Grenze International Journal of Engineering and Technology, June Issue

A Multi-Layer Secure File Protection Framework with Blockchain-Backed Integrity Verification and IPFS-based Storage

Adithya K P[1], Adithya U[2], Diya Mariya Augustine[3], Fathima Ali Pulikkool Veetil[4] and Adithya C H[5]
[1-5]Department of Computer Science and Engineering, Vimal Jyothi Engineering College, Chemperi, Kerala, India
Email: adithyaakp2004@gmail.com, adithyaupendran@yahoo.com, diyaelavumkal4@gmail.com, fathimaali2424@gmail.com, adithyach@vjec.ac.in

***Abstract*— Personal computing systems frequently store sensitive digital information, making them targets for unauthorized access and insider misuse. Conventional file protection mechanisms rely primarily on single-layer security approaches such as password authentication or encryption, which become insufficient once partial system access is obtained. This paper presents Secure Vault, a multi-layer secure file protection framework integrating AES-256 encryption, machine learning–based behavioral authentication, steganographic concealment, file camouflage, decentralized storage using IPFS, a local Blockchain integrity ledger, and deception-based honeytrap detection. Files are encrypted locally before concealment or decentralized storage. A local blockchain maintains immutable hash-linked records to ensure tamper detection. Experimental evaluation demonstrates improved resistance against credential compromise, file discovery, and integrity tampering while maintaining manageable computational overhead in desktop environments.**

***Index Terms*— Secure File Vault, AES Encryption, Keystroke Dynamics, Steganography, Blockchain Integrity, IPFS, Honeytrap Security.**

I. INTRODUCTION

The swift expansion of digital data storage has resulted in a greater dependence on local encrypted vault systems to safeguard confidential organizational and personal data. Strong mathematical guarantees against direct cryptanalysis are offered by contemporary cryptographic algorithms like AES; yet, the encryption primitive itself is rarely the focus of real-world attacks. Rather, adversaries take advantage of behavioral predictability, password repetition, forensic discovery of encrypted containers, and inadequate procedures for authentication. For this reason, in real-world deployment scenarios, the confidentiality that encryption offers by itself does not always equate to complete security.

Single-factor authentication, most frequently password-based access control, is the foundation of traditional safe vault designs. Although this method provides some security, it is still susceptible to replay-based assaults, shoulder surfing, credential leaks, and brute-force efforts. Additionally, encrypted vault files are frequently simple to locate inside file systems, which makes them desirable targets for forensic examination. Tamper-evident logging techniques, deception-based misdirection, and behavioral authentication are seldom combined into a single design in existing systems.

Grenze ID: 01.GIJET.12.2.141_24

Due to this fragmentation, there are significant weaknesses in resistance to post-compromise analysis and unwanted access. This paper suggests a multi-layered secure storage architecture that combines behavioral biometrics, steganographic hiding, deception-driven redirection, tamper-evident integrity tracking, and cryptographic confidentiality in order to overcome these drawbacks. The suggested method uses a defense-in-depth approach rather than encryption as the only line of security. Unauthorized access attempts are diverted to realistic decoy environments, encrypted files are optionally hidden within cover media, keystroke-based behavioral profiling is used to enhance authentication, and a hash-linked ledger with optional decentralized anchoring maintains file integrity. This study primarily makes four contributions. It begins by presenting a dual-vault deception model that logs behavioral signs and reroutes questionable access attempts into controlled decoy settings. Second, it uses timing-based typing dynamics to construct an ensemble-based behavioral biometric authentication system that enhances password verification. Third, it reduces the forensic visibility of encrypted assets by integrating camouflage and steganographic hiding techniques. Fourth, it uses a tamper-evident hash-linked ledger with optional IPFS-based anchoring to provide distributed backup and integrity verification. These elements work together to create a coherent architecture that aims to strengthen defenses against attacks from several vectors. This paper evaluates the proposed architecture through experimental analysis of authentication performance, tamper detection capability, deception effectiveness, and computational overhead. Authentication accuracy is assessed using behavioral similarity metrics, while system integrity and concealment mechanisms are tested under simulated adversarial conditions. The results demonstrate that layered integration of behavioral authentication and deception mechanisms can significantly enhance resistance to unauthorized access attempts compared to traditional password-only encrypted vault systems, while maintaining practical usability and performance.

II. Related Work

In order to combat both conventional and quantum threats, recent research has investigated hybrid security models that include steganography, encryption, and quantum key distribution. For practical encryption, entanglement-based protocols like E91 produce safe keys that are used with hash functions and AES. Entropy, NPCR, and UACI are used to test the resilience of steganographic techniques, such as LSB and deep learning approaches, which are employed to disguise encrypted data. However, rather than concentrating on more comprehensive layered protection structures, the majority of current techniques prioritize secure transmission.

By combining honeypot techniques with honey encryption to reduce brute-force attacks, research has enhanced password manager security. While honeypot systems track illegal interactions and issue intrusion alarms, honey encryption reduces the efficacy of offline guessing efforts by producing believable decoy outputs when wrong passwords are submitted. Despite being successful in enhancing credential protection, these methods mostly concentrate on password security rather than all-encompassing multi-layered storage systems.

Keystroke dynamics-based user authentication employing deep learning models to differentiate real users from fraudsters has been the subject of recent behavioral biometrics research. While interpolation and feature normalization techniques are used to accommodate changes in password length and input structure, methods like Siamese neural networks have been used to identify similarity patterns across typing samples. By balancing the rates of false acceptance and rejection, statistical decision frameworks and adaptive thresholding further improve resilience. These techniques are usually assessed as stand-alone systems rather than included into larger multi-layered secure storage architectures, despite the fact that they exhibit good authentication performance in controlled situations. The effects of layered obfuscation strategies on the structural visibility and detectability of protected artifacts have been investigated, especially in relation to static analysis and pattern-based categorization. According to research, changes such layered code updates, control flow alteration, and structural reorganization can have a substantial impact on complexity measurements and lower the dependability of traditional detection methods. Crucially, detectability is more affected by the existence and variety of obfuscation layers than by their application sequence, suggesting that multi-layer concealing techniques can successfully boost resistance against structural analysis. These results demonstrate how well layered camouflage and concealment techniques work in secure storage infrastructures.

In order to combat data exfiltration ransomware, recent studies have suggested clever decoy-based protection methods that use incredibly lifelike fake documents in workplace settings. These systems create fake files that closely resemble sensitive data using Generative Adversarial Networks, which raises the possibility of adversary involvement during the reconnaissance and staging stages. Active containment techniques including network isolation, traffic blocking, and process throttling are used in conjunction with real-time monitoring of decoy interactions to stop outgoing data transmission as soon as malicious activity is identified. High engagement rates,

quick reaction times, and minimal operational overhead were shown in an experimental test against many modern ransomware variants, underscoring the value of clever decoy tactics in early-stage intrusion prevention.
By altering the least significant bits of pixel intensity levels inside RGB pictures, LSB replacement techniques hide hidden data and enable information embedding with the least amount of perceptual distortion. To improve secrecy and lessen predictable embedding artifacts, advanced solutions include multi-level encryption, structured bit replacement patterns, and secret key-driven embedding. Controlled bit replacement techniques and channel-wise manipulation further enhance camouflage by more evenly dispersing concealed data throughout the picture structure. These methods show how organized LSB embedding may improve data concealment while preserving visual coherence and thwarting easy detection.
Studies on malware analysis have demonstrated that superficial file-level hashing methods are extremely susceptibletoevasionbysmallstructuralchangesthatmodifycryptographichasheswithoutaffectingtheirfundamental operation. To get around signature-based detection while maintaining behavioral consistency, attackers might alter properties like section names, sizes, and metadata. These results underline the drawbacks of depending just on hash verification at the surface level and stress the significance of more profound structural integrity methods in security designs. Using machine learning models, keystroke dynamics authentication uses temporal typing characteristics, such as inter-key delay and key press time, to differentiate authentic users from fraudsters. Neural network designs that learn user-specific typing rhythms for static or continuous authentication are among the methods, as are classifiers like Support Vector Machines and Random Forests. These techniques show that software-based behavioral biometrics may be used as an extra layer of authentication without the need for specific hardware. Adaptive encryption frameworks use machine learning classifiers to classify data before applying the proper cryptographic algorithms, therefore dynamically adjusting the encryption strength according to the sensitivity of the data. By using methods like K-Nearest Neighbors to separate sensitive from non-sensitive data, these designs allow for hybrid encryption for sensitive data and lighter encryption for less sensitive stuff. By customizing encryption techniques rather than implementing consistent protection for all data, this context-aware method strikes a balance between security and computing performance.
Frameworks for coverless steganography conceal data without changing the carrier picture by combining image-based mapping with cryptographic security. In hybrid models, secret data is encrypted using AES, and safe key exchange is made possible via Elliptic Curve Cryptography, which guarantees secrecy and secure transmission. Metadata transmission is not required since encrypted message segments are routed to preconstructed picture datasets using deterministic methods rather than directly embedding data into pixel values. Secure and secret communication is supported by feature extraction techniques like SURF, which allow for reliable picture retrieval even in the presence of distortions without compromising visual content.
In order to assess user consistency and attacker distinction, structured datasets that record press-to-press, release-to-press, and hold-time metrics have been used to study fixed-text keystroke dynamics authentication. These datasets allow for the examination of false acceptance and false rejection trade-offs under threshold-based and distance-based classifiers by combining both authentic user samples and actual attack attempts. Results show that password attributes and typing speed affect classification reliability, emphasizing the value of simulating attacker behavior in behavioral authentication systems in addition to real user behaviors.

III. PROPOSED METHODOLOGY

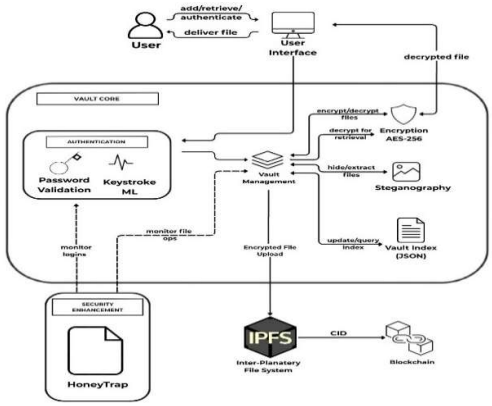

Figure 1: System Architecture of Secure Vault

The Secure Vault framework is implemented as a layered secure file protection system combining authenticated encryption, behavioral authentication, steganographic concealment, file camouflage, decentralized storage, and a local blockchain- based integrity ledger.

In this Secure Vault system, protection responsibilities are distributed across independent modules to reduce single-point compromise risk.

The overall architecture is illustrated in Fig. 1. The system follows a sequential workflow beginning with authentication and proceeding through encryption, storage, concealment, and integrity recording.

A. Behavioral Authentication Module

Authentication combines password verification with statistical and machine learning–based keystroke dynamics analysis.

During user profile creation, multiple typing samples are collected. For each user, statistical templates are computed:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} F_i \quad (1)$$

$$\sigma = \sqrt{\left((1/n)\Sigma(Fi - \mu)^2\right)} \quad (2)$$

where Fi represents extracted feature vectors.

Feature Extraction

A fixed-length 40-dimensional feature vector is generated including:

- Total typing time
- Average dwell time
- Average flight time
- Typing speed
- Rhythm consistency
- Dwell and flight time variances
- Pressure variance
- Error and backspace counts
- Individual dwell timings (first 15)
- Individual flight timings (first 14)
- Digraph timing features

Statistical Similarity Scoring

During authentication, normalized deviation is computed:

$$D = \frac{|F_c - \mu|}{\sigma + \epsilon} \quad (1)$$

where $\epsilon = 0.001$ avoids division instability.

Confidence is computed as:

$$\text{Confidence} = \frac{1}{1+\overline{D}} \quad (2)$$

Authentication succeeds when:

$$\text{Score} \geq 0.5$$

Ensemble Machine Learning

When sufficient samples are available, the system trains:

- Isolation Forest (anomaly detection)
- One-Class SVM (boundary modeling)
- K-Means clustering (pattern grouping) Final authentication score integrates:
- Statistical similarity
- ML ensemble prediction
- Pattern similarity
- Behavioral heuristics

Fraud indicators are raised if multiple anomaly signals are detected.

Enhanced Feature Vector

The system extracts a fixed-length feature vector of 40 dimensions including:

- Total typing time
- Average dwell time
- Average flight time
- Typing speed
- Rhythm consistency
- Dwell time standard deviation
- Flight time standard deviation
- Pressure variance
- Error rate
- Backspace count
- First 15 dwell timings
- First 14 flight timings
- First 5 digraph timings

Ensemble machine learning models (Isolation Forest, One-Class SVM, and K-Means) are trained when sufficient samples are available to enhance anomaly detection.

B. Encryption and Storage Pipeline

Encryption is performed locally using Fernet authenticated symmetric encryption.

$$K_f = \text{PBKDF2\_HMAC\_SHA256}(PW, S_f, 100000{,}32) \quad (3)$$

$$C = \text{Fernet}(K_f, P) \quad (4)$$

After encryption, the following operations occur sequentially:

- Compute file hash
- Upload encrypted file to IPFS
- Receive CID
- Record hash and CID in blockchain ledger
- Optionally embed encrypted data into image
- Apply system-level camouflage renaming This workflow is illustrated in Fig.2.

C. Steganography Module

Encrypted data may be concealed inside the selected RGB images using Least Significant Bit substitution. For an image of width W and height H:

$$\text{Capacity} = 3WH \quad (8)$$

Required bits:

$$\text{Required} = 8(L + 4) \quad (9)$$

Embedding proceeds only if:

$$\text{Required} \leq \text{Capacity} \quad (10)$$

where L is payload size and 4 bytes represent length metadata.

D. File Camouflage Module

Encrypted or stego files are renamed using system-like file names and are stored in randomly selected directories that resemble operating system paths. A JSON mapping file records the associations between original and obfuscated filenames.

This camouflage layer reduces any kind of casual discovery risk faced during filesystem inspection.

E. IPFS Storage Module

Encrypted files may be uploaded to a local IPFS node. The system receives a content identifier (CID):

$$CID = IPFS(C) \quad (11)$$

IPFS uses content-based addressing. This makes any file modification result in a different CID

F. Local Blockchain Integrity Ledger

A lightweight single-node blockchain ledger used for recording the file integrity metadata. The encrypted file hash is computed:

$$H = SHA256(C) \quad (12)$$

Each block contains:

$$Block_i = index_i, timestamp_i, file_id_i, file_hash_i, ipfs_cid_i, prev_hash_i, hash_i \quad (13)$$

$$\text{hash}_\text{i} = \text{SHA256}(\text{Block}_\text{i}) \quad (14)$$

This ledger is implemented locally without any distributed consensus and is used only for tamper detection.

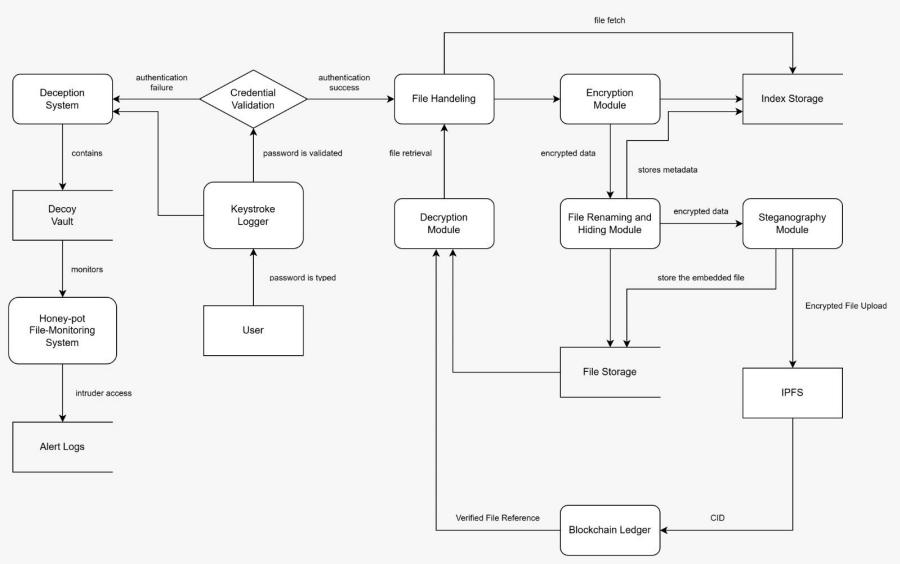

Figure 2: Encryption, IPFS Upload, Blockchain Recording, and Secure Vault Workflow

G. Decoy and Honeytrap Module

A decoy vault structure containing realistic but non-sensitive files is generated alongside the primary vault. Access to decoy files is logged as intruder detection. Behavioral fraud indicators further strengthens any suspicious activity detection.

IV. Experimental Setup

Experiments were conducted on a Windows-based system with standard desktop hardware. File sizes ranged from 1MB to 50MB.

Evaluation focused on:

- Encryption latency
- Authentication accuracy
- Steganographic embedding reliability
- Blockchain integrity verification
- IPFS upload and retrieval consistency

V. Results and Discussion

A. Encryption Performance

Larger files required more time to encrypt. The PBKDF2 key derivation created overhead due to its predefined iteration count.

B. Authentication Accuracy

TABLE I: AUTHENTICATION PERFORMANCE METRICS

Metric	Value
True Acceptance Rate (TAR)	90%
False Rejection Rate (FRR)	10%
False Acceptance Rate (FAR)	4.55%
True Rejection Rate	95.45%
Overall Accuracy	93.75%
Total Attempts	112

The behavioral authentication detected and rejected login attempts when valid credentials were reused by intruders. The ensemble scoring enhanced decision stability and reduced false positives compared to relying on a single matching approach.

C. Steganographic Evaluation

The embedded images does not lose their visual integrity. The hidden data was successfully recovered during controlled experiments.

D. Integrity Verification

Tampered files were successfully identified through blockchain hash mismatch checks. Also IPFS CID verification offered additional assurance of data integrity.

VI. CONCLUSION

This work presented Secure Vault, a multi-layer secure file protection framework integrating authenticated encryption, behavioral authentication, steganographic concealment, decentralized storage, blockchain-based integrity verification, and deception-driven intrusion detection. The multi-layered design distributes the security responsibilities with independent modules, which reduced the reliance on any single mechanism.
Experimental evaluation indicates that the framework improves resistance against credential mishap, file discovery, and integrity tampering. Also maintains the practical usability for desktop environments.

FUTURE WORK

Future works include cloud implementation, and improved steganographic resistance against statistical analysis. Also, real-time alerting mechanisms can be integrated into the honeytrap detection module to notify users of suspicious access attempts through secure messaging or system notifications.

REFERENCES

[1] A. Sykot, M. S. Azad, W. R. Tanha, B. M. M. Morshed, S. E. U. Shubha, and M. R. C. Mahdy, “Multilayered security system: Integrating quantum key distribution with classical cryptography to enhance steganographic security,” Alexandria Engineering Journal, vol. 121, 2025.

[2] M. Barochiya, P. Makhijani, H. N. Patel, P. Goel, and B. Patel, “Password manager security using honey encryption algorithm and honeypot technique,” Procedia Computer Science, 2024.

[3] V. Medvedev, A. Budz˘ys, and O. Kurasova, “A decision-making framework for user authentication using keystroke dynamics,” Computers & Security, vol. 155, Aug. 2025.

[4] S. Raubitzek, S. Schrittwieser, E. Wimmer, and K. Mallinger, “Obfuscation undercover: Unraveling the impact of obfuscation layering on structural code patterns,” Journal of Information Security and Applications, vol. 85, 2024.

[5] S. Liu and X. Chen, “Mitigating data exfiltration ransomware through advanced decoy file strategies,” Research Square, 2023.

[6] S. Rahman, J. Uddin, H. Hussain et al., “A novel and efficient digital image steganography technique using least significant bit substitution,” Scientific Reports, vol. 15, p. 107, 2025.

[7] A. Abuadbba et al., “Enhancing malware fingerprinting through analysis of evasive techniques,” arXiv preprint arXiv:2503.06495, 2025.

[8] A. Arsh, N. Kar, S. Das, and S. Deb, “Multiple approaches towards authentication using keystroke dynamics,” Procedia Computer Science, vol. 235, 2024.

[9] P. R. Kumar and S. Goel, “A secure and efficient encryption system based on adaptive and machine learning for securing data in fog computing,” Scientific Reports, vol. 15, p. 11654, 2025.

[10] S. A. El-Rahman, A. E. Mansour, L. Jamel, M. A. Alohali, M. Seifeldin, and Y. Alkady, “C-hide: A steganographic framework for robust data hiding and advanced security using coverless hybrid image encryption with AES and ECC,” IEEE Access, vol. 13, 2025.
[11] H. N. Risto and O. H. Graven, “Fixed-text keystroke dynamics authentication data set — collection and analysis,” Annales des Tel´ ecommunications´ , vol. 79, pp. 731–743, 2024.