SECURIFY – A Hybrid Cryptography Framework for Confidential and Tamper-Proof Data Security

Prof. Abhijeet More
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
abhijeetdmore242@gmail.com

Prof. Tejashree Patil
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
ptejashree2024@gmail.com

Vaibhavi Deshmukh
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
vaibhavideshmukh31@gmail.com

Pratik Patil
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
pratikvpatil0021@gmail.com

Khushboo Verma
Dept. Of Computer Application
Pillai HOC College of Engineering and Technology
(Mumbai University)
Rasayani
Maharshtra,India
khushbooverma525@gmail.com

Abstract- **Nowadays, organizations have radically changed their approach to handling digital data. Today, sensitive data such as financial records, personal information, and corporate documents are being stored and shared through various online platforms. We found in the study that even though encryption such as AES and RSA methods are extensively implemented, encryption alone is not always sufficient to solve real security problems, especially in multiuser systems with shared files. Many times, the lack of an approval system and action tracking leads to a higher risk of improper use. We thought up of SECURIFY, a secure file management system that integrates cryptographic measures with a controlled access procedure in order to mitigate this problem. SECURIFY is a secure file management system that integrates cryptographic measures with a controlled access procedure to help mitigate this problem. Every file stored is encrypted with AES, GCM, and to safeguard the files encryption key, RSA key wrapping is used. To further protect the integrity of the stored data, we use HMAC, SHA256 to provide tamper detection. On the contrary, the major feature of the system is that it supports a request permission workflow mechanism. A user cannot decrypt a file at his/her own discretion; the file owner or the administrator must first grant permission. The system records in audit logs all the actions and changes that are carried out in the framework. The framework was developed in Python and a Streamlit user interface was added to make the framework both very simple and practical. During the evaluation, we found that the system could handle encryption and decryption very efficiently.**

Keywords: Hybrid Cryptography, AES, RSA, HMAC, Secure Vault, Access Control, Data Integrity.

I. Introduction

Organizations handle an immense amount of sensitive financial records, strategic assets, and personally identifiable information (PII) [1]. Where there are walls, there are doors. Cyber attackers have escalated their ransomware, insider threats, and data manipulation methods to outsmart the traditional security defences [1], [3]. Although Standard symmetric encryption algorithms such as AES are used to maintain the secrecy of the data, however, they are not enough to handle the multi aspect and complex modern cyber-attack scenarios [2], [9].

Encryption safeguards the data while it is being transmitted [2]. However, it is unable to prevent changes of the files [5]. Furthermore, if the keys are leaked, the encryption cannot decide who gets to decrypt the data [2], [5]. Confidentiality, without integrity and governance, leaves a significant security gap [8]. SECURIFY closes this circle. We created a hybrid system that combines the advantages of fast AES file encryption with RSA asymmetric key encapsulation [2], [9]. In order to completely remove the risk of silent tampering, the system verifies the integrity based on HMAC [5], [10]. Anytime someone changes the content without permission, it gets detected immediately [5]. But mere cryptography does not cut it [8]. In contrast to other tools that only concentrate on simply hiding the content, SECURIFY requires a very strict and technologically controlled access mechanism through workflows [8]. No one user is allowed to decrypt a file unless an administrator has explicitly given the authorization. With this governance model, the administrator is held accountable. The complete audit trail is generated with each user action such as uploads, access requests, approvals, and rejections [6], [8]. The logs serve as a tamper, evident record, which can be used for forensic investigations [6]. This paper explains the SECURIFY system design and the way that it operates, showing that combining process, based

governance with hybrid cryptography effectively defends against highly advanced attacks [2], [8]. We have tried to show that encryption combined with the oversight of administrators is a very effective tool.

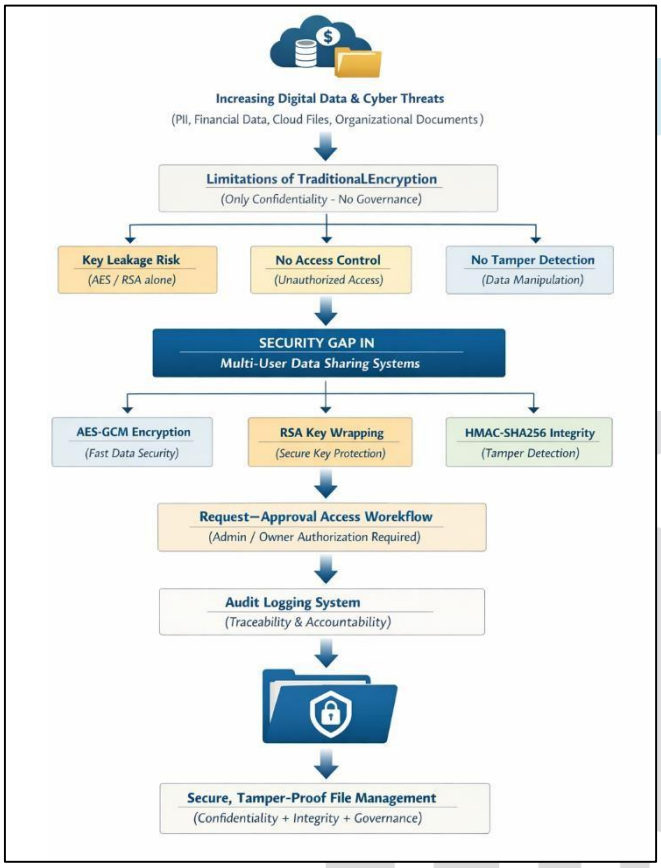

Figure 1.1: Overview of SecurifyFramewrok

II. Background

The digital world today witnesses a tremendous increase in data generation as the expansion of online systems, cloud services, and network applications is rapid [1], [2]. Digital transmission and storage methods are used to transfer sensitive information which includes personal records and financial details and healthcare data and organizational documents [1]. Digital transformation has improved operational efficiency and user accessibility but it creates major security threats which include unauthorized access and data breaches and harmful data alterations [3], [5]. Basic encryption techniques which traditional security methods use do not provide adequate protection for data in multi-user systems which require both access control and integrity verification and accountability functions [2], [8].

Hybrid cryptography has become an indispensable technology as it integrates symmetric encryption and asymmetric encryption to build secure systems [2], [9]. The symmetric algorithms are able to encrypt large data quickly but the asymmetric algorithms are used to create secure ways for key exchange and user authentication [2]. Protection of data confidentiality is more about the combination of integrity verification and access control systems rather than just encryption [5], [8]. The SECURIFY framework, which is presented as a solution, will be able to deal with these difficulties by hybrid encryption, integrity checking, access approval, and audit logging being combined into one system [6], [8]. Such a method keeps the data secret, honest, and traceable at all times, thus, if we talk about the data lifecycle, it is, therefore, very suitable for the current secure data management applications [5], [9].

III. Motivation

The growth in data leaks, cyberattacks, and the leaking of private information has unveiled the weakness of traditional data protection systems. Even widely used encryptions such as AES and RSA cannot be seen as the ultimate means of data protection since these tools are not entirely secure. Therefore, they lack access control, ownership tracking, and activity monitoring features. Moreover, data is shared between various stakeholders, e.g., cloud storage platforms, educational systems, and corporate networks, as these environments do not have adequate permission processes and supervision mechanisms. The lack of accountability mechanisms leads to a major security flaw that enables illegal access to sensitive information without any unauthorized persons being able to find out.

SECURIFY framework has been created to give users a comprehensive security solution that is capable of going beyond the standard encryption capabilities. The scheme offers hybrid cryptography in combination with integrity verification and access request, approval workflows and audit logging to safeguard data confidentiality, prevent data from being tampered with and provide traceability. The project exists because there is a need for a security framework which both technical and non-technical users can implement while it provides strong cryptographic protection. The development team wants to establish trust in digital data-sharing systems through SECURIFY by using transparent systems and accountable processes and restricted access methods.

III. Literature Review

“Enhancing Secure Communication in Industry 4.0 using Digital Signatures and Two-Level Encryption” by Singh et al. [1] designed a two layer encryption system for Industry 4. 0 settings that combines digital signatures with symmetric encryption to guarantee safe communication between industrial machines. Mainly, their method concentrated on securing data transmission through authentication verification and message tampering prevention. Although the communication security aspect of their solution was robust, it lacked measures for secure storage or access control of stored files.

“RSA and AES Based Hybrid Encryption Technique for Cloud Computing” by Mehta et al. [2] came up with a hybrid cryptographic scheme that uses AES for data encryption and RSA for secure key exchange in cloud computing systems. Through the synergy of symmetric and asymmetric cryptography, their system was able to elevate confidentiality and performance for extensive data. Nevertheless, their system did not have a well, defined multi, user access control and lacked logging and governance features, thereby making it less appropriate for collaborative environments.

"ES-SECS/GEM: An Efficient Security Mechanism for SECS/GEM Communication" by Chen et al. [3] employed the SECS/GEM protocol to develop encryption and replay, attack prevention techniques for securing industrial communication systems. Their minimalist mechanism was able to increase the reliability of network communication; however, it neither offered protection to the stored files nor did it include the implementation of user, level authorization controls.

"Hybrid Cryptography Scheme for Secured Data Transmission" by Patel et al. [4] put forward a dual, layer hybrid encryption system that was designed to safeguard the information being transmitted in IoT networks. Although the platform enhanced the security of transmissions, it did not include features such as audit logging or approval processes based on workflows.

"A secure file storage system utilizing SHA-256 along with a hybrid AES–RSA approach" by Yadav et al. [5] combined AES encryption, RSA key protection and SHA-256 hashing to deliver confidentiality and integrity. Their method checked file integrity at the time of retrieval; nonetheless, it was mostly targeting single, user environments and was not equipped with structured access governance.

"Blockchain-Enabled Data Security Framework for Cloud Environment" by Park et al. [6] investigated blockchain incorporation in cloud security to generate tamper, proof audit trails. Even though blockchain brought transparency and immutability, the solution resulted in computational overhead and complexity of implementation.

"ECC and AES Based Hybrid Encryption for IoT Healthcare" by Gupta et al. [7] leveraged ECC with AES for smart healthcare IoT devices to diminish the key size while still having robust encryption. As a matter of fact, it was very efficient with limited devices; however, the framework was very much specific to the domain and was not equipped with centralized multi, user management.

"Attribute-Based Encryption for Secure Cloud Data Sharing" by Dutta et al. [8] suggested using attribute, based encryption (ABE) to allow precise access control in cloud data sharing. ABE is undoubtedly a very interesting tool in providing authorization based on user's characteristics; however, on the other hand, the complications in implementation and key management issues hindered its scalability.

"Enhanced Hybrid Security Model Using AES, RSA, and SHA-512" by Roy et al. [9] intensified hybrid encryption scheme by combining AES, RSA, and SHA, 512 to reach a high standard of confidentiality and integrity. This model delivers a very robust, multi, layer security and efficient data protection. Nevertheless, the major focus of the model is on cryptographic strength, and hence, it lacks certain features such as workflow, based access control, audit logging, and governance, which makes it less appropriate for secure multi, user environments.

"Two-Phase Hybrid Encryption and Integrity Verification Framework" by Kaur et al. [10] presented a two step process of encrypting and verifying data reliability. Although they have offered multi layered protection, their methods did not incorporate formal request-permission workflows and centralized auditing mechanisms.

The review of the literature shows that the majority of current solutions relate only to either encryption algorithms or to the enhancement of integrity verification. Whereas hybrid cryptographic models extensively improve the confidentiality aspect and blockchain, based systems primarily increase transparency, majority of the existing frameworks lack the combination of encryption, integrity checking, controlled access workflows, and audit logging into a single architecture that is suitable for multi, user environments. In fact, the uncovered gap is what backs up the proposal of the SECURIFY framework, which attempts to integrate cryptographic protection with structured governance and traceability.

IV. Existing System

The system uses AES and RSA encryption algorithms to secure files during their storage and transmission process [2], [9]. The algorithms deliver strong cryptographic protection yet they function as separate security systems that do not link to user authentication systems or ownership control systems [2], [8]. Users who have access rights to encrypted files can access shared files without needing to follow any formal approval process in numerous systems [8]. The method fails to meet requirements for secure access which must be enforced in multi-user environments that include cloud services and academic institutions and organizational networks [5], [8].

The current systems do not provide necessary capabilities for verifying system integrity and tracking system activities [5], [6]. The system lacks any methods to identify unauthorized changes to encrypted documents while it also fails to create complete records of user activities [5]. The system lacks methods to establish responsibility which results in traceability problems that prevent tracking of wrongful actions and security incidents [6], [8]. Traditional encryption systems present operational challenges because they require technical expertise which makes them difficult for regular users to handle [2]. The existing security framework requires enhancement to protect data through encryption while establishing access controls and integrity verification processes and monitoring capabilities which SECURIFY system will address through its proposed solutions [8], [9].

V. Problem Statement

online platforms store and share highly sensitive data, including financial records, personal information, medical records, and company files, through web based services. The data protection system leverages encryption strategies as its main security weapon; nevertheless, the current system only provides very basic security measures and, as a result, it lacks main components for user access, system ownership, and data security verification. The system generates loopholes through which unauthorized users may gain access to and manipulate encrypted data without the system being able to detect them, especially in scenarios where multiple users and cloud storage are involved.

Integrating features such as data encryption, user access control, tamper detection, and the thorough recording of

users' activities in the system could result in a well-functioning and straightforward design for a system. Besides, the system should not put all its trust in very strong encryption algorithms but also efficient key management, and it should even create a way that controls file access through the stages of request and permission while keeping audit records so that the assigning of responsibilities is possible. Digital data sharing platforms need to put these solutions into practice to ensure trustworthy operations. The SECURIFY framework presented is a cryptography-based one that combines hybrid and solutions to tackle the problems of the data lifecycle and constantly ensures confidentiality, integrity, and transparency.

IV. Proposed System

SECURIFY integrates controlled access, comprehensive auditing, and hybrid cryptography into a unified secure file management architecture. The system encrypts sensitive payloads using AES-GCM. RSA locks the key material. We demand more than simple secrecy.

The framework enforces integrity via HMAC verification and strict accountability through a mandatory request-approval workflow. This protocol eradicates careless handling. Administrators or file owners must explicitly authorize every interaction. No user views decrypted data without a verified audit trail and direct approval. Covert access is impossible. The system can be used in a business where trust and traceability and secure storage is crucial since all important activities like uploads, access requests, approvals, decryptions, and downloads are logged in audit logs.

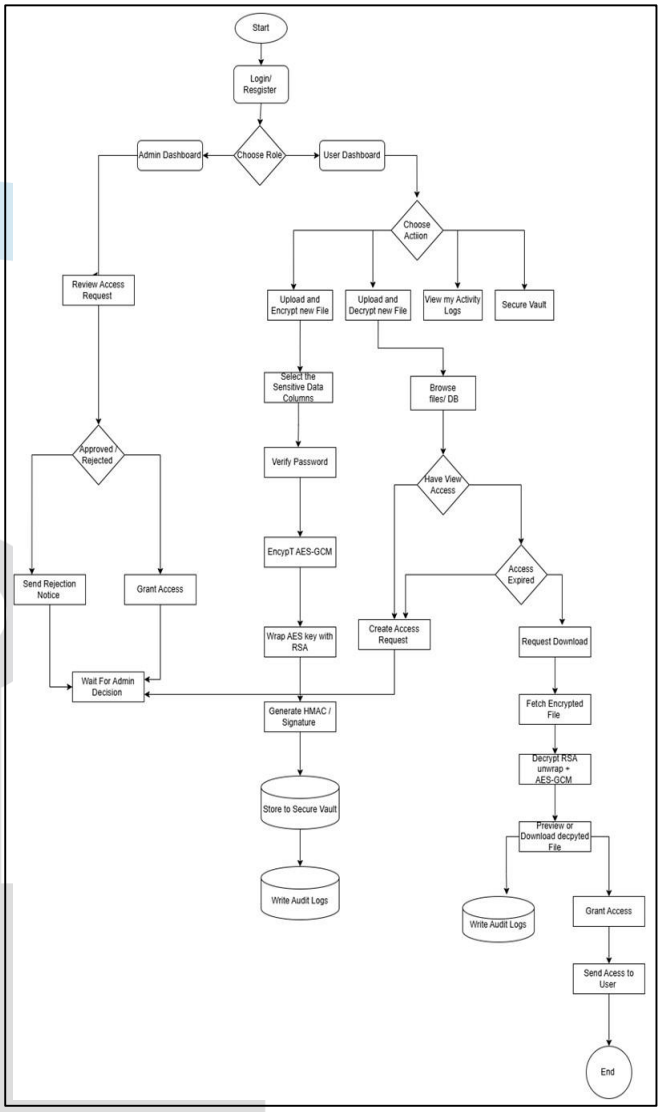

Figure 2. Flow Chart

A. System Flow

The system flow of SECURIFY is meant to ensure that all the interactions with a file undergo a series of safe, controlled and fully traceable tasks during the time period that a user uploads a document until another user is allowed to view it. In order to make sure that the system is only accessed by the authorized users, the process begins when a registered user logs into the application through the authenticated interface. On authentication, the user can post a file and it is posted directly to the backend where SECURIFY begins its security pipeline. The system creates a specific Data Encryption Key for the upload process after the encryption module receives the file. The engine encrypts file payloads with AES-GCM encryption. The process produces ciphertext together with an authentication tag which verifies the integrity of the data. SECURIFY simultaneously wraps the Data Encryption Key using RSA. The system protects the decryption key from unauthorized access because the RSA private key serves as the only method to unlock it even when an attacker gains access to the metadata layer.

The system calculates HMAC as soon as encryption is done to set up methods for identifying tampering. Users are barred from getting hold of the ciphertext, RSA, wrapped AES key, AES, GCM tag, and metadata by the vault. Access can only be given to authorized people. SECURIFY requires a protocol based on workflows. Users make access requests in a formal way through the interface. The system keeps them as pending until the file owner or administrator gives the go ahead. Supervision is a must.

The system starts its fetching process as soon as it gets the go, ahead. It is the system that decrypts the AES Data Encryption Key after it has been encrypted by the RSA public key using the RSA private key. Before the system continues with the decryption, it conducts a check on the HMAC and the authentication tag of the AES, GCM, first. After the successful completion of all the verification tests, the system will then carry out AES, GCM decryption. The system generates a secure, temporary user interface through which it displays the original data obtained after reconstruction. The audit log is recording the full journey of every action starting from upload through encryption to request and approval and finally download. The everlasting documents which keep the forensic tracking and give the verification of the responsibilities in full.

B. Encryption

SECURIFY changes raw file uploads into secure restricted asset. Getting into the system is controlled through authorized paths only and by permission. The first step in the encryption procedure is to generate a unique Data Encryption Key (DEK) that will be used to encrypt each individual file that is submitted. The key encrypts the file through AES-GCM which produces an authentication tag that establishes the integrity link between encrypted data and its transformation from plaintext to ciphertext. The policy specifies different HMAC values which security personnel can use to check file integrity through encrypted documents. The AES key protection process requires its actual encryption key to be safeguarded thus the system encrypts the generated key using the RSA public key. The stage produces its final results through an encrypted file which contains an RSA-wrapped key along with its corresponding tags and integrity values that are stored in a secure vault. The process guarantees that any individual who obtains access to the storage area will be unable to access the file until they pass through the SECURIFY decryption system and obtain permission.

C. Decryption

The system starts decryption only after it verifies that the user requesting access has permission from the file owner or administrator. SECURIFY receives the encrypted file and related metadata from the safe vault upon approval of a request. The system first uses the RSA private key to decrypt the stored AES Data Encryption Key. The AES-GCM decryption procedure is carried out using this recovered key, and the authentication tag created during encryption is checked to make sure the ciphertext hasn't been changed. To give an extra layer of integrity testing, if an extra HMAC was calculated, it is recalculated and in comparison, to the stored value. The system reconstructs the original document and makes it visible and accessible to the authorized user for viewing or downloading only when all integrity checks are successful. No manipulated file is secretly opened, no unauthorized user can circumvent the procedure, and every successful or unsuccessful decryption attempt is documented for future monitoring and compliance thanks to this managed decryption pipeline.

III. SECURIFY Framework

SECURIFY delivers a comprehensive secure file management architecture. It provides the principles like Integrity, Confidentiality and limited access. Encryption AES is used by The System to protect file data. The Decryption Keys are protected by RSA to Prevent un wanted recovery. This Framework requires HMAC verification for every file is generated in order to identify manipulation. We don't just use Cryptography alone.

SECURIFY mandates a workflow-based access control protocol. Users must submit explicit access requests. Only the file owner or administrator authorizes these requests. Audit logs capture every upload, permission change, and download. This transparency eliminates abuse and ensures accountability.

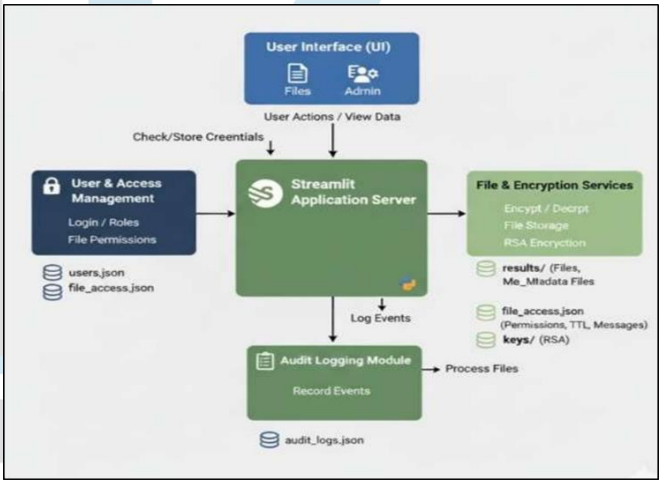

Figure 1. System Architecture

A. Hybrid Cryptography

SECURIFY is a hybrid cryptography-based model. We combine symmetric and asymmetric algorithms to get the best of both cryptographic rigor and high, performance throughput. Symmetric encryption directly handles the data payload. It is the key to operations being done at very high speed. Whereas, asymmetric protocols maintain decryption keys highly secure by their permanent isolation from the system. The system can give the full range of security against threats while, at the same time, it is able to perform at the highest speed level possible. The two, layer security system first encrypts and decrypts confidential data at high speed, then the encryption keys are kept securely. By means of hybrid cryptography which integrates AES and RSA to give separate security to file content and their keys, SECURIFY is going to be able to achieve productivity. The multiple, multi-layer security systems protect data via several encoding

methods which are out of reach of the attackers even after one security layer is compromised.

B. AES-GCM Encryption

SECURIFY uses Advanced Encryption Standard Galois/Counter Mode AES, GCM for encryption to protect uploaded files, which is a highly secure symmetrical encryption method. The organization picked AES, GCM because it provides both security and privacy protection while its encryption system with built, in authentication features also defending against unauthorized content changes. AES, GCM encrypts the uploaded data by using a unique Data Encryption Key that works on each individual file upload to minimize the risks of key reuse and key disclosure incidents. The GCM mode encrypts files while creating an authentication tag that checks the encrypted message's integrity; this function provides extra security which detects system tampering besides the system's HMAC method. The AES-GCM system protects large amounts of sensitive information through its fast-processing speed and low computational requirements and its strong defense against cryptographic attacks. This is why it serves as the perfect match to protect secure file storage system of SECURIFY.

C. RSA Key Protection

The encryption key used for AES must be safeguarded in addition to the file content. SECURIFY stores the Data Encryption Key after encrypting it with the RSA asymmetric cryptographic method. The AES key becomes available through public key and private key encryption which enables the system owner and other authorized administrators to decrypt the file. The division of responsibilities between different users establishes a secure method for protecting sensitive information. The attacker who possesses encrypted file access will remain unable to obtain the AES key because he needs the RSA private key. The system will maintain protection against all types of threats because it secures both external and internal risks including theft of keys through illegal methods.

D. HMAC-Based Integrity Verification

SECURIFY uses HMAC-SHA256 for file integrity verification which it applies to encrypted files to confirm that these files maintain their original state throughout their storage period. The system uses both the encrypted data and a secret integrity key to create a Hash-Based Message Authentication Code after a file has been encrypted. . When a file is retrieved, SECURIFY generates the HMAC again. The system checks this hash against the one that is stored and unchangeable. If a difference, even the smallest one is found, a tampering alert is raised. This method provides a guarantee of the integrity of the data. It removes the means for the data to be altered without the knowledge of the users.

E. Secure Vault Storage

The Secure Vault keeps separate and isolated all the encrypted payloads, keys, and metadata. Users get access to cleartext only after a decryption and an integrity check of a very high standard have been performed.

HMAC values and wrapped keys are present in structured schemas. This arrangement offers an excellent balance between the retrieval speed and a stringent security posture. SECURIFY separates storage from the interface dramatically reduces the leak paths of accidental disclosures. A direct file exposure is still impossible.

F. Access Control

Strict access control is the main feature of SECURIFY architecture. We do not allow direct access to the encrypted assets. The system implements a request, approval workflow that is mandatory. Users are required to submit formal requests for decryption privileges. Administrators check the requests against the users' identity and the operational need. Approval triggers the automated release sequence. The engine unwraps the RSA-protected key and validates the HMAC. The user gains access only after successful verification. This protocol enforces accountability. It neutralizes unauthorized data exfiltration.

VI. Results and Analysis

6.1 Login Page

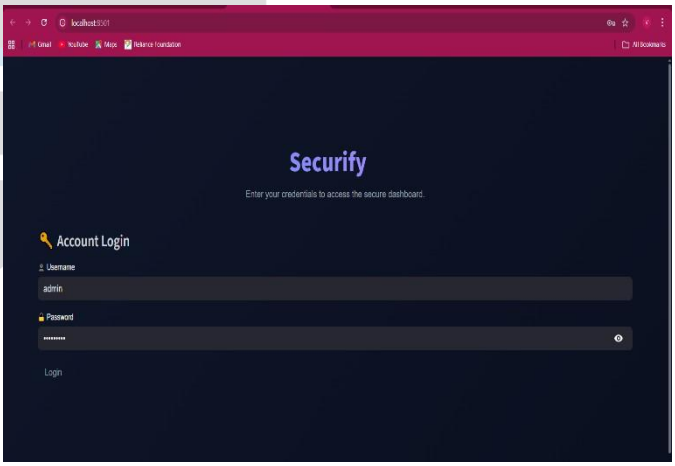

Figure 6.1: Login Page

Shows the login screen where users enter their credentials to securely access the system.

6.2 Admin Dashboard

Figure 6.2: Admin Dashboard

Displays the admin panel used to manage users, handle access requests, and monitor system activities.

6.3 Encryption Page

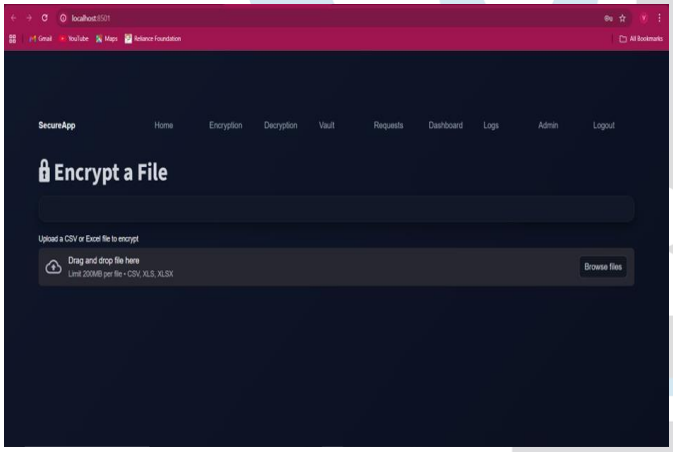

Figure 6.3: Encryption of File

Shows the process of uploading and encrypting a file using AES-GCM to keep the data secure.

6.4 Successful Encryption

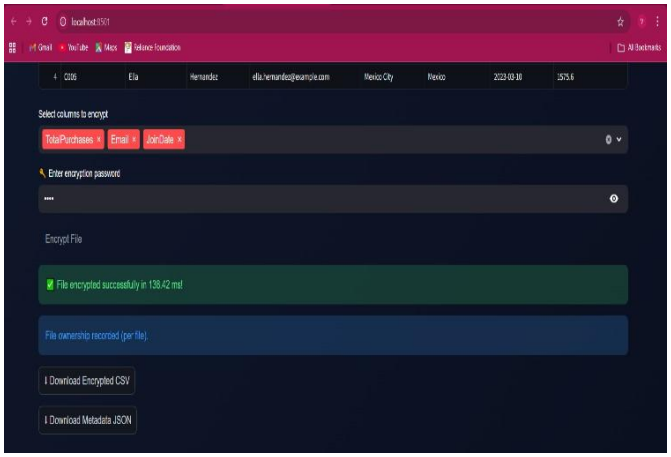

Figure 6.4: Successful Encryption

Indicates that the file has been encrypted successfully and stored safely in the system.

6.5 Decryption Of File

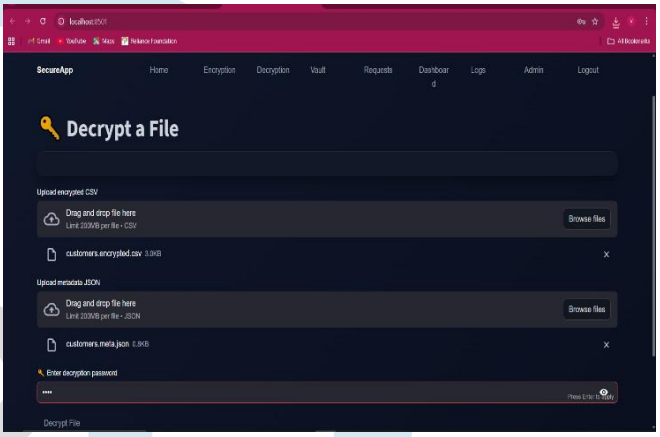

Figure 6.5: Decryption Of File

This figure shows the decryption process after user approval. The system retrieves the encrypted file and begins converting it back to original form.

6.6 Successful Decryption

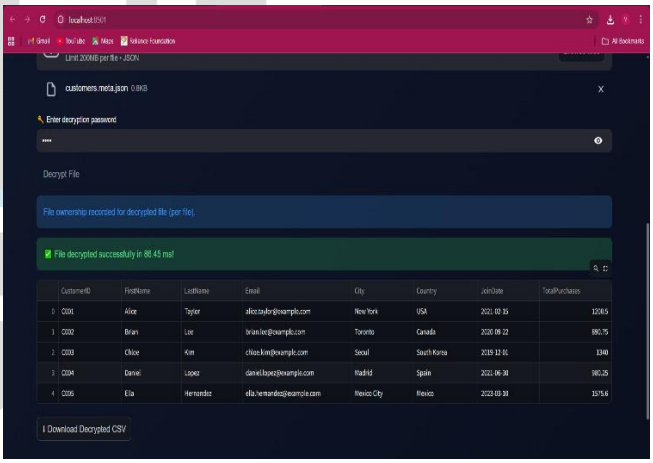

Figure 6.6: Successful Decryption

This figure displays the successful recovery of the original file. It confirms that the decryption process and integrity checks are completed.

6.7 Audit Logs

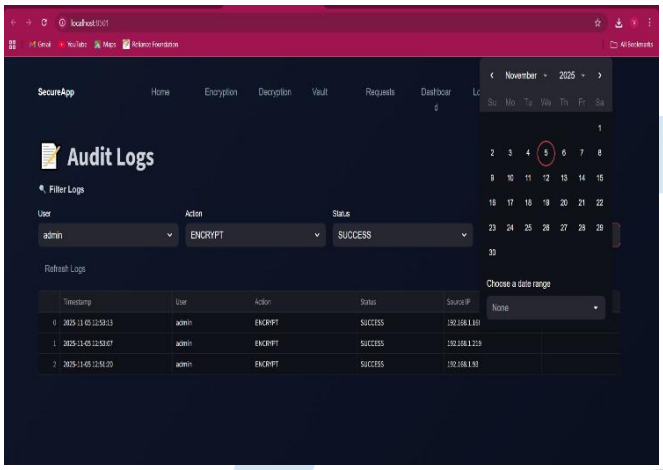

Figure 6.7: Audit Logs

This figure shows the audit log system that records all user activities. It helps in tracking actions like uploads, requests, and approvals.

6.8 Performance Dashboard

Figure 6.8: Performance Dashboard

This figure presents system performance metrics such as encryption and decryption time. It helps in analyzing system efficiency.

6.9 Secure Vault

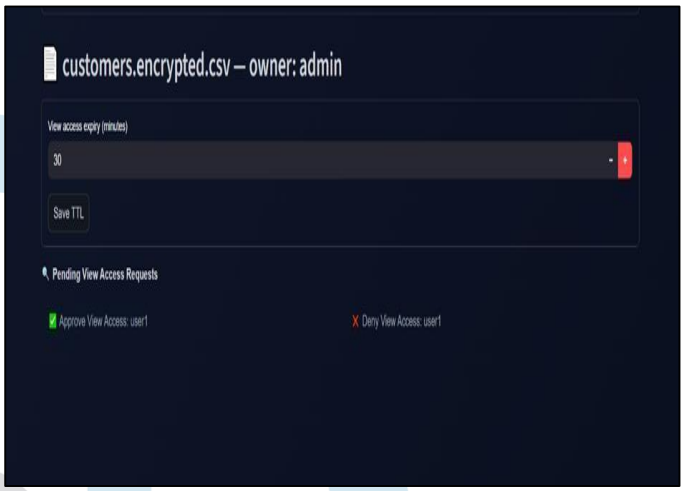

Figure 6.9: Secure Vault

This figure shows the secure storage area of the system. It stores encrypted files, keys, and metadata in a protected environment.

A. Implementation and Experimental Results

AES, GCM, RSA, 2048, and HMAC, SHA256 are the primary security mechanisms of the SECURIFY system, which was developed in Python 3. 10. To facilitate the workflow of uploading, requesting, approving, and receiving encrypted files, a Streamlit, based interface was implemented. In order to ensure that no user will ever be able to see a plaintext file, all encrypted data, keys, metadata, and logs were stored in a highly secure vault structure. Different types of files, including simple text files and large media files, were mainly used for experimental testing purposes. AES, GCM proved to be fast in encryption with average timing of encryption is 0. 12 seconds and average timing of decryption is 0. 07 seconds. RSA key wrapping is barely noticeable in terms of latency. Integrity tests were very thorough. In addition to the AES, GCM authentication tag, the HMAC verification successfully detected all alterations. Even the slightest changes at the byte level caused an instant alarm.

We put the system under heavy concurrent operations. Several users were uploading, requesting, and downloading content at the same time. The time to response of the system did not change. Based approval mechanism workflow has demonstrated its value. It not only managed authorization very tightly but also prevented internal misuses. Besides, we have also examined the audit trail. The event logging system recorded all the activities very tightly one after another to support forensic compliance. The result of the numerous tests is that SECURIFY provides high, performance encryption and strong tamper detection that are fit for production environments.

I. Comparative Analysis Table

Security Feature	Existing System	SECURIFY
Hybrid Encryption (AES + RSA)	Yes	Yes
Integrity Verification (HMAC / Hashing)	Partial	Yes
Workflow-Based Access Control	No	Yes
Audit Logging	No	Yes
Multi-user Support	Limited	Yes
Tamper Detection	Basic	Strong (HMAC + GCM Tag)
Governance & Approval Mechanism	No	Yes
Centralized Secure Vault	No	Yes
Traceability of Actions	No	Yes
Performance Overhead	Low	Slightly Higher (Acceptable)

B. The Proposed System's Comparative Analysis

Techniques	Message Size	Time taken	Encryption	Decryption
AES-GCM Encryption	1 MB	0.14 s	0.14 s	0.07 s
AES-GCM + RSA Key Wrapping	1 MB	0.14 s (AES-GCM) + 0.02 s (RSA)	0.16 s	0.09 s
AES-GCM with HMAC Integrity Verification	1 MB	0.14 s + 0.01 s	0.15 s	0.08 s
Complete SECURIFY Pipeline (AES-GCM + RSA + HMAC)	1 MB	AES-GCM: 0.14 s + RSA: 0.02 s + HMAC: 0.01 s	0.17 s	0.09 s

VII. Conclusion

SECURIFY is offering a package of security features that will help to protect sensitive digital assets from being harmed. It integrates workflow oriented granting of right of use with hybrid cryptography and tamper evidence logging. The system uses AES, GCM for authenticated encryption and RSA for secure key encapsulation. HMAC is used to apply highly strict tamper detection. This set of three measures efficiently closes the data to being leaked, tampered with or accessed by unauthorized individuals. We don't allow decryption without control in any way. It is through the formalized request and approval workflow that it is guaranteed that every access is a purposeful one. There has to be a responsible decision maker who can authorize each and every file release. SECURIFY is a high speed security technology that can deliver securely multi layered security enforcement while doing so without any major loss in speed conducted. We provide the traceability feature when confidentiality alone is not sufficient. The detailed audit trail makes compliance easy and forensic investigation can be easily conducted. SECURIFY sets up a very safe, well, organized environment that is capable of completely counteracting present threats. This design offers a scalable base for the next security upgrades.

VIII. Future Work

One way to further develop the SECURIFY framework is through enhancing its scalability capacity while simultaneously implementing advanced security measures that very much support large scale deployment. A blockchain based audit log can be embedded to offer totally tamper, proof and immutable records for higher forensic trustworthiness. the platform can be enabled with cloud integration and distributed storage techniques to facilitate enterprise-level data management. Moreover, an AI, driven anomaly detection feature can be added to track suspicious behaviour and insider threats as if happen. Row level access control in files can be implemented to grant or deny access to certain parts of highly confidential documents based on user permission. Additionally, performance enhancement for large files and parallel encryption processing can make the system's operation faster in a high workload setting.

IX. Acknowledgement

We would like to sincerely thank our esteemed mentor and the Faculty from Pillai HOC College of Engineering and Technology , Department of Computer Applications for their unwavering support, encouragement, and advice during the course of this project. Their insightful comments and helpful criticism significantly improved SECURIFY's technical and research components. We also appreciate our university for providing the environment, resources, and infrastructure needed to make this work possible. Lastly, we would like to express our gratitude to our colleagues and project partners for their collaboration, timely conversations, and contributions that enabled us to successfully finish this research and improve the system.

IX. References

[1] S. Singh and R. Kumar, "Enhancing Secure Communication in Industry 4.0 using Digital Signatures and Two-Level Encryption," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 2, pp. 112–118, 2023.

[2] P. Mehta and V. Sharma, "RSA and AES Based Hybrid Encryption Technique for Cloud Computing," International Journal of Computer Applications, vol. 185, no. 12, pp. 21–27, 2023.

[3] L. Chen and J. Wang, "ES-SECS/GEM: An Efficient Security Mechanism for SECS/GEM Communication," IEEE Access, vol. 11, pp. 51210–51219, 2023.

[4] A. Patel and D. Deshmukh, "Hybrid Cryptography Scheme for Secured Data Transmission," International Research Journal of Engineering and Technology (IRJET), vol. 10, no. 5, pp. 1528– 1534, 2023.

[5] R. Yadav and S. Nair, "A secure file storage system utilizing SHA-256 along with a hybrid AES–RSA approach," Int. J. Innovative Res. Computer Communication Eng., vol. 11, no. 2, pp. 89–95, 2023.

[6] J. Park and M. Lee, "Blockchain-Enabled Data Security Framework for Cloud Environment," IEEE Transactions on Cloud Computing, vol. 12, no. 4, pp. 248–259, 2024.

[7] N. Gupta and A. Jain, "ECC and AES Based Hybrid Encryption for IoT Healthcare," Journal of Network and Computer Applications, vol. 224, pp. 103–118, 2024.

[8] S. Dutta and R. Agarwal, "Attribute-Based Encryption for Secure Cloud Data Sharing," International Journal of Information Security Science, vol. 13, no. 1, pp. 56–63, 2023.

[9] T. Roy and M. Chatterjee, "Enhanced Hybrid Security Model Using AES, RSA, and SHA-512," International Journal of Emerging Technologies in Engineering Research (IJETER), vol. 10, no. 7,pp. 141–147, 2022.

[10] H. Kaur and V. Bansal, "Two-Phase Hybrid Encryption and Integrity Verification Framework," International Journal of Engineering Trends and Technology (IJETT), vol. 71, no. 9, pp. 45–50, 2023.

[11] S. Rath, P. Kumar, and A. Singh, "AES-RSA: An Innovative Hybrid Security Framework for File Authentication, Integrity, and Data Secrecy Model," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 3, pp. 245–254, 2024, Publisher: IJISAE.

[12] A. Gautam and R. Vishwakarma, "Enhancing Communication Security in Hybrid Cloud Environments Through an Innovative Cryptographic Algorithm," International Journal of Intelligent Systems and Applications in Engineering (IJISAE), vol. 12, no. 2, pp. 112–121, 2024, Publisher: IJISAE.

[13] Y. Zhang, L. Chen, and H. Zhao, "Enhancing Blockchain-Based Audit Data Privacy via Hybrid Chaotic and RSA Encryption: Mechanism Design and Performance Evaluation," Journal of Cloud Computing, vol. 14, no. 1, pp. 1–15, 2025, Publisher: Springer.

[14] M. Farshadinia, S. Rezaei, and H. Khosravi, "Designing a Layered Framework to Secure Data via Improved Multi-Stage Lightweight Cryptography in IoT Cloud Systems," IEEE Access, vol. 13, pp. 34521–34535, 2025, Publisher: IEEE.

[15] J. Mozo, D. Fernández, and P. García, "Quantum-Classical Hybrid Encryption Framework Based on Simulated BB84 and AES-256: Design and Experimental Evaluation," Future Generation Computer Systems, vol. 156, pp. 210–223, 2025, Publisher: Elsevier.