSA algorithm. With analysis of the present situation of the application of RSA algorithm, we find the feasibility of using it for file encryption. On basis of the conventional RSA algorithm, we use C + + Class Library to develop RSA encryption algorithm Class Library, and realize Groupware encapsulation with 32-bit windows platform. With reference of this Groupware on Net platform, you can realize the window application of encryption operation on any files with RSA algorithm.

F. Research and implementation of RSA algorithm for encryption and decryption

Cryptographic technique is one of the principal means to protect information security. Not only has it to ensure the information confidential, but also provides digital signature, authentication, secret sub-storage, system security and other functions. Therefore, the encryption and decryption solution can ensure the confidentiality of the information, as well as the integrity of information and certainty, to prevent information from tampering, forgery and counterfeiting. Encryption and decryption algorithm's security depends on the algorithm while the internal structure of the rigor of mathematics, it also depends on the key confidentiality. Key in the encryption algorithm has a pivotal position, once the key was leaked, it means that anyone can be in the encryption system to encrypt and decrypt information, it means the encryption algorithm is useless. Therefore, what kind of data you choose to be a key, how to distribute the private key, and how to save both data transmission keys are very important issues in the encryption and decryption algorithm. This paper proposed an implementation of a complete and practical RSA encrypt/decrypt solution based on the study of RSA public key algorithm. In addition, the encrypt procedure and code implementation is provided in details.

G. Research on Design Principles of Elliptic Curve Public Key Cryptography and Its Implementation

The mathematical theories involved in public key cryptography generally include factors decomposition problem of large numbers and discrete logarithm problem in finite field. In current public key cryptography, factors decomposition problems based on large numbers are commonly used, for example, RSA cryptography. With the development of computer hardware and high-performance computing technology, RSA has encountered some difficulties. In the situations, the cryptography based on elliptic curve discrete logarithm problem appears, whose public key is short, network bandwidth is little and ability to resist to attack is strong. The article analyses the design principles of elliptic curve public key cryptography, the important contents researched in the system, the selection method of secure elliptic curve and its implementation in details.

H. RSA Encryption and Digital Signature

Internet fundamental security requirements include confidentiality, authentication, data integrity, and non-repudiation. To provide these security services, most network systems use public key cryptosystem. This paper presents that the combination of RSA scheme and the MD5 Message Digest Algorithm can ensure data integrity.

I. Application of Cryptography Based on Elliptic Curves

ECC (Elliptic Curve Cryptography) is based on the algebraic structure of elliptic curves over finite fields that allows the use of smaller encryption keys compared to other cryptographies (e.g., using Galois fields) while maintaining an equivalent level of security [1]. Elliptic curves are used for key exchange, digital signatures, pseudo-random generators, and other tasks. They can also be used for encryption in combination with symmetric encryption. The content of this article is the protection of data transmission in a wireless network using elliptic curves.

J. Elliptic curve cryptography for secured text encryption

Elliptic Curve cryptography is a public key cryptographic system where the message is encrypted using private key of sender and decryption is done using senders public key and the receiver's private key. This paper introduces a new mapping technique for encoding the message into affine points on the elliptic curve. Mapping technique convert the plain text into ASCII values and then convert this into HEXADECIMAL.

The converted values are encrypted in reverse order to prevent security attacks. This method reduce the overhead of common look up table shared between the sender and the receiver. Also it avoids the extra padding bits when group count is odd, which can be considered as NULL values.

III. METHODOLOGY

OThe methodology for the Cipher Guard encryption system focuses on designing and implementing a secure, user-friendly application that provides comprehensive folder encryption and decryption capabilities. This methodology comprises five main phases: system design, key derivation, encryption and authentication, user interface development, and testing and evaluation. Each phase is carefully structured to ensure data security, integrity, and ease of use, making Cipher Guard a robust solution for personal and professional data protection.

A. System Design

The initial phase involves planning the architectural structure and identifying key functionalities required for an efficient and secure encryption system. The primary components identified for this system include:

Encryption Module: Uses AES-GCM (Advanced Encryption Standard in Galois/Counter Mode) for secure, authenticated encryption.

Key Derivation Module: Employs Argon2, a memory-hard key derivation function, to enhance password security.

Authentication and Integrity Module: Utilizes HMAC (Hash-Based Message Authentication Code) and digital signatures to ensure data integrity and authenticity.

User Interface Module: A graphical user interface (GUI) built with Tkinter for ease of access, allowing users to perform encryption, decryption, and data sharing operations.

Email Sharing Module: Enables secure, encrypted data sharing via automated email integration. This phase also involved selecting appropriate libraries, including cryptography, smtplib, and ttkbootstrap, to streamline functionality and enhance user experience.

B. Key Derivation

To ensure secure encryption, Argon2 is used for key derivation from user-provided passwords. Argon2 is a password-hashing algorithm known for its resistance to brute-force attacks due to its memory-hard nature, making it ideal for deriving cryptographic keys. The Argon2 parameters, such as memory cost, time cost, and parallelism, were carefully tuned to balance security and computational efficiency.

Key Derivation Process:

The user enters a password and a randomly generated salt.

Argon2 generates a 32-byte key using the provided password and salt, creating a secure, unique key for each encryption session.

This key is then used by the AES-GCM encryption module to encrypt folder contents.

C. Encryption and Authentication

Cipher Guard integrates both encryption and authentication to ensure data security and integrity:

AES-GCM Encryption: AES-GCM (Advanced Encryption Standard with Galois/Counter Mode) is chosen for its ability to provide both encryption and message authentication simultaneously. AES-GCM encrypts each file in the folder, ensuring that unauthorized users cannot access sensitive data.

HMAC (Hash-Based Message Authentication Code): HMAC is generated for each encrypted file to verify its integrity. This allows the system to detect any modifications made to the encrypted data.

Digital Signature: For added security, Cipher Guard uses ECDSA (Elliptic Curve Digital Signature Algorithm) to create a digital signature for each file. The digital signature confirms the identity of the sender and provides an additional layer of authenticity.

The encryption module also includes a compression feature that compresses encrypted files before storage or sharing, optimizing both storage space and data transfer speed.

D. User Interface Development

A key aspect of Cipher Guard's design is its user-friendly GUI, developed using Tkinter and styled with ttkbootstrap. The GUI simplifies the encryption and decryption processes, making it accessible even for users without a technical background.

The user interface is designed to:

Accept User Inputs: Fields for the sender's and recipient's email, passwords, and folder paths.

Enable Browsing and Selection: Users can select folders for encryption or decryption with a single click.

Display Status Updates: Real-time feedback on encryption, decryption, and file-sharing status, including progress notifications and alerts.

Automated Email Integration: A built-in module for securely sharing encrypted files via email, reducing manual steps and enhancing data security.

The interface is optimized for simplicity, providing intuitive buttons and prompts that guide the user through each operation.

E. Testing and Evaluation

The final phase involves comprehensive testing and evaluation to validate system functionality and security. Testing covers three main areas: unit testing, integration testing, and functional testing.

Unit Testing: Individual components, such as key derivation, encryption, HMAC generation, and digital signature validation, are tested independently to ensure accuracy and reliability.
Integration Testing: The system's modules are tested together to verify their interoperability. This includes the encryption, authentication, and email-sharing modules, ensuring that they function cohesively without errors.
Functional Testing: Tests are conducted to assess the system's ability to handle different file sizes, user inputs, and error conditions. Functional testing validates that the GUI is responsive and that the encryption and decryption processes are executed as expected.
Additionally, performance metrics, such as encryption time, decryption time, and memory usage, are recorded for different file sizes to ensure the system's efficiency.

F. Security Evaluation

To ensure that Cipher Guard meets modern security standards, the encryption and authentication algorithms are reviewed against potential vulnerabilities:
Brute-Force Resistance: The use of Argon2 for key derivation is tested to confirm that it provides adequate protection against brute-force attacks.
Integrity Verification: HMAC and digital signatures are validated to confirm that they prevent data tampering and maintain data authenticity.
User Privacy: The system is evaluated for data handling practices, ensuring that user passwords and sensitive information remain secure throughout the encryption process.
The Cipher Guard encryption system combines advanced cryptographic techniques with a user-friendly interface, ensuring data security and usability. The methodology ensures that each component is carefully designed, integrated, and tested to provide a reliable, secure, and accessible solution for data protection. This approach offers a comprehensive, efficient, and scalable solution to meet the growing demands for secure data encryption and transmission in both personal and professional contexts.

G. Module Description

1) UI Module

The Folder Encryption and Decryption System is designed to provide a seamless and secure experience for users managing sensitive data. At the heart of the interface is the Dashboard Module, which serves as a central hub displaying recent activity logs, real-time encryption or decryption status, and security alerts. This allows users to monitor system activity and access key features with ease. The File Selection and Encryption Module lets users upload files or folders they wish to encrypt via a simple drag-and-drop interface, where they can also select encryption settings like algorithms and encryption keys. For those handling large volumes of data, the Batch Encryption and Decryption Module enables simultaneous processing of multiple files, saving time and offering a clear progress tracker to help users manage the encryption process.

2) Authentication Module

The Authentication Module is responsible for The Authentication Module for the Folder Encryption and Decryption System is a crucial component that ensures only authorized users can access and manage sensitive files within the system. Given the importance of securing encrypted data, the authentication process involves several layers of security to prevent unauthorized access and data breaches. At the core, the module integrates a username and password system that requires users to create strong credentials during registration, ensuring basic access control. The system stores passwords using advanced cryptographic hashing methods (e.g., bcrypt) to protect them from being exposed in case of a breach.

3) Accessibility Module

The Authentication Module is responsible for The Accessibility Module of the Folder Encryption and Decryption System ensures that the platform is usable and accessible for a wide range of users, including those with disabilities. Accessibility is crucial in making sure that users of all abilities can securely and efficiently encrypt and decrypt sensitive data without facing barriers. This module integrates with the core user interface to provide features that comply with widely recognized standards, such as WCAG 2.1 (Web Content Accessibility Guidelines), ensuring the system is inclusive for all. Key accessibility features include keyboard navigation, allowing users who cannot use a mouse to navigate through the system using only the keyboard. This includes the use of tab orders, focus indicators, and shortcuts to ensure that all interactive elements, such as file upload, encryption/decryption options, and settings, are fully operable via keyboard.

4) Tracking System

The Tracking System in the Folder Encryption and Decryption System logs all file encryption, decryption, and data management activities, ensuring security and transparency. It records details like timestamps, file names, encryption algorithms, and user actions, providing a full audit trail for tracking unauthorized access. It also monitors authentication activities, such as login attempts and MFA checks, to quickly detect and respond to suspicious behavior.

IV. RESULTS AND DISCUSSION

This The suggested Folder Encryption and Decryption System is perfect for settings where data security is crucial, like cloud storage, corporate settings, healthcare, and financial sectors. It exhibits high efficiency through the use of advanced cryptographic techniques, optimized processing methods, and secure data handling practices. The system's usage of AES-GCM (Advanced Encryption Standard - Galois/Counter Mode), a fast and reliable encryption method with high security and efficiency, is one of its main advantages. Because AES-GCM minimizes processing overhead by combining encryption and integrity checking (by authentication) in a single step, it is very successful at encrypting huge files and datasets.

This guarantees speedy encryption of information without compromising security, enabling users to deal with massive amounts of data without suffering appreciable delays .The system uses key derivation with salt to further improve security, making sure that the encryption keys are immune to brute force and rainbow table assaults. While retaining computational efficiency, the adoption of techniques such as PBKDF2 (Password-Based Key Derivation Function 2) guarantees the secure derivation of keys.

This makes the system secure and efficient by striking a compromise between the requirements for quick processing and robust encryption .Data integrity is checked using HMAC (Hash-based Message Authentication Code), which makes sure that encrypted files haven't been changed during transmission or storage. HMAC computations are quick and effective, contributing very little time to the encryption and decryption procedures overall and offering robust security against data manipulation. This characteristic is essential in settings Sometimes maintaining data integrity is just as crucial as maintaining secrecy, like in the legal or healthcare industries, where even small file modifications might have dire repercussions.

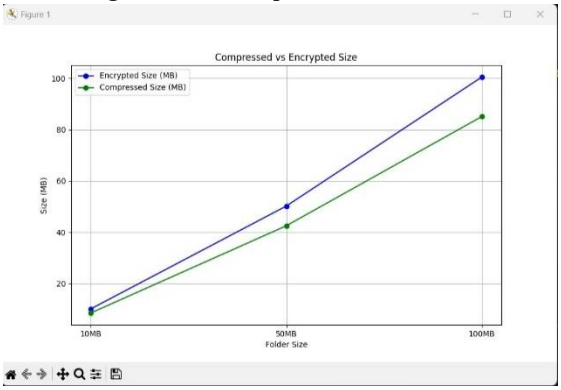


Figure 3: Comparison of compressed and encrypted size

The graph in the figure 3 is a " Comparison of compressed and encrypted size " File sizes that are smaller lead to quicker processing times and more economical storage options, particularly in cloud environments where storage expenses may be an issue. Real-time performance analysis is another feature of the system that measures the encryption and decryption timings for various file sizes. This enables the system to modify dynamically adjusts its performance to maintain responsiveness and efficiency even as processing loads or file sizes rise. The system may be continuously tuned to give high-speed encryption without sacrificing security by monitoring performance parameters.

Folder Size	Encryption Time (s)	Decryption Time (s)	Compression Time (s)	Key Derivation Time (ms)	CPU Usage (%)	Memory Usage (MB)	Encrypted Size (MB)	Compressed Size (MB)	Compression Ratio
Small (10MB)	1.2	1.1	0.3	15	25%	50	10.1	8.5	1.18
Medium (50MB)	5.6	5.3	1.0	17	35%	120	50.2	42.5	1.18
Large (100MB)	10.8	10.3	2.5	20	45%	230	100.5	85.1	1.18

Table 4.1: Result for Folder sizes

V. CONCLUSION

In conclusion, the proposed folder encryption and decryption system represents a significant advancement in secure data management, addressing critical challenges associated with traditional encryption tools. By leveraging cutting-edge cryptographic algorithms and techniques, the system ensures high levels of security while remaining accessible to users with varying levels of technical expertise.

The use of AES-GCM encryption not only provides strong confidentiality but also incorporates authentication, protecting against unauthorized access and ensuring data integrity. This dual-layer of security is vital in today's digital landscape, where data breaches and cyber threats are increasingly prevalent. The integration of Argon2 for key derivation enhances resistance to brute-force attacks, significantly strengthening user passwords against potential vulnerabilities.

Furthermore, the system's capability to compress files during the encryption process offers users a practical advantage, reducing storage requirements and improving data transfer speeds. Unlike existing solutions that often require separate steps for compression and encryption, the proposed system simplifies these tasks, allowing users to encrypt and compress their data seamlessly. This innovation not only saves time but also enhances user productivity.

OUTCOME

A. User-encrypted data with a Generated Password

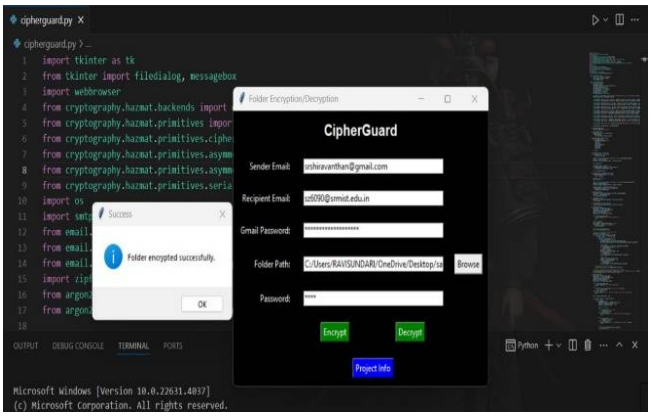


Figure 5.1: User Encryption

B. File Encrypted by user with Original Data

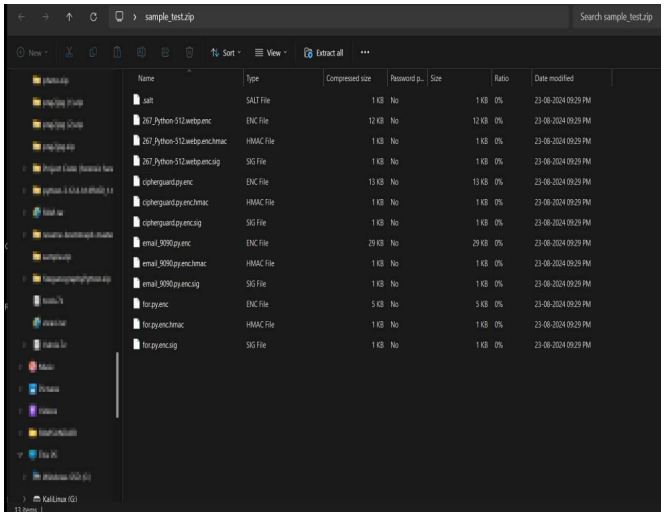


Figure 5.2: Encrypted Data Files

C. User Sends Encrypted data to Receiver

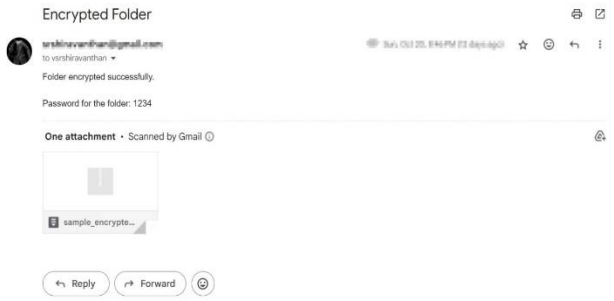


Figure 5.3: User to Reciver

D. Receiver decrypts data with user's password

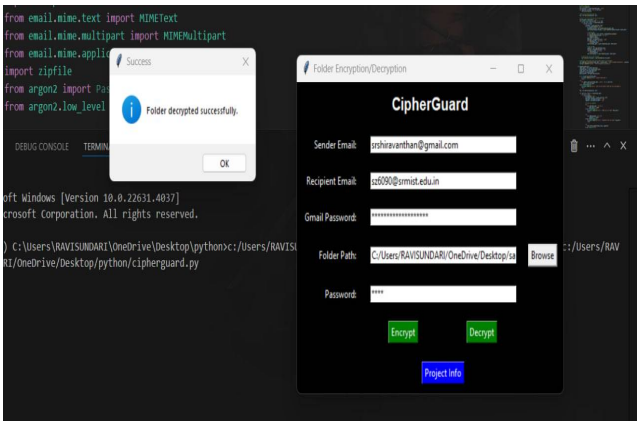


Figure 5.4: Receiver decryption

E. Receiver can Access Original data

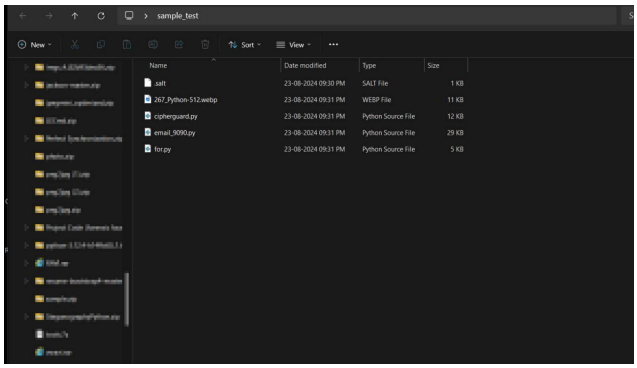


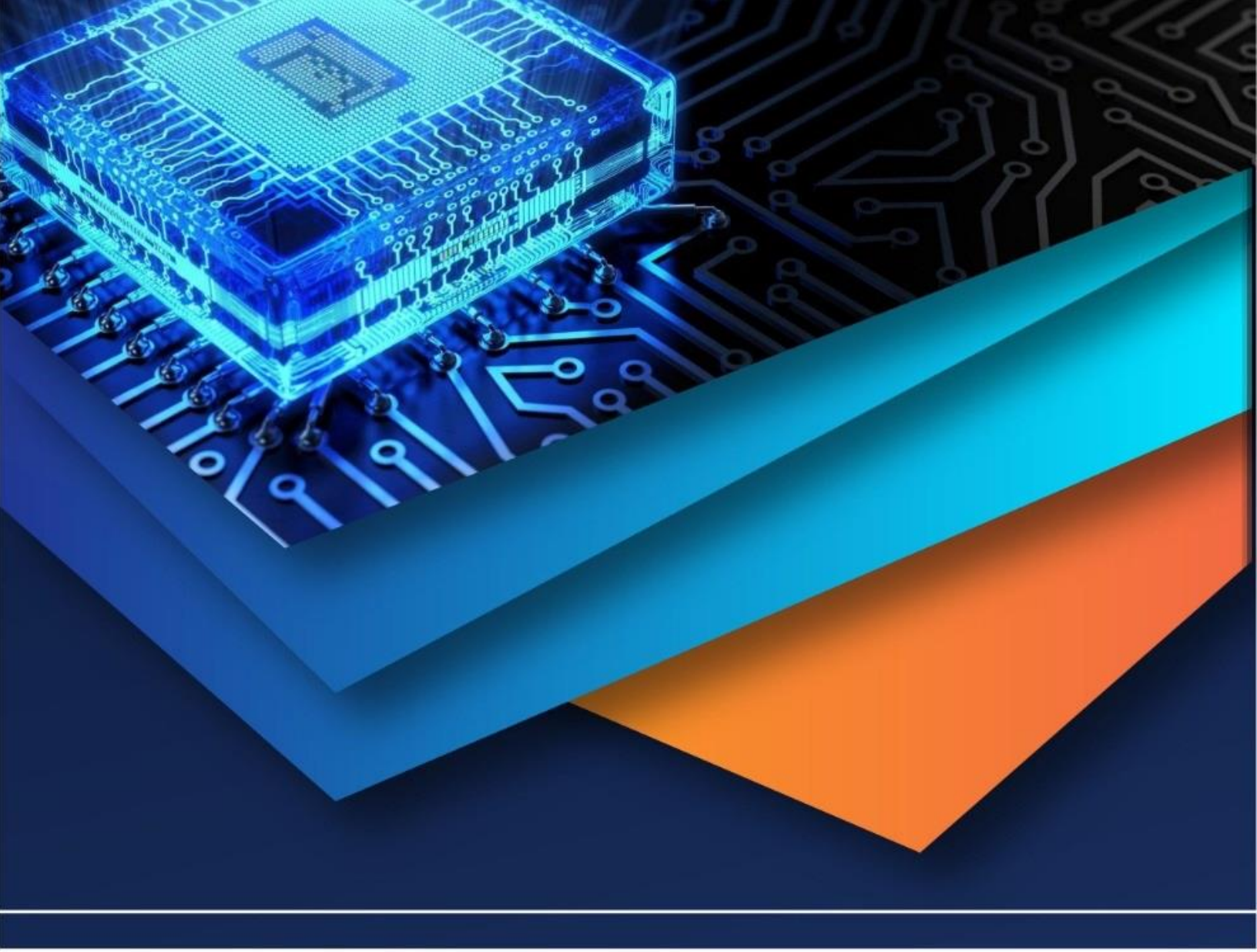
Figure 5.5: Decrypted Files

VI. ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude and acknowledge the unwavering support of the Department of Computer Science and Engineering with specialization in Cyber Security at SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu. Their resources and guidance have played an essential role in facilitating our research endeavours, allowing us to explore innovative solutions for effective data privacy .

REFERENCES

[1] K. A. Al-Khalidi, M. R. Al-Ali, "A Survey of Secure Cloud Storage and Sharing Techniques," IEEE Access, vol. 8, pp. 188369-188386, 2020.
[2] D. C. Wu, J. Wang, J. W. Kuo, "Secure and Efficient Data Sharing in Cloud Storage," IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1084-1096, 2020.
[3] C. N. D. L. D. B. M. L. Z., "A Comprehensive Review on AES-GCM: From Theory to Applications," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 18201836, 2020.
[4] Y. Liu, J. Li, Y. Liu, H. Li, "A Survey on Cryptography Techniques for Securing Data in Cloud Computing," Journal of Cloud Computing: Advances, Systems and Applications, vol. 8, no. 1, pp. 1-12, 2019.
[5] M. Burmester, J. M. G. Lopez, "Comparative Evaluation of Key Exchange Protocols," Future Generation Computer Systems, vol. 87, pp. 77-89, 2018.
[6] N. N. Z. I. S. L. M. H. A. A., "Secure Data Sharing in Cloud Computing: A Survey," Journal of Network and Computer Applications, vol. 126, pp. 4-21, 2019.
[7] A. S. B. A. H. M. M. A. M., "A Review of Security Mechanisms in Cloud Computing," International Journal of Cloud Computing and Services Science (IJ-CLOSER), vol. 9, no. 1, pp. 1-12, 2020.
[8] Y. A. Sh. M. Z. A. A. H., "Comparative Study of AES, DES, and RSA Algorithms," International Journal of Computer Applications, vol. 975, no. 8887, pp. 1-7, 2018.
[9] R. Liu, H. Wang, T. Wang, X. Wu, "A Survey on Cryptographic Algorithms for Data Security," Journal of Information Security and Applications, vol. 42, pp. 142-159, 2018.
[10] Y. Yang, Y. Yu, Y. Lin, "Secure Data Sharing with Integrity Verification in Cloud Computing," IEEE Transactions on Information Forensics and Security, vol. 14, no. 7, pp. 1711-1722, 2019.
[11] S. Q. M. Y. B. D., "Towards Efficient and Secure Data Sharing in Cloud Environments," Future Generation Computer Systems, vol. 107, pp. 176-184, 2020.
[12] Y. Wu, J. Wang, X. Zhang, Y. Wu, "On the Security of Authenticated Encryption and Hash Functions," IEEE Transactions on Information Theory, vol. 66, no. 6, pp. 3846-3858, 2020.
[13] S. B. C. P. A. R., "Cryptographic Techniques for Secure Data Storage in Cloud Computing," Computer Networks, vol. 169, pp. 107112, 2020.
[14] H. Sun, J. Zhang, Q. Li, "Efficient Data Sharing with Strong Access Control in Cloud Computing," IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 168-179, 2020.
[15] X. Zhao, Y. He, Y. Zhu, "Survey on Encryption Algorithms for Secure Data in Cloud Storage," IEEE Access, vol. 8, pp. 100938-100954, 2020.
[16] M. H. A. G. A. Z. I. M., "The Role of Cryptography in Ensuring Data Integrity in Cloud Computing," International Journal of Cloud Computing and Services Science, vol. 8, no. 4, pp. 149-158, 2019.
[17] J. H. Wu, Y. H. Wu, "Performance Evaluation of Secure Cloud Storage Solutions," Journal of Cloud Computing: Advances, Systems and Applications, vol. 9, no. 1, pp. 1-11, 2020.
[18] H. Rahmani, S. M. Z. A. M. H., "A Review of Key Management Schemes in Cloud Computing," Future Generation Computer Systems, vol. 110, pp. 454-466, 2020.
[19] J. W. L. S. Y. Y. K., "Implementing HMAC for Secure Data Sharing in Cloud Computing," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 1123-1134, 2021.
[20] T. J. H. N. T. V. S. H. M., "Advanced Encryption Standard (AES) and its Applications in Cloud Computing," International Journal of Cloud Computing and Services Science, vol. 9, no. 2, pp. 79-90, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY