RESEARCH PAPER

A Hybrid Approach for Secure Banking Transactions using Blockchain

Dr. N Palanivel[1], V Kumaraguru[2], K Oviya[3], J Sheerin Farhana[4], B Banumathi[5], G Soundarya[6], M S Ajay [7]

[1] Professor/Head [2]Assistant Professor [3,4,5,6,7] B.Tech Students — Dept. of Computer Science & Engineering, Manakula Vinayagar Institute of Technology, Puducherry, India

hodcseicb@mvit.edu.in | kumaragurucse@gmail.com | oviyak8030@gmail.com | sheerinfj51011@gmail.com | balajibanumathi2004@gmail.com | sithandhra@gmail.com | ajaymurugan08052004@gmail.com

Abstract

Due to the continued increase of internet banking, the security of storage, transfer, and verification of financial transactions has emerged as a major technological concern. The use of central databases is extensively used when the information of transactions is registered in the traditional banking systems. Compared to the standard encryption algorithm, like the Advanced Encryption Standard (AES), which was capable of providing the confidentiality of the information that was being in fact transmitted, it did not conceal the signs of the presence of sensitive information and removed the circumstances that could be introduced by the centralized storage. The negative is that the centralized systems are vulnerable to cyber attacks, insider manipulation, unauthorized changes, and verifiable audit trail. These issues imply that there should be a strong, decentralized and resilient infrastructure that will ensure the confidentiality integrity and privacy of digital banking infrastructure are secured. The given security architecture of this paper is based on the multi-layered architecture which is premised on AES encryption, the InterPlanetary File System, the LSB-based steganography, and Ethereum blockchain to achieve the stable and immutable system of transactions. In the proposed model, the account number and IFSC code, amount and the time of the transaction are first encrypted using the AES so as to transform it into a secret. Its stego-hided ciphertext is then encoded in a digital picture in such a way that it remains totally undetected as this is concealed in the pixel values. The created stego-image is uploaded to the decentralized IPFS storage which is offered a distinct Content Identifier (CID) that will ensure content-based storage and avoid corruption of data. The prototype that has been adopted clearly indicates a secure and unobstructed encryption, data hiding, decentralization of the storage, and blockchain referring that could be audited by any third party in an unreliable and transparent way. Since the experiment has revealed, the system is extremely confidential, one cannot afford to offer unauthorized access and impeccable storage and sensitive data of transactions is not processed. The suggested framework suggests the implementation of cryptographic rigour, invisibility of the data, decentralized storage and immutability of the blockchain as a solution to the future and scalable approach to enhanced security of the digital banking. It provides the base of new financial models which are privacy, transparency and trust based financial models.

Keywords—AES Encryption, Steganography, Blockchain, InterPlanetary File System (IPFS), Secure Banking Transactions, Data Confidentiality, and Integrity

How to cite this article: Palanivel N, Kumaraguru V, Oviya K, Farhana JS, Banumathi B, Soundarya G, Ajay MS., A Hybrid Approach for Secure Banking Transactions using Blockchain. Int J Drug Deliv Technol. 2026;16(45s): 45-60; DOI: 10.25258/ijddt.16.45s.4.

I. INTRODUCTION

The high level of development of digital banking technologies has radically changed the sphere of the financial industry and made it possible to transfer funds, direct payments, mobile banking, and computerized financial services. With the increasing number of the users of digital platforms, online transactions have been exponentially increasing and, in the process, bring sensitive financial information closer to cyber threat. As per the global cybersecurity reports, banks have been among the most attacked sectors given that the information they hold is very valuable. Current attacks are associated with mass data breaches, ransomware, former credential theft, and inside interference with transaction logs. This dynamic threat environment requires the creation of security solutions that go beyond the conventional encryption or centralized storage solutions.

Even though cybersecurity is continuously being enhanced, old fashioned banking platforms are still characterized by inherent vulnerability. In the vast majority of cases, financial institutions are still relying on centralized databases where the records of transactions are held at one point. These architectures have a number of shortcomings: they form a single point of failure, can be susceptible to unauthorized access, can allow malicious insiders to alter the transaction logs, and do not provide much transparency in forensic audits. On encrypted data, attackers can still intercept and analyze the content and even make an attempt of decrypting the ciphertext since encryption does not conceal the presence of sensitive data. Also, centralized storage does not offer tamper-proof properties, i.e. histories of transactions can be modified without leaving behind verifiable audit trails.

In order to mitigate these weaknesses, researchers and industries are transitioning to the multi-layered security designs that adhere to cryptography, data hiding,

decentralized storage, and blockchain-driven immutability. Inspired by the latter, the current work suggests a complex security framework that combines the use of the AES encryption, the LSB steganography, the InterPlanetary File System (IPFS), and the blockchain technology to provide the confidentiality, invisibility, decentralization, and immutability of the records of banking transactions. The layers have a different security property that, when combined, can constitute a powerful solution that can withstand current cyberattacks.

A. AES Encryption:

In the proposed framework the first layer is the Advanced Encryption Standard (AES), which is a well-known and accepted symmetrical-key encryption algorithm by the government. AES uses both 128 bits and block sizes of 128 bits and key lengths of 128, 192, or 256 bits. It is strong because it has a resistance to attacks by brute force, is efficient in computational performance and has a well established mathematical structure. In this system, the details of the transaction, including account number, IFSC code, amount, and timestamp, are encrypted with AES and then they are not subjected to further processing.

Even in case the raw transaction data is snatched or intercepted by unauthorized people, this cannot be read without the proper secret key because it is AES. Nevertheless, AES is a powerful algorithm but the ciphertext is recognizable as encrypted information. It can be detected and targeted by attackers to be decrypted or to be cryptanalyzed. This restriction gives rise to the necessity of an extra layer of the ciphertext itself.

B. Steganography for Data Invisibility:

To curb the issue of detectability with encryption, the second layer uses Least Significant Bit (LSB) steganography which is a process of concealing secret messages in digital media. In this scheme, the AES encrypted message is hidden in a cover image through alteration of the least significant bits of pixels. These changes are small and they can not be seen visually, so the resulting stego-image is exactly the same as the original image.

The significance of Steganography is that although the attacker may be able to access the image stored, he or she will not know that there is any hidden information. This layer can greatly decrease the chances of targeted attacks, brute-force and unauthorized analysis. Through the utilization of encryption, together with steganography, the system will achieve the necessary level of confidentiality and invisibility two crucial elements in secure financial data management.

C. IPFS for Decentralized Storage:

Despite the fact that steganography obfuscates sensitive content, keeping the stego-image in a centralized server would still present the system with the conventional risks. In order to resolve this, the architecture suggested employs the Interplanetary File System (IPFS) which is a decentralized peer-to-peer network that is aimed at substituting place-based addressing with content-based addressing. Upon uploading the stego-image into IPFS, it is broken into smaller bits and distributed over several nodes and a unique Content Identifier (CID) is generated using cryptographic hashing.

The CID is an evil-doer reference. Any alteration of the stego-image by any bit will be detected as a CID alteration indicating any attempt of tampering. Through decentralized storage, the system eradicates points of failure, decreases reliance on centralized banks or cloud databases, and provides high availability and integrity of records stored in the system. IPFS is the backbone that helps to use a distributed and resilient approach to storing sensitive files that relate to transactions.

D. Blockchain for Immutability and Auditability

The last level of the framework connects the blockchain technology, namely, Ethereum blockchain. Blockchain offers a decentralized recordkeeping that is immutable and transparent. The IPFS-generated CID, as well as metadata to the transaction (transaction ID, timestamp), is stored with the help of the smart contract. When this information is stored on blockchain it is unreliable to modify or remove, ensuring auditing that is tamper-proof and long-term integrity.

The trustless verification process is also introduced by blockchain: any authorized user can access the CID, find the stego-image of the IPFS and receive the ciphertext and decrypt it as well as verify the transaction information without using the internal systems of the bank. This will avoid insider threats and bring out transparency in financial auditing.

E. Motivation for Multi-Layered Security

The technologies applied to this system address the following security issues:

Issue	Solution Provided
Confidentiality	AES Encryption
Detectability of Sensitive Data	Steganography
Single Point of Failure	IPFS Decentralization
Tampering of Logs	Blockchain Immutability

These layers combined form a holistic security architecture which ensures the safety of all the transaction data at each level- creation, storage, transmission and verification.

II. Problem Statement

Despite a greatly advanced digital banking technologies development, the security of transaction records is a critical and unsolved issue. The traditional banking systems are mainly based on central databases, and the traditional encryption methods to save the sensitive information. Nevertheless, these methods have some limitations that are inherent in them. The centralized databases provide a single point of failure which is very vulnerable to the cybercriminals. Any successful breach would potentially

reveal extensive customer information, which would result in significant financial crimes, identity theft, as well as image damages among financial institutions. Besides, insider threats further increase the risk, since privileged users can leave behind traces of manipulated or stolen records via their access without being discovered.

Conventional encryption methods like AES and RSA are able to ensure confidentiality however they fail to hide the presence of encrypted information. Once spotted, encrypted files are likely to be subjected to cryptography and interception attacks. This publicly raises the chances of the attackers engaging in brute-force or statistical attacks to steal the confidentiality of the secured information.

The other weakness of traditional storage systems is that they lack immutability. Malicious internal users or trespassers can modify or remove transaction logs and records without leaving any trace. This absence of tamper evidence is quite dangerous to accountability and it compromises customer confidence on financial services.

Moreover, most available systems do not offer auditable and clear records. Strict regulations like GDPR, RBI guidelines, and PCI-DSS have to be followed by financial institutions, where not only protecting data is required, but also verifiable and reliable audit trails. It is highly difficult to prove compliance, accountability or dispute without auditability.

These problems highlight the need to have an anti-tamper, decentralized, and privacy-respecting security model to address the limitations of centralized security models. This system should ensure confidentiality, integrity, anonymity as well as immutability of banking transaction records and at the same time provide regulatory compliance and auditability. Traditional encryption and centralized storage cannot be expanded to help financial institutions create a more reliable and trustful digital banking environment.

III. Literature Review

The secure management of digital financial transactions has become a prominent area of research as the global banking sector increasingly shifts toward online platforms and remote-access services. With the growing reliance on digital banking comes an equally significant rise in cyber threats, ranging from large-scale data breaches to insider manipulation of transaction logs. Consequently, securing transactional data has motivated researchers to explore cryptography, steganography, blockchain, and decentralized storage technologies. Earlier studies have introduced important foundations, yet existing literature reveals substantial shortcomings in areas such as decentralization, auditability, scalability, and data invisibility. This section critically reviews the major research contributions in these fields and establishes the necessity for the proposed multi-layered framework.

The safe handling of online financial transactions has now become an eminent field of study with the banking industry going more online and resorts to remote access services in the worldwide banking set up. As the use of digital banking increases, so does the number of cyber attacks, both on a large scale, such as data breaches, and smaller, such as insider attacks on the transaction logs. As such, the protection of transactional information has inspired academics to consider cryptography, steganography, blockchain, and decentralized storage solutions. The previous research has also brought significant premises, but current literature has shown significant gaps in the fields of decentralization, auditability, scalability and invisibility of data. This segment is a critical review of the significant works of research on these areas and identifies the need to have the proposed multi-layered framework.

The research of Tabirca et al. [1] is one of the most powerful works in data hiding in financial systems as they proposed a sophisticated method of fractal-based steganography that uses Fibonacci numbers and Discrete Wavelet Transform (DWT). Their algorithm has the ability of storing sensitive banking transaction data as digital images yet visual imperceptibility and strength is very high and has been measured using conventional measures like Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE). In spite of their methodology, which develops the state of image-based steganography, it lacks decentralized storage and blockchain immutability. Therefore, the data contained in the stego-image can be compromised when stego-image is modified or exterminated in centralized storage. Also, their model lacks any cryptographic functionality, which poses a possible security threat considering that steganography is not sufficient to resist advanced steganalysis. Thus, Tabirca et al. make significant contributions to enhancing data invisibility, but their solution is small-scale when it comes to long-term integrity and tamper-evident needs of financial record keeping systems.

Surprisingly, in a similar line of thought, Devadiga et al. [2] came up with a hybrid model of e-banking security that integrates classical cryptography, steganography, and data mining systems to identify fraudulent activities. Encryption of transactional information using AES was done and the encrypted information was subsequently integrated into media files. Though efficient in ensuring confidentiality and concealment of data, their model was completely based on centralized storage architectures, which by nature have multiple single-point failure threats. There is a risk of unauthorized entry, corruption of databases, and insider manipulation with centralized repositories- problems that have been common with actual banking cases. The model by Devadiga also lacked immutable transaction verification, audit logging and decentralized trust mechanisms and thus cannot be used in systems where transparent and tamper free financial compliance is needed. Therefore, although their work has been effective in incorporating several security tiers, it is limited by reliance on central databases, making it ineffective in contemporary distributed banking systems.

To address some of these drawbacks, Aljabery [3] proposed a privacy preserving architecture that incorporates the use of AES encryption, image steganography and the use of a private blockchain technology. Their system encrypts the data of transactions, inserts the ciphertext into images and subsequently stores the resulting stego-images in a permissioned blockchain network. This would enhance tamper resilience and make sure that any secret information cannot be tampered with without being noticed. However,

decentralization and transparency is highly constrained by the fact that a private blockchain limits the control of the blockchain nodes to one organization or a consortium. This decreases the trustless quality of the system, the scale as well as the interoperability. The infrastructures in private blockchains include small and limited distribution of nodes, poor resistance against attacks, and external verifiability. Moreover, the system also lacks decentralized content-addressable storage like IPFS, and thus it is not efficient to work with large media files. Though the model proposed by Aljabery is a step towards decentralized tamper-proof recordkeeping, it has certain limitations: the model lacks scalability and the block validation mechanisms are limited, which makes it unsuitable in a large-scale banking setting.

The other applicable research by Anjum et al. [4] investigated the interaction between the Ethereum blockchain and LSB steganography. Their model has the hash of the stego-image stored on the Ethereum network, which can be later used to determine the integrity of the image. This piece of work proved the usefulness of immutability of the blockchain and decentralized verification processes. Nevertheless, the cost and scalability of storing direct file references or hashes on blockchain is a costly and scalable endeavor because of expensive gas costs and bandwidth. Moreover, their model is not banking transaction specific; it lacks structured metadata (i.e. transaction ID, timestamp, account number) and does not consider the regulatory or audit needs typical in a financial context. Besides, their strategy mostly deals with verification and not security through concealment or distributed storage, and some security aspects remain unaddressed.

So, although Anjum et al. do a great job of combining steganography and decentralized verification, the solution is not quite adequate to the requirements of secure banking transaction recording.

To add to these technical-related contributions, the study of Conti et al. [5] points out the major weaknesses that are linked to blockchain-based steganographic systems. Their activity is devoted to steganalysis algorithms, which can reveal any hidden data based on the analysis of pixel patterns, LSBs, and clustering algorithms used on media files, which are mentioned or stored in blockchain networks. They show that in the case of weak or poorly applied embedding schemes, machine learning and statistical analysis could reveal them even when the files are disseminated over decentralized systems. This supports the idea of multi -layered protection with steganography being reinforced with a robust cryptography, decentralized storage, and verification that cannot be changed. Even though the primary focus of Conti et al. is to present the vulnerabilities instead of defining the possible countermeasures, their results highlight the frailty of standalone steganographic methods and establish the necessity of encryption and blockchain-enhanced architectures.

There are a few gaps that are observed in the overall body of literature. To start with, most of the current systems rely on centralized or semi-centralized storage and are therefore susceptible to manipulation and rogue alterations. Second, cryptography and steganography are sometimes combined in one study, but they do not include any implementation of diverse blockchains, which is essential to ensure immutability and distributed verification. Third, blockchain-based systems technology are limited in scalability, decentralization, and transparency (imperfect private). Fourth, there are limited existing literature that incorporates decentralized content-addressed file systems (like IPFS) that are crucial to storing big files (such as images) in a cost-effective, tamper-evident fashion. Finally, the literature is not specifically relevant to the requirements of banking transactions recording, where auditability, confidentiality, invisibility, and tamper-proof integrity are to be ensured to exist in an integrated system.

In this way, there is a still evident gap in research on the design of an end-to-end secure architecture to synergistically integrate encryption to ensure confidentiality and steganography to ensure invisibility, decentralized storage to ensure distribution-resistance and immutable auditing through the use of a public blockchain.

The proposed work fills this gap by introducing a comprehensive multi-layered framework—integrating AES encryption, LSB steganography, IPFS decentralization, and Ethereum blockchain immutability—to deliver a robust solution for secure, scalable, decentralized, and tamper-proof banking transaction recording.

Author(s)	Year	Technique/ Focus	Limitations
Tabirca *et al.* [1]	2025	Fractal-based steganography (Fibonacci + DWT)	No blockchain integration; lacks decentralization.
Devadiga *et al.* [2]	2017	Cryptography + Steganography + Data Mining	Relies on centralized storage vulnerable to insider threats.
Aljabery [3]	2024	AES + Steganography + Private Blockchain	Private blockchain limits scalability and transparency.
Anjum et al. [4]	2023	LSB Steganography + Ethereum Blockchain	Storage overhead; not optimized for banking transactions.

Conti *et al.* [5]	2021	Blockchain steganalysis	Exposes vulnerabilities in steganography; does not propose mitigation.

Table I. Summary of Related Work on Cryptography, Steganography, and Blockchain-Based Security

This review identifies the fact that although the available studies present viable ways to ensure data security, they are not usually sufficiently scalable, decentralized, and auditable, which is needed to capture banking transactions. This opens the possibility of implementing a new framework that will utilize AES encryption, steganography, IPFS and a public blockchain in order to offer a safer protection of the data.

IV. Existing System

Most of the contemporary data security techniques in storage and transmission frequently involve encryption, steganography, and blockchain to ensure a higher degree of security. Among the most popular techniques is to encrypt the sensitive information with the help of the Advanced Encryption Standard (AES) and subsequently to insert it into the digital photographs by means of steganography. The encrypted message is concealed in particular parts of the image making the stego-image to appear the same as the original image. This way, the information that has to be kept confidential will be hidden without revealing it to the unauthorized audience and at the same time maintain its privacy.

These stego-images are invested in a personal blockchain to ensure further security. The cryptographic hash and use of linked-block construction of the blockchain ensures data integrity and attempts to tamper with the data can be spotted. The framework offers several levels of security by using a combination of these technologies--lessening the exposure of sensitive information, interception prevention and immutable stored records.

Nevertheless, there are significant issues. The application of a private blockchain restricts the property of scaling and decentralization because only authorized nodes are permitted to do so. The steganography block embedding is also a complication and efficiency can be compromised. Above all, these systems are generic security solutions and do not address the specific requirements of the banking transaction systems. There are things that can only be found in real-world banking, such as real-time processing, regulatory compliance, and transparent auditing, missing in the current implementations.

V. Proposed System

The suggested system presents the multi-layered security system that combines four specialized elements such as the AES encryption, LSB steganography, Ethereum blockchain immutability, and IPFS-based decentralized storage to create a secure, scalable, and cheat-proof network of banking registration. In contrast to previous systems that are based on centralized servers or private blockchains, the proposed framework will provide confidentiality, invisibility, decentralization, and immutability all at the same time. The section will provide the system design in terms of theoretical, mathematical and implementation level.

A. AES Encryption Module (Confidentiality Layer)

Advanced Encryption Standard (AES) is a symmetric block cipher that is commonly applied in securing sensitive data, such as banking transactions and communication within the government. AES uses block size of 128 bits using key length of 128 bits, 192 bits or 256 bits. AES-128 is the chosen variant in this system because it is highly resilient to brutality attacks, has an advantageous computational efficiency, and is appropriate when allowing lightweight financial encryption processes.

A banking transaction normally entails::

$$T = \{ A\_src, A\_dest, IFSC, Amt, Time \}$$

This is converted into a serialized plaintext string:

$$P = Serialize(T)$$

AES has to be split into fixed sized 16 blocks of plaintext and padded according to PKCS#7:

$$Pad(P) = P \,||\, b \times n,\ n = 16 - (|P| \bmod 16)$$

1) AES Mathematical Transformation:

To achieve this, AES manipulates a number of transformations to a 4x4 block of bytes known as the state represented as:

$$State \equiv \begin{bmatrix} s(0,0) & s(0,1) & s(0,2) & s(0,3) \\ s(1,0) & s(1,1) & s(1,2) & s(1,3) \\ s(2,0) & s(2,1) & s(2,2) & s(2,3) \\ s(3,0) & s(3,1) & s(3,2) & s(3,3) \end{bmatrix}$$

AES-128 is 10 round-based and the round contains:

1. SubBytes
2. ShiftRows
3. MixColumns
4. AddRoundKey

*SubBytes***:** Non-linear Byte Substitution

$$S(r,c) \leftarrow SubBytes(\ State(r,c)\ )$$

ShiftRows: Cyclic Shift Operation

i Shifts row i to the left:

$$ShiftRows(State) \equiv \begin{bmatrix} s(0,0) & s(0,1) & s(0,2) & s(0,3) \\ s(1,1) & s(1,2) & s(1,3) & s(1,0) \\ s(2,2) & s(2,3) & s(2,0) & s(2,1) \\ s(3,3) & s(3,0) & s(3,1) & s(3,2) \end{bmatrix}$$

MixColumns: GF(28) Matrix Multiplication

All the columns are multiplied by a constant matrix:

$$MixCol\_Matrix \equiv \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

AddRoundKey: XOR with Round Key

$$\text{State} \leftarrow \text{State} \otimes \text{RK}$$

2) AES Encryption Algorithm:

Algorithm 1: AES Encryption
Input : P (plaintext), K (key)
Output : C (ciphertext)

1. P' ← Pad(P)
2. X ← toState(P')
3. RK ← expandKey(K)

4. X ← X ⊗ RK[0]

5. for r = 1 → 9:
   X ← byteSub(X)
   X ← rowShift(X)
   X ← colMix(X)
   X ← X ⊗ RK[r]

6. Final step:
   X ← byteSub(X)
   X ← rowShift(X)
   X ← X ⊗ RK[10]

7. C ← toBytes(X)
8. return encode64(C)

3) CBC Mode AES of the Transaction Data:

AES in CBC (cipher block chaining) mode is used to prevent regularity of pattern:

$$C(0) \leftarrow IV$$

$$C(k) = E\_K(P(k) \otimes C(k-1)), \text{ for } k \geq 1$$

This is to ensure that different ciphertexts are generated by the same transactions.

B. Steganography Module

The second important element in the proposed multi-layer security model is steganography, which will be used to achieve invisibility, even in scenarios where the encrypted financial information is intercepted by the adversaries by the storage medium. Although AES encryption has the effect of converting the transaction data to incomprehensible ciphertext, the ciphertext remains identifiable as encrypted text. The visibility provokes the additional analysis of cryptology or attempts of attacks. In order to prevent this threat, the encrypted data is steganographically hidden in an innocuous looking image file by treating the less significant bits of pixel values, the Least Significant Bit (LSB) steganography, without visual loss.

In contrast to cryptography, which regulates secrecy, steganography regulates concealment and allows to communicate safely without any suspicion. Within the banking system this dual security - of confidentiality + invisibility - at least means that attackers will not even realize that sensitive data is located in a banking system.

1) Digital Image Representation , LSB Theory:

The digital picture is sampled as an intensity matrix of pixels. For grayscale:

$$\Omega = \{ (r,c) \text{ such that } 0 \leqslant r < M \text{ and } 0 \leqslant c < N \}$$

For RGB images:

$$\text{Pixel}(r,c) \triangleq \langle R(r,c), G(r,c), B(r,c) \rangle$$

Value of each channel of pixel is typically provided as a 8 bit integer:

$$R(r,c) \in \{0,1\}^8 \equiv (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)$$

Where:

- b_0= Least Significant Bit
- b_7= Most Significant Bit

The LSB embedding algorithm only modifies the b0 that will have minimal visual distortion.

2) Mathematical Model of LSB Embedding:

Let:

- $C = c_1 c_2 c_3 \ldots c_n$be the AES encrypted message (binary).
- $P = \{p_1, p_2, \ldots, p_n\}$be values of pixel values of the cover image.

LSB embedding equation:

$$b' = (b - (b \bmod 2)) + g, \quad \text{where } g \in \{0,1\}$$

Meaning:

- Destroy the Pixel LSB that is there.
- Replacing it by the ciphertext bit.

Expanded form:

$$b' = (b \,\&\, 254) \mid g$$

Inter-channel embedding In the case of RGB images, embedding is inter-channel:

$$(R', G', B') \Rightarrow \text{set of embedding locations}$$

This distributes the ciphertext evenly and thus it is less noticeable.

3) Better Version of Block-based LSB Embedding:

To enable the project to be robust, the continuous LSB embedding has been incorporated into over a channel, and where feasible, over a row.

If:

$$L\_C := |C|$$

Capacity of the image:

$$\text{Capacity} \triangleq M \cdot N \cdot 3 \text{ bits}$$

Embedding condition:

$$|C| <= \text{Capacity}$$

Which is met by the fact that AES ciphertext is small (~200-400 bits).

4) Steganography Embedding Algorithm:

Algorithm 2: LSB Embedding of AES Ciphertext into Image

```
Input  : Image I, Cipher data C
Output : Stego image S

1. B ← convertToBits(C)
2. idx ← 0

3. for each coordinate (r,c) in I do
     for each channel ch ∈ {R, G, B} do
       if idx < length(B) then
         bit ← B[idx]
         ch ← (ch & 254) | bit
         idx ← idx + 1
       end if
     end for
   end for

4. S ← updated image I
5. return S
```

5) Extraction Algorithm (Reverse Process):

Given stego-image S:

Algorithm 3: LSB Extraction

```
Input  : Stego image S
Output : Cipher data C

1. B ← empty bit sequence

2. for each position (r,c) in S do
     for each color channel ch ∈ {R, G, B} do
       bit ← ch & 1
       append bit to B
     end for
   end for

3. C ← bitsToBytes(B)
4. return encodeBase64(C)
```

6) Visual Imperceptibility Analysis:

The distortions introduced are analyzed using:

Mean Squared Error (MSE)

$$MSE \triangleq (1 / (M \cdot N)) \; \Sigma_{r=0}^{M-1} \Sigma_{c=0}^{N-1} [I(r,c) - S(r,c)]^2$$

Peak Signal to Noise (PSNR) Ratio.

$$PSNR \triangleq 10 \cdot \log_{10} ((MAX^2) / MSE)$$

In order to attain good steganography:

- PSNR > 40 dB
- MSE ≈ 0

The latter is accomplished by virtue of the fact that AES ciphertext is small in your project.

C. IPFS Storage Module

The third significant part of the proposed system is the InterPlanetary File System (IPFS). It deals with one of the most significant constraints of current banking architectures centralized storage. In the old systems, encrypted or steganographic files are stored in central databases or privates servers which can be easily deleted, tampered with by insiders, hardware can fail and can be unlawfully altered. This framework incorporates IPFS, a peer-to-peer distributed storage protocol based on content-based addressing and not location-based addressing, to achieve decentralization, availability and tamper-evident storage. This is to make sure that files are accessed by their cryptographic hash and not their server address.

1) Architecture of IPFS

The IPFS is based on the distributed architecture on the four pillars:

1. Distributed Hash Table (DHT)
2. Merkle Directed Acyclic Graph (Merkle-DAG)
3. Content-Addressed Storage (CAS)
4. Peer-to-Peer (P2P) Routing Mechanism

Any stego-image uploaded to IPFS is chunked, hashed and stored and can be accessed anywhere in the globe with a single Content Identifier (CID).

2) IPFS Process of Data Chunking and Hashing Process

IPFS does on uploading the stego-image::

Step 1: Chunking

A file is subdivided into small blocks (256 KB on average):

$$S \triangleq \langle ch_1, ch_2, \ldots, ch_n \rangle$$

Step 2: Hashing Each Chunk

All chunks are hashed by the SHA-256 or BLAKE3:

$$h\_k \triangleq H(ch\_k)$$

Any change made to the chunk to change the hash would alter the hash entirely and this is due to the hash.

Step 3: Construction of Merkle Tree

All of the chunks are hashed and the hashes combined into a merged Merkle-DAG:

$$Root \triangleq H(h_1 \;//\; h_2 \;//\; \ldots \;//\; h_n)$$

Step 4: CID Generation

The CID consists of:

$$CID \triangleq SHA256(prevHash \;//\; ts \;//\; data \;//\; nonce)$$

Example:

`bafybeia6dz…`

Such CID is the only way to distinguish your stego-image in the whole IPFS global network.

3) Mathematical Characterization of CID

Let:

- D= input file (stego image)

- $H(D)$= cryptographic hash function

Then:

$$CID \triangleq H(M)$$

Where:

$$H(M) \triangleq SHA\text{-}256(b_1 \| b_2 \| \dots \| b_n)$$

Where b_i is representing every chunk of file..

The CID is altered regardless of whether or not a bit in the file is altered:

$$D_1 \neq D_2 \Rightarrow CID(D_1) \neq CID(D_2)$$

This is a must-have property that is used to detect tampering.

IPFS is Necessary in Banking Security:

1. One-Stop Proficiency is eradicated

In centralized servers:

$$Failure(s\_central) \Rightarrow System\_Failure$$

2. Tamper-Proof Storage

In case an attacker alters the file stored:

$$CID\_tamp \neq CID\_orig$$

Therefore, the blockchain record will immediately note the tampering.

3. Persistent Storage

IPFS copies the pieces in numerous nodes.

4. Low Storage Cost

The off-chain storage of images is used and CID is stored on blockchain.

4) IPFS Retrieval Workflow

In the case of retrieving a stego-image:

Step 1: CID is downloaded out of blockchain.

Step 2: Peer discovery via DHT

Step 3: Chunks obtained in distributed peers.

Step 4: Rebuilding a file with the help of Merkle-DAG.

Step 5: Extraction of ciphertext LSB ciphertext begins.

Formally:

S ← rebuild(ch_k)

5) Mechanism of File Availability.

The IPFS follows the following procedures in file availability.

Due to the possibility of being pinned by several peers, the availability will be high, even in the case that the original node is disconnected..

6) Upload Algorithm (Backend Implementation) IPFS

Algorithm 4: Upload Stego-Image to IPFS

Input : Stego image S

Output : Content ID (CID)

1. Divide S into fixed blocks (~256 KB each)

2. for every block blk do

h_blk ← SHA-256(blk)

end for

3. Construct a Merkle structure using {h_blk}

4. rootHash ← top hash of the Merkle structure

5. CID ← rootHash

6. store blocks in local IPFS node

7. share CID within IPFS network

7) IPFS Retrieval Algorithm:

Algorithm 5: Retrieve Image from IPFS

Input : CID

Output : Stego image S

1. peers ← lookupDHT(CID)

2. blocks ← fetchBlocks(peers, CID)

3. for each block blk in blocks do

verify hash of blk

end for

4. S ← assemble(blocks)

5. return S

D. Ethereum Blockchain Module (Immutability & Auditability Layer)

The blockchain element of the proposed architecture offers the required attributes of immutability, decentralization, auditability, and transparency that conventional banking processes do not offer. Even though confidentiality and invisibility are guaranteed by AES and steganography respectively, the two do not eliminate the possibility of file tampering, unauthorized deletion, or post-hoc changes in the transaction records. The Ethereum blockchain overcomes these gaps by acting as a registry that is tamper-resistant and all the records of transactions are permanently stored and cryptographically authenticated. Ethernet is selected since it supports smart contracts that can be used to provide programmable transaction logic and storage of metadata including timestamps, transaction IDs, and IPFS CIDs. In contrast to the overall system of the private blockchain that was employed in the literature in the past [3], Ethereum offers public verifiability, or any authority can independently verify that a banking transaction record is valid and untouched.

1) Blockchain Architecture

Etherium is a decentralized registry, which is supported by thousands of computers worldwide. Each node maintains a replica of ledger and it is involved in validating transactions.
A blockchain block contains:

$$Block \triangleq \langle idx,\ ts,\ prevHash,\ TxList,\ rootHash,\ nonce \rangle$$

Each block is chained using:

$$H\_block \triangleq H(\ header(Block)\ )$$

$$prevHash(Block\_k) \triangleq H(\ Block\_\{k-1\}\ )$$

Such hashing chain provides immutability.

2) Banking System Smart Contract Structure.

The system proposed applies a smart contract named: *BankTransactionRegistry:* It has mappings of the type:

public ledger mapping (uint - TxRec);

Where:

```
struct TxRec {
   string contentID;
   uint time;
   string transactionID;
}
```

Each record contains:

- CID → stego-image stored in IPFS
- Timestamp → Exact time to process
- TxID → Unique transaction id

Mathematical representation:

Let:

$$R\_k \triangleq \langle CID,\ ts,\ txID \rangle$$

Smart contract storage:

$$Ledger \triangleq \{\ R_1, R_2, \ldots, R_n\ \}$$

Once inserted:

$$R\_k \rightarrow \text{immutable}$$

3) Ethernet Life Cycle of transaction:

In case of a submission of a transaction:
Step 1 — Transaction Creation
Includes fields:

$$Tx \triangleq \langle\ nonce,\ gasPrice,\ gasLimit,\ receiver,\ value,\ payload\ \rangle$$

***Step 2* —** Signature
Using ECDSA:

$$\sigma \triangleq Sign\_ECDSA(\ sk,\ H(tx)\ )$$

Step 3 — Networks Broadcasting
Sent to peer nodes in order to verify.
Step 4 — Miner/Validator Execution
Smart contract function is performed:

storeTx(txID, CID, ts)

Step 5 — Block Inclusion
After confirmation, transaction is included to the block.
Step 6 — Immutability

$$H(Block\_k) \triangleq f(\ H(Block\_\{k-1\})\ )$$

Any modification of the record disrupts the chain.

4) Hashing and Merkle Trees

Ethernet employs the Merkle Patricia Trees to store contract states.
One of the transactions is included in the Merkle tree:
Merkle Root

$$root \triangleq H(\ h_1\ //\ h_2\ //\ \ldots\ //\ h_n\ )$$

If any leaf changes:

$$root' \neq root$$

So when one is tampering, it cannot be detected.

5) Consensus Mechanism

Ethereium has Proof of Stake (PoS) which guarantees:

- Decentralization
- Energy efficiency
- Byzantine fault tolerance

The blocks are signed by known validators who put money in ETH.
Security Guarantee: In case attacker desires to do history anew.
Security Guarantee: If attacker wants to rewrite history:

$$Stake\_adv\ >\ 0.51 \times TotalStake$$

This is financially impractical.

6) Smart Contract implementation Algorithm

Algorithm 6: transaction metadata on blockchain
Input : txID, CID, ts
Output : permanent record

1. payload ← createPayload(txID, CID, ts)
2. sig ← signWithKey(payload)
3. submitTx(payload, sig)
4. on contract execution:
   ledger[txID] ← ⟨ CID, ts, txID ⟩
5. include transaction in a block
6. wait until block confirmation (finality)
7. return success

7) Unalterable Audit trail on Banking transaction:

The main benefit of Ethereum usage is immutability:
In case a bad insider wants to tamper with any record:

$$H(R_1) = H(R_2)$$

Because the information stored in blockchain is immutable:

$$R_i \text{ remains unchanged forever}$$

Thus the system ensures:

- Non-repudiation
- Tamper-proof logging
- Transparent auditing
- Public verifiability

8) Ethereum Over Private Blockchain

Feature	Private Blockchain	Ethereum Public Blockchain
Decentralization	Low	Very High
Transparency	Restricted	Global
Tamper Resistance	Moderate	Maximum
Trust Model	Authority-based	Trustless
Auditability	Limited	Public
Security	Depends on owner	Distributed across thousands of nodes

E. Workflow of the Proposed System

The proposed system will be a four-layer hybrid security system that will convert transaction data into a cryptic, decentralized, and immutable digital value. The workflow implies a series of transformations that ensure the confidentiality (AES), invisibility (LSB steganography), decentralization (IPFS), and immutability (Ethernet blockchain). The next layer reinforces the first one, and this is a unified system which is designed to overcome all the underlying vulnerabilities of the current banking systems.

1) Complete Workflow Steps

Step 1: Data Acquisition of transaction
The usual components of a banking transaction are:

$$T \triangleq \langle \text{srcAcc, destAcc, IFSC\_code, amt, ts} \rangle$$

This information is gathered at the front end interface.

Step 2: AES Encryption
Data on the transactions is sequential:

$$P \leftarrow \text{encode}(T)$$

AES-128 in CBC mode encrypts it:

$$C \leftarrow \text{Enc\_AES}(P, K, IV)$$

The output ciphertext:

- Cannot be interpreted
- Has no meaningful pattern
- Without the key it cannot be undone

Step 3: Steganography Implementation
Ciphertext C is converted to bitstream C_{bits}:

$$B \leftarrow \text{toBits}(C)$$

The cover image I is modified:

$$I'(r,c) \leftarrow I(r,c) - (I(r,c) \bmod 2) + B[k]$$

Where I' = the stego-image.

Step 4: Storing Stego-Image to IPFS
The stego-image is broken down, hashed and stored in distributed nodes.
IPFS computes the CID:

$$CID \triangleq \text{SHA-256}(\text{blocks}(S))$$

CID develops into the permanent reference.

Step 5: Blockchain Submission
Ethereum contract stores:

$$Rec \triangleq \langle CID, txID, ts \rangle$$

This history can never be changed once mined.

Step 6: Retrieval
To verify:

1. Fetch CID from blockchain
2. Get stego-image from IPFS
3. Bring out cipher text by the LSB decoding.
4. Decrypt using AES and key

Reconstruction:

$$P \leftarrow \text{Dec\_AES}(C)$$

2) FOUR-LAYER MATHEMATICAL INTEGRATION OF THE FOUR LAYERS

Layer 1: Encryption

$$B \leftarrow \text{bits}(P)$$

Layer 2: Stego Mapping

$$S \leftarrow \text{embed}(I, C)$$

Layer 3: IPFS Addressing

$$CID \triangleq H(S)$$

Layer 4: Registration of Blockchain

$$\text{Entry} \triangleq \langle CID, txID, ts \rangle$$

The combined operation is:

$$\text{System}(P) \triangleq \text{Ledger}(H(\text{embed}(\text{Enc}(P))))$$

This compound function is injective and fingerprinted.

*3) **END-TO-END SYSTEM ALGORITHM (FULL SYSTEM)***

Algorithm 7: Complete Secure Transaction Recording System

Input : Transaction data T
Output : Hash of a blockchain (CID, txID, ts)
1. M ← encode(T)
2. K ← genKey()

3. IV ← genIV()
4. C ← Enc_AES(M, K, IV)
5. B ← toBits(C)
6. S ← embedLSB(I_cover, B)
7. CID ← addToIPFS(S)
8. txID ← pushToChain(CID, ts)
9. return ⟨ CID, txID, ts ⟩

4) Security Analysis

The hybrid structure offers security of defense-in-depth.

Confidentiality (AES Layer): AES-128 ensures

- Opposition to the rugged force
- Non-linear substitutions
- Diffusion properties & confusion properties
- Security against known-plaintext attacks

Even if stego is detected:

$$\text{Attacker} \Rightarrow \text{no access to key}$$

Invisibility (Stego Layer):

- There is minimal visual distortion
- PSNR > 40 dB
- Small statistical attack detectability.
- Even in case captured by attackers:

Even if attackers capture the image:

$$\text{appearance}(S) \approx \text{appearance}(I)$$

Decentralization (IPFS Layer): IPFS defends against

- File deletions
- Central server failure
- Unauthorized modifications
- Insider attacks

If file changes:

$$CID_{new} \neq CID_{ref}$$

The blockchain immediately flags mismatch.

Immutability (Blockchain Layer): Blockchain gurantees

- Zero modifications
- Zero deletion
- Zero forgery
- Verifiable timestamp

The tampering cannot take place due to:

$$H_{block} \triangleq H(\text{header(Block)})$$

Any alteration of field transforms the whole chain.

5) System Architecture Model

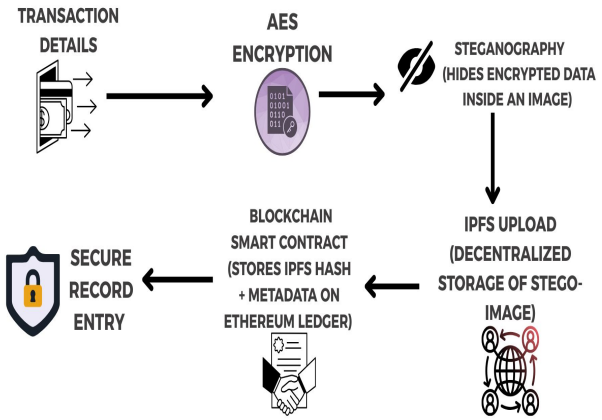


Fig.1. Flow diagram of Secure Banking Transactions using Blockchain

VI. Implementation Design

The suggested framework is structured into a multi-step process that combines the use of cryptography, steganography, decentralized storage and blockchain technology in guaranteeing the safe storage of banking operations. It all starts by gathering of the transaction details including account number, IFSC code, transaction amount, and time in order that are organized and are ready to be processed in a secure way.

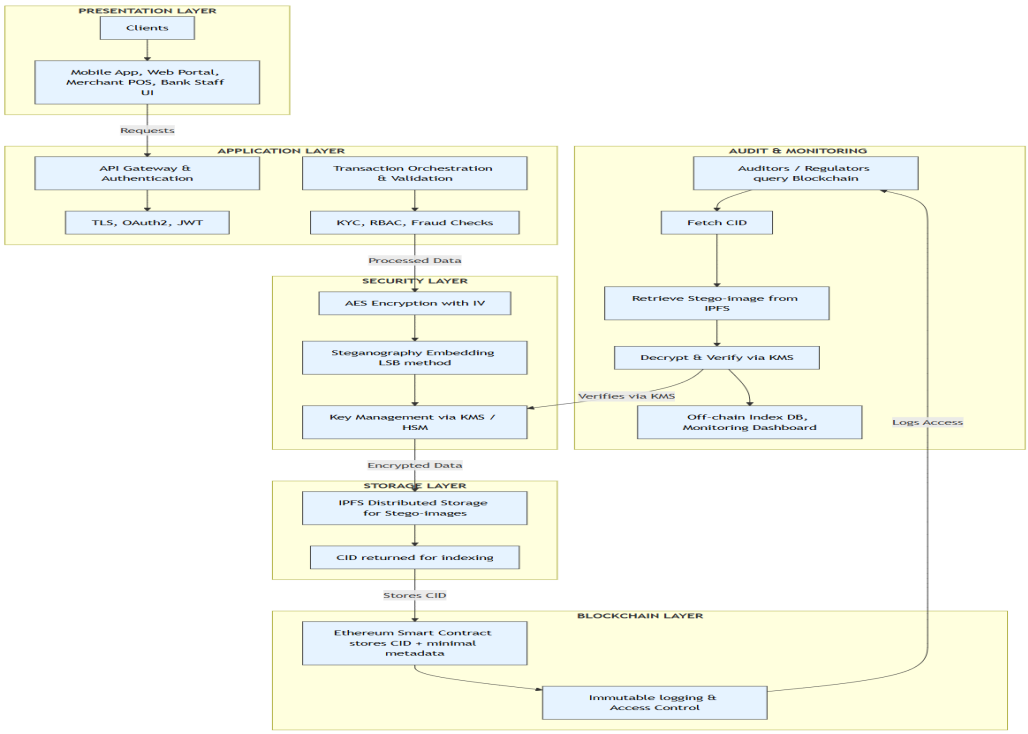


Fig.2. Detailed Architecture diagram of Secure Banking Transactions using Blockchain

Phase 1: Encryption

i. The Advanced encryption standard (AES) is used to encrypt the transaction details.

ii. Each transaction has a unique Initialization Vector (IV) that is used to promote increased security and discourage pattern recognition..

iii. The end result is a ciphertext that offers privacy to the original transaction information.

Phase 2: Steganography

i. Steganography Least Significant Bit (LSB)-based steganography is used to hide the ciphertext in a lossless cover image.

ii. This creates a stego-image which looks exactly like the original thus concealing the sensitive information and minimizing the risk of targeted attacks.

Phase 3: Decentralized Storage (IPFS)

i. The stego-image will be uploaded into the InterPlanetary File System (IPFS), which is a decentralized storage platform using a peer-to-peer approach.

ii. IPFS uses a Content Identifier (CID) that acts as a verifiable and permanent reference to the stored item.

Phase 4: Blockchain Recording

i. The little metadata (such as transaction ID and timestamp) are stored on the Ethereum blockchain through smart contracts and can never be changed.

ii. This ensures that storage is tamper free, auditability and decentralized checking of the reference to the transactions.

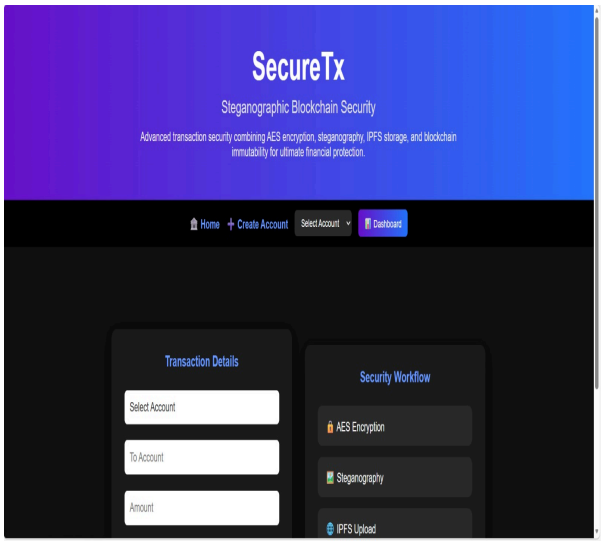
Retrieval Process

i. During the retrieving process of the blockchain CID, the stego-image is retrieved using its equivalent in the IPFS.

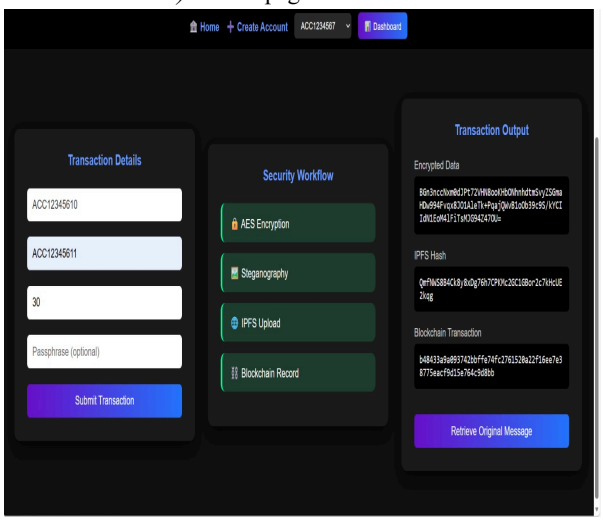
ii. The ciphertext is extracted by steganography decoder and is subsequently decrypted by the assistance of AES that restores the initial information of transactions.

The architecture guarantees the confidentiality, invisibility, decentralization, immutability and integrity of records of digital banking transactions in addition to the vulnerabilities of the centralized and privately operated blockchain-based systems.

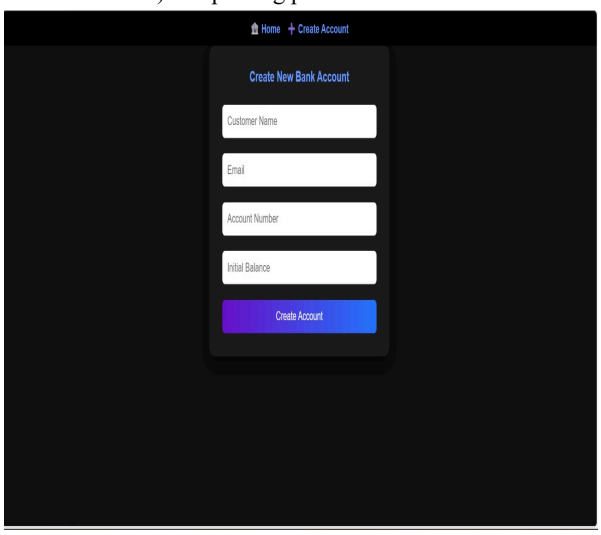
Fig.2. Output



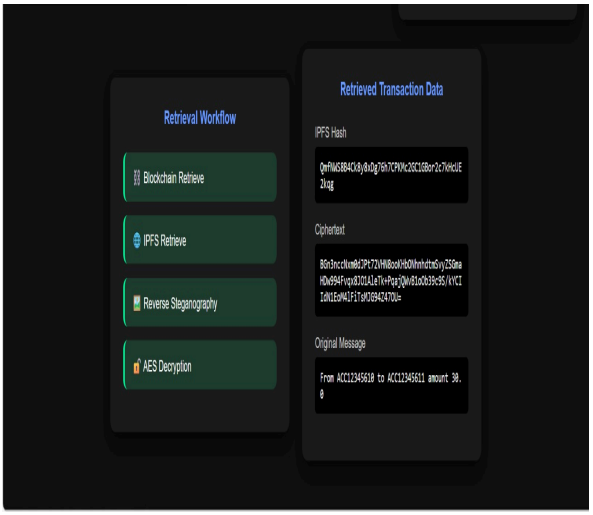
a) Home page



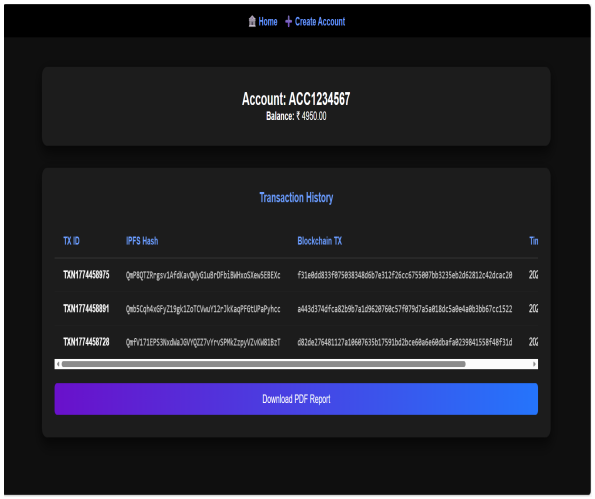
c) Uploding process



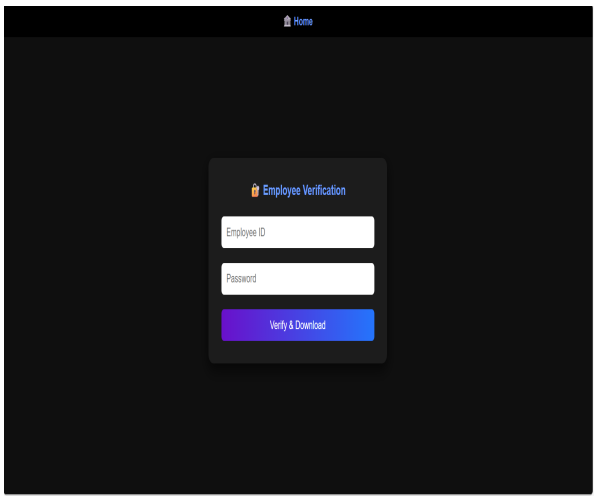
b) Create account



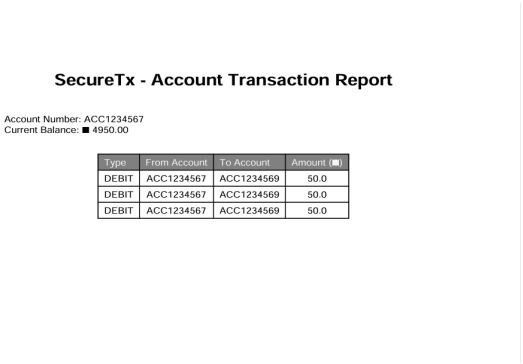
d) *Retrival process*



e) Transaction record dashboard



f) Password Authentication



g) generated final report

iii. Comparative Analysis

Feature	**Existing System** (AES + Steganography + Private Blockchain)	**Proposed System** (AES + Steganography + IPFS + Ethereum Blockchain)
Data Security	AES encryption with steganography	AES encryption + steganography + decentralized storage
Storage	Centralized / Private blockchain nodes	Distributed via IPFS, accessible through CID
Transparency	Centralized / Private blockchain nodes	Distributed via IPFS, accessible through CID
Scalability	Restricted due to private blockchain constraints	Scalable with IPFS and Ethereum integration
Tamper Resistance	Based on private ledger consensus	Immutable via Ethereum blockchain consensus
Confidentiality	Encrypted data, but visible existence of storage records	Encrypted + hidden inside images, reducing visibility to attackers
Auditability	Limited, requires access to private chain	Verifiable and transparent through smart contracts

Table II. Comparative Analysis of Existing and Proposed Systems

iv. Expected Advantages

The system of the project has several advantages over the existing secure storage systems since it is a combination of encryption, steganography, IPFS, and blockchain. The principal benefits are estimated as being:

A. Confidentiality

The sensitive banking transaction information is encrypted through the AES encryption mechanism to ensure that it is decrypted using the corresponding cryptographic key.

B. Invisibility

Steganography conceals the encrypted message in the digital pictures making it indistinguishable with the ordinary files on them and reducing the likelihood of interruption.

C. Decentralization

The IPFS offers a very high availability and resilience by making sure that the data is ultimately scattered and spread all over a peer-to-peer network thus, has no points of failure.

D. Immutability

CIDs are stored in the Ethereum blockchain which provides the guarantee that when the records of the transactions are recorded, they cannot be changed or removed.

E. Auditability

The blockchain provides an exact verified account of the transaction, which is auditable and regulation compliant.

Together, all these advantages create an unbreakable and safe system of providing security in the accounting of the banking dealings, which will additionally build trust and transparency in the financial systems over the internet.

v. Applications / Use Cases

The proposed framework is not banking specific which implies that it can be applied in other diverse fields where safe, unaltered, and un-disclosed record management is an essential consideration. These significant uses are:

A. Digital Banking Transactions

Keeps customer records of transactions, in case of fraud and unauthorized access (safe keeping).

B. Financial Audit Trails

Refers to deliver non-revocable transaction histories in order to meet regulation entities and audit.

C. E-Governance Systems

Secure personal citizen data such as identity documents, transactions and land records of the public services.

D. Healthcare Data Management

Facilitates the protection of confidentiality of the medical records through storage and distribution of patient data.

E. Intellectual Property Protection

The data of the stores: digital assets of copyright are stored in decentralized storage which is tamper proof.

F. Supply Chain and Trade Finance

Maintains records of international trade, logistics, and transaction that could be verified and secured, hence, transparency.

These applications suggest how the proposed system will be capable of enhancing trust, secrecy, and integrity in the aspects where the sensitive information must be guarded against manipulations and unauthorized access..

vi. Ethical and Privacy Implications

A. Importance of Protecting Customer Data

Banking transactions are characterized with highly sensitive financial and personal information. Any form of unauthorized access may violate cyberspace and cause fraud and identity theft, not to mention mistrust.

Being a moral necessity and a professional requirement within a digital banking system, data confidentiality, integrity, and availability will be provided.

B. Compliance with Regulations

The proposed structure meets the international and the local legislation such as:

i. **General Data Protection Regulation (GDPR) -** is a document, which offers safe data storage, limited access, and user permission of data utilization in the European Union.

ii. **Reserve Bank of India (RBI) Guidelines** - emphasize on localizing data, the possibility of creating a secure record of transactions, and the powerful encryption of digital transactions. These regulations assure technical security, legal and moral responsibility.

C. Transparency vs. Privacy

Even though blockchain can ensure transparency, it must achieve this by balancing it with customer privacy by encrypting sensitive data before being stored.

Encryption such as AES, steganography and access control smart contract can be utilized to protect the exposure of personal banking information and yet allow auditing.

vii. Future Enhancements

A. Post-Quantum Cryptography Integration

Quantify-resistantize AES to prevent future quantum computing attacks on banking data.

B. AI/ML for Fraud Detection

Use a machine learning model to analyze the transactional data to identify any fraud or anomalies.

C. Steganography Advanced Techniques

Deep learning methods to improve loss resilience in steganography to steganalysis and image compression.

D. Zero-Knowledge Proofs (ZKPs)

Make transactions checkable without exposing sensitive data and this provides privacy to blockchain-based banking.

E. Scalability Improvements

Optimize IPFS and blockchain to reduce storage overhead and latency in banking transactions with great volumes.

F. Banking Support Association between borders

Enlarge the system of international transactions of multi- currency and international finances regulations.

viii. Conclusion and Future Scope

The paper gives a multilevel defense mechanism of documenting banking transactions on AES encryption and steganography, IPFS, and Ethereum blockchain. The suggested solution design would solve critical defects of current systems since it offers the chance of being confidential with encryption, being invisible with steganography, being decentralized, with the assistance of IPFS, and being immutable with blockchain. As these technologies have been incorporated, the design framework has helped to protect, integrate and audit the sensitive financial data to rule out the apprehension of utilizing centralized solutions as well as standardized cryptographic techniques.

Though the provided work mainly presupposes the model of a safe and decentralized storage of the transactions, it may provide the guidelines in the further implementation and improvements in reality. Moreover, it is possible to add machine learning-based anomaly detection in the future to the model to identify suspicious behavior and post-quantum cryptography to circumvent new threats. In addition, the latency, storage capacity, and scalability aspects of the performance will also be tested to confirm the functionality of the performance of the framework in the large-scale banking environment.

The proposed model will be useful in addressing this paper by coming up with a secure, tamper free, and a technologically improved digitized banking system..

Acknowledgment *(Heading 5)*

The authors would like to extend their sincere thanks to their faculty mentors and the Department of Computer Science and Engineering for guiding them in the development of the research work. The authors would also

like to acknowledge the availability of resources to shape the idea into a conceptual framework.

REFERENCES

[1] A. I. Tabirca *et al.*, “Enhancing Banking Transaction Security with Fractal-Based Image Steganography Using Fibonacci Sequences and Discrete Wavelet Transform,” *Fractal and Fractional*, vol. 9, no. 2, p. 95, 2025.

[2] N. Devadiga *et al.*, “E-Banking Security Using Cryptography, Steganography and Data Mining,” *Int. J. Comput. Appl.*, vol. 164, no. 9, pp. 1–6, 2017.

[3] N. Palanivel, K. Madhan, A. Venkatvamsi, G. Madhavan, S. B and L. Priya G, "Design and Implementation of Real Time Object Detection using CNN," 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2023, pp. 1-5, doi: 10.1109/ICSCAN58655.2023.10394752.

[4] S. Anjum *et al.*, “Blockchain Based Image Steganography,” *Int. J. Adv. Comput. Theory Eng.*, vol. 8, no. 1, pp. 25–29, 2023.

[5] M. Conti *et al.*, “Steganographic Analysis of Blockchains,” *Sensors*, vol. 21, no. 12, p. 4078, 2021.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[6] H. V. Patil, “Crypto-Stego: A Hybrid Method for Encrypting Text Messages or Text Files within Images Using AES and LSB Algorithms,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 23s, pp. 2780–2785, Dec. 2024.

[7]M. R, N. P, N. L. Christy S, M. K, P. N and M. U, "Integrating CNN and Random Forest Algorithm for Multi Satellite Image Compression in Data Mining," 2024 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2024, pp. 1-5, doi: 10.1109/ICSCAN62807.2024.10894190.

[8] A. Chaudhary, A. Sharma, and N. Gupta, “Designing a Secured Framework for the Steganography Process Using Blockchain and Machine Learning Technology,” *International Journal of Intelligent Systems and Applications in Engineering*, 2024.

[9] V. Raja and K. S. Suresh, “Deep StegBlock: Deep Learning-Enhanced Steganography for Secure Communication in IoT Devices Using Blockchain,” *Educational Administration: Theory and Practice*, vol. 30, no. 4, pp. 2958–2972, 2024, doi: 10.53555/kuey.v30i4.1963.

[10]J. P. R, A. A, M. K, P. N, G. V and A. S. S, "Generalized Discriminate Analysis for Classification Algorithms in a Tuned Machine Learning Model for Steganalysis," 2024 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2024, pp. 1-5, doi: 10.1109/ICSCAN62807.2024.10894458.

[11]S. Sharma and P. Kumar, “An Enhanced Hybrid Cryptographic Technique Using AES and RSA for Secure Data Transmission,” *International Journal of Computer Applications*, vol. 178, no. 7, pp. 10–14, May 2019.

[12] L. Xu, C. Xu, and Z. Li, “Research on Image Steganography Based on Improved LSB Algorithm,” *IEEE Access*, vol. 8, pp. 103739–103749, 2020.

[13] A. Kalla, A. Alenezi, and M. A. Abdrabou, “Blockchain and IPFS-Based Framework for Secure Healthcare Data Sharing,” *IEEE Access*, vol. 9, pp. 106776–106788, 2021.

[14]P. N, E. P, M. K, S. P. T, S. K. C and S. K. R, "Enhanced QR CODE Scanning and Blockchain Technology for Drug Packaging System," 2024 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2024, pp. 1-5, doi: 10.1109/ICSCAN62807.2024.10894169.

[15]Y. Zhang, J. Wen, and X. Luo, “Blockchain-Based Secure Data Storage and Sharing for Banking Applications,” *Future Generation Computer Systems*, vol. 134, pp. 116–127, 2022.