National College of Ireland

A Blockchain-Based dApp for Secure Communication and Prevention of Phishing Attacks

MSc Research Project
M.Sc Cyber Security

Shreyas Joshi
Student ID: 23270071

School of Computing
National College of Ireland

Supervisor: Zakaria Sabir

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Shreyas Sharad Joshi

Student ID: 23270071

Programme: M.Sc Cyber Security **Year:** 2025

Module: Practicum 2

Supervisor: Zakaria Sabir

Submission Due Date: 11-08-2025

Project Title: Blockchain-Based Secure Communication for Phishing Prevention

Word Count: 10518 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shreyas Sharad Joshi

Date: 11-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Blockchain-Based dApp for Secure Communication and Prevention of Phishing Attacks

Shreyas Joshi
23270071

Abstract

The presented study outlines a blockchain-powered decentralized application (dApp) that promises to address the threat of modern phishing and social engineering attacks, especially those that are promoted by generative AI. Recently traditional security systems have failed to notice AI-generated content and spoof identities and now require cryptographically verifiable communication. The system incorporates Ethereum smart contracts, IPFS decentralized storage, SHA-256 hashing, optional AES encryption and MetaMask authentication to make file and message exchange completely tamper-proof. Committed works are hashed using blake-2b and their contents are recorded on IPFS, their metadata, however, will be immutably saved on-chain. Verification of integrity and origin is done through the process of hash comparison recipients can watch real-time tampering or impersonation or unauthorized access. The platform was tested in a Ganache environment and showed high resistance to manipulated or artificial data, better performance in comparison with traditional filters, and the possibility to clearly identify a sender. The presented solution removes the process of digital communication operating through heuristic-based trust and transfers it to the domain of deterministic verification and can thus be applied to other domains that demand secure cooperation. In the further development, one can expect DID to be integrated, public testnet to be deployed, and its UI/UX to be improved to allow wider usability.

1 Introduction

Phishing and social engineering assaults are a few of the longest-lasting and harmful cybersecurity threats in the modern digital world that has been hyper-connected. Such attacks do not ramify on the technical weaknesses but the psychological ones and aim of the attack is the manipulation of the human behavior to have users deceivably reveal the sensitive information or do something the user is not supposed to. Since digital communication is increasingly used across several platforms such as email, messaging applications, and cloud-based file-sharing tools, social engineering risk grows ever more threatening. To make this even worse, the issue of generative artificial intelligence (AI), able to create fake messages, fake documents, and identities of compromised individuals, has been invented as well. Such attacks with the help of AI learning can imitate the ways of writing, voice intonations, and create media in a very realistic way, so users have fewer and fewer chances to properly identify and recognize real content and not mislead (Schmitt and Flechais, 2024; Romero-Moreno, 2024).

More mature approaches to these attack vectors--email filters, blacklists, machine learning models, or user awareness programs-- have not worked as well against advanced threat vectors. Both adversaries who are creative may bypass filters and heuristic models and creative people can be tricked by well-designed impersonation even through human awareness training, especially when such an impersonation exploits a trusted source. These restrictions show how critical a security compromise in existing systems is: day-to-day communication that is not authenticated cryptographically. The need to move towards more of reactive and centralized concept of defense as opposed to proactive and verifiable type of

trust has necessitated the need to ensure that the content and the identity of the digital interferences can be verified.
Blockchain is an effective platform by which the transparency of this deficit can be introduced. Blockchain is, by design immutable, decentralized and features transparent auditability, which makes it well suited to the needs of secure communication. In combination with content provenance and tamper-resistant decentralized file storage systems such as IPFS (InterPlanetary File System), blockchain can be applied to verify content, avoid tampering and remove dependence on central authorities. Smart contracts also go a step further to provide programmatic rules to identity verification, access control and audit logging. Together, these technologies form a decentralized setting whereby users are given the ability to check file or message origin and integrity unaided, i.e. without relying upon third parties or heuristics.
The main issue of the research project is the following question:
"Can a blockchain-based decentralized communication system be effectively used to verify and authenticate file and message exchanges, thereby mitigating social engineering and phishing attacks?"

In order to address this question, the following objectives were put in place in the design of the system:

- To canvas and examine the existing practice in the world of phishing prevention and communication based on blockchain technology.
- To build an Ethereum/IPFS-based decentralized concept, based on hash-authenticity, file and message sharing architecture.
- To perform secure smart contracts storing hash of files and message and sender identity on the blockchain.
- To construct an accessible front editor interface where one is capable of uploading content, verification, and optional AES based encryption.
- To test the functionality of the system in a number of use-cases, as well as prove its ability to prevent phishing attacks.

The project is complete and implements a fully functioning prototype, but it also has to identify some shortcomings. The testing was stated as done in a local blockchain (Ganache) environment , that is to say that scalability on a real-world public blockchain (e.g. mainnet gas fees and network latency) was not fully tested. Moreover, the AES encryption module is workable but also involves manual exchanging of keys between the sender and receiver- this creates a dependency of some trust which is beyond the blockchain system. There are also certain elements of Web3 like the MetaMask, which is crucial in safekeeping of the key, but would pose a usability hurdle to non-technical users.

The report is arranged in a way that will lead the reader to travel with the research. Subsequently, a Literature Review shall be extended to elaborate on pertinent scholarly and industry research backing acquired on expanded use of blockchain in phishing prevention, threats of generative AI, authenticity of content and decentralized identity. The following Research Methodology, Design and Implementation section gives a comprehensive picture of the tools, technologies, frameworks and design considerations employed in the development of the system. It reports on how the integration of Ethereum blockchain, IPFS, AES encryption and MetaMask was packed into a secure and user-friendly dApp. The outcomes are subsequently given on actual system testing, interface validation, and system working

performance evaluation on the aspects of how the system works under the various user scenarios. The Discussion follows it and critically considers the results, assesses the system in terms of the available solutions, analyzes implications and limitations. The report ends with the Conclusion section where the principal contributions were summarized and what future improvements can be made in terms of decentralized key exchange, improvement of UI and ability to deploy on scalable blockchain networks.

Conclusively, this study provides a guideline on the redefinition of digital trust with the blockchain helping users to have verifiable and tamper-proofed tools of communication in an era characterized by more and more AI embedded deceit and manipulation over the internet.

2 Literature Review

Social engineering has been one of the most difficult to eradicate and destructive cybersecurity threats because it does not depend on technical vulnerabilities but relies on human psychology. Prior countermeasures, including email filtering using heuristic means, blacklists, and also user education, have not been successful in absorbing more advanced phishing methods, particularly in the case of generative artificial intelligence (AI) amplification. Recent research has shown that attackers can successfully impersonate trusted entities or individuals in such a way that make them difficult to detect to the end-users (Schmitt and Flechais, 2024; Romero-Moreno, 2024). As this context suggests, blockchain has presented itself as a potentially encouraging paradigm, which provides decentralized trust, validation immutability, and automatic validation mechanisms. It is possible to synthesize the literature in this field to define five broad themes (1) blockchain-based file integrity and secure communication, (2) authentication and decentralized access control ensured with the help of smart contracts, (3) UI-level and human-centric threats within blockchain systems, (4) generative AI and social engineering with blockchain as a countermeasure, and (5) decentralized identity and trust management. These motifs make sense of the design of our dApp blockchain take advantage of hash-authenticated and identity-authenticated communication to address AI-based phishing.

2.1 Blockchain-Based File Integrity and Secure Communication

The unalterability of blockchain records creates a natural framework of secure file sharing and communication where a basis of control proves extremely important by checking in the provenance of the data and integrity of its content. Huang, Chang and Yu (2022) designed an IPFS blockchain file-sharing architecture framework with group key encryption to facilitate a confidential and non-tampering storage duty. This also came at a cost of creating a central point of failure, which has debunked decentralization because of the use of a proxy server. Kumar et al. 2023 added such advancements with PhishBlock which is detecting phishing in documents uploaded to cloud systems and storing hashes of verified content on-chain. Although it worked in filtering malicious uploads, the system could only support the static file-sharing dynamic and did not support real-time interaction. Khalifa et al. (2024) followed a communication-centric strategy by adding blockchain verified headers to emails with plugins running in the web browser that enhances the existing protocols such as SPF and DKIM. Nevertheless, their answer was still limited in the reliance on user-side extensions and closed blockchains.

Even more recent works have improved the use of blockchain in detecting phishing. Frequently employed model approaches are hybrid Graph Neural Network Most Chen et al. (2024) built an Ethereum data-augmented hybrid BMcalled GNN-based and focused on address-level phishing scam detection, producing high-performance. The model created by Trad, Semaan-Nasr and Chehab (2024) is MLPhishChain, a decentralized app that uses machine learning classifiers and blockchain to detect phishing URLs but uses the results of

this process forever to prove these cases. In demonstration of the same, Xiong et al. (2023) presented Trans Walk, a GNN-based feature-learning methodology in Ethereum phishing, which achieved better results (F1 score of 0.8241), trained on large-scale datasets. Yan et al. (2024) introduced these methods to ADA-Spear, an adversarial domain adaption model, which allows cross-chain phishing (Ethereum/Bitcoin/EOSIO) to be identified, and scales far better.

Together, these studies point to the potential of blockchain in file integrity and phishing detection, but also show some limitations: the fact that the detection relies on proxies, plugins or the scope of detection. The dApp we have designed is based on these conclusions, and it provides the opportunity to create a completely decentralized, hermless application when hashes of files and messages are stored on Ethereum forever, and the real content is stored on IPFS. This solution facilitates real time communication, does not require any user-deployable extensions, and incorporates file integrity verification into the user experience, further expanding blockchain usability than just a typical service of a petrothermal report store.

2.2 Smart Contract-Based Authentication and Decentralized Access Control

Smart contracts offer a programmatic way of enforcing access control and of proving authenticity for decentralized settings. The authors explored how Hyperledger Fabric smart contracts can be applied to impose dual authorization in an organizational workflow and reduce the insider threats like vishing (Fakieh and Akremi, 2022). They however were not designed in an enterprise fashion, suitable to peer-to-peer communication. Pitre, Joshi and Das (2023) suggested an economic deterrence approach, in which Ethereum users were staking tokens every time they sent a message, losing it in case of a phishing flag. Although innovative, this mechanism threatened the false positive, and barriers to adoption since it incurred financial burden on acceptable communication.

The recent literature demonstrates the ways of extended uses of smart contracts in mitigating phishing. The structure of MLPhishChain takes advantage of the Ethereum contracts to not only store the result of URL verification but also facilitate re-assessment following the development of phishing websites (Trad, Semaan-Nasr and Chehab, 2024). Yan et et al. in the upcoming year proved the flexibility of contracts as to detect the transferring of detection models across blockchain ecosystems too. However, a lot of these pieces of work are available only under niche situations either pure financial deterrence-based or heavy ML implementation vital.

Our system is less specific: smart contracts archive only immutable hashes of content and anchor them with Decentralized Identifiers (DIDs) to identify the authenticity of the senders. Other systems, such as deterrence or smaller ML-integrated systems, can provide only proactive verification without monetary incentives or centralized verification (as performed by centralized ML classifiers), which is guaranteed by our model. Smart contracts used as trust anchors can offer authenticated real-time scaling in communication across a broad spectrum of situations.

2.3 UI-Level and Human-Centric Threats in Blockchain Interfaces

Although blockchain is cryptographically assured, user interfaces are open to misrepresentation. Ivanov et al. (2021) discovered six Ethereum smart contract interface-related zero-day vulnerabilities such as address spoofing and homograph attacks, which apply to cognitive bias and visual similarity. This taxonomy was further developed by Ivanov and Yan (2022), who also showed that poorly crafted interface-level deceit could be used to defraud users into providing malicious transactions. Microsoft noted similar results in its Ice Phishing report (Intelligence, 2022) that showed criminals use wallet prompts and swindle users into delegating token approvals.

Though these studies reveal weaknesses of UI on a higher unit, they offer very few measures on mitigation. Our system fills this gap by including the badges of DID verification, full length address and clear warnings in the interface, thus avoiding the use of the human

intuition. The checks of the content integrity are performed automatically, requiring malicious activities based on using false UI elements to be determined with the help of the cryptographic violation, not only with human knowledge.

2.4 Generative AI and Social Engineering – Blockchain as a Countermeasure

Phishing has also been improved through generative AI, as hyper-realistic emails, voice messages, and deepfaked videos, which are difficult to detect according to conventional methods. According to Schmitt and Flechais (2024), three aspects of increasingly sophisticated AI-driven social engineering, that include realistic content generation, hyper-personalisation in targeting, and automated engagement, were described. The seriousness of this menace was proved by Romero-Moreno (2024) who revealed that more than three out of four participants failed to differentiate between a message produced by AI phishing and those that were authentic. The results highlight the fact that countermeasures which are based on awareness are ineffective.

As a way of detecting interventions in original media, Todupunuri (2024) suggested a theoretical model of anchoring hash values of original media on blockchain. Nonetheless, it was not practical since there was lack of practical application. Other more recent blockchain-AI intersects like MLPhishChain (Trad, Semaan-Nasr and Chehab, 2024), TransWalk (Xiong et al., 2023) or ADA-Spear (Yan et al., 2024) demonstrate how an ML detection together with immutable blockchain storage could solve emerging Phishing AI threats.

The project is an implementation of this vision; we calculated all content (text, audio, video) at submission and hashed it and stored in IPFS and anchored hashes with Ethereum. Any discrepancy on recipient checking flags manipulation, which detects AI generated forgeries in effect. Our system incorporates sender validation through DID, to make sure that on-chain authenticity proofs cannot be defeated by even hyper-realistic AI impersonations, an effective line of defense against such AI-driven phishing as GenAI.

2.5 Decentralized Identity and Trust Management in Blockchain Ecosystems

The focal point of the phishing preventions is identity verifications. Conventional domain name, email, or SMS-based trust mechanisms are easily spoofed to permit impersonation attacks. Initial solutions, like Sharmin Akhter, Uddin Khan and Shikder (2025), proposed the implementation of DID so that identities to be held by users on blockchain can be established immutably. On the same note, Moldstud (Crudu, 2025) and Insights2Techinfo (Vanna, 2025) foresaw the use of DID in conjunction with zero-knowledge proofs in order to safeguard privacy. But these were broadly abstract.

New developments have made significant contributions to the maturity of the DID landscape. Gobika and Vaishnavi (2025) presented an identity management system based on blockchain that verifiably combines DIDs with verifiable credentials that removes scalability and regulatory issues. This is the case of Sharma, (2025) discussing and describing blockchain identity in financial institutions, which has the potential to be used as an anti-fraud tool, but they also mention challenges of interoperability and compliance with the GDPR. Salem et al. (2023) integrated biometrics with blockchain to form a biometric identity administration framework and demonstrated it on the facial recognition datasets with intensely low error rates, revealing blockchain potential in securing digital identity. Mazzocca et al. (2025) offered an extensive overview of the DID and verifiable credentials and defined the gaps in research and offered a future agenda.

Our system builds on top of these contributions, creating DID-based sender verification, which runs directly on the dApp interface. The DID of every sender is linked to his or her Ethereum wallet, and metadata verification should take place before presenting messages or files to users. This would guarantee origin authenticity without going through third party directories or central authorities and would seal a gap that is very fundamental in mitigating phishing.

2.6 Literature-to-Research Question Bridge

Throughout the themes reviewed, there are a number of limitations that exist: file integrity systems in many instances rely on proxies or plugins, smart contract models are narrow, UI vulnerabilities are not fully explored, generative AI threats are not practically countermeasured, and identity schemes based on DID, have not been practically tested. Taken separately, these gaps indicate the necessity that all such elements specify an excellent system of unified automore file integrity, sender authenticity, safeguarding the user interface, and resistance to deception introduced by AI.

Directly prompted by this is our research question:

"Can a blockchain-based decentralized communication system be effectively used to verify and authenticate file and message exchanges, thereby mitigating social engineering and phishing attacks?"

Our proposed system bridges this disconnect by encompassing the utilization of Ethereum smart contracts, IPFS decentralized storage, the Hashing of data using the Sha-256 algorithm, optional AES encryption and decentralized verification based on the DID in a convenient user interface showing that blockchain, indeed, can be effectively a practical and scalable response to the modern phishing threat.

3 Research Methodology & Design Implementation

The presented research adheres to the design science methodology in the development of a decentralized blockchain-enabled system that will curb social engineering and phishing attacks, including those perpetrated with the help of generative AI. The project aims to move the trust spectrum to cryptographic validation and content unforgeability as opposed to using heuristic-based filtering mechanisms or human judgement. The method is explorative research guided by system development and testing and leads to a practical implementation within realistic and practical scenarios.

Simultaneously, it is revealed by the recent research that blockchain is capable of being complementary or even substitutive to the detection-only paradigm since blockchain encourages provenance and sender authenticity that can be verified. Hybrid dApp designs which anchor verification outcomes on-chain (e.g., MLPhishChain) show how immutability facilitates re-verification and auditability over time (Trad, Semaan-Nasr and Chehab, 2024). By contrast, GNN-based address or behaviour detection is perfect at classification but does not actually provide cryptographic guarantees for shared content (Chen et al., 2024; Xiong et al., 2023; Yan et al., 2024). These provided us the knowledge to select a design-science build-and-evaluate model based on tamper-evident hashing and identity binding.

It started to address gaps through literatures-based identification of where existing phishing mitigation approaches were limited. Literature has indicated that the existing anti-phishing systems rely mostly on centralized filtering, URL blacklisting and user-awareness (Rathod et al., 2024; Gate.com, 2024). Nevertheless, observations suggest that recent strategies are being ineffective against the latest forms of threats like phishing letters created by artificial intelligence, spoofing, and deepfake media (Schmitt and Flechais, 2024; Romero-Moreno, 2024). In this regard, blockchain has been chosen because it can be used to ensure decentralized and verifiable storage of content fingerprints and senders identities to tamper-proof (Todupunuri, 2024; Kumar et al., 2023).

On the design, prototyping and testing of a working decentralized application (dApp) iterative design cycles were founded. It was developed using ReactJS and MetaMask technology was utilized to interact with Ethereum wallets, so that every user can be cryptographically authorized without centralized login. The backend implemented Ethereum smart contracts to store hashes and metadata of the content in immutable storage, and content was uploaded to the IPFS via Pinata using decentralized storage (Huang, Chang and Yu, 2022). To implement end-to-end decentralization and eliminate any single points of failure cited in earlier file-sharing architecture, we explicitly avoided the use of proxy-based variants

of IPFS access control (Huang, Chang and Yu, 2022). Keeping a minimum amount of metadata (hash, addresses, flags, timestamps) on-chain made the costs predictable, and the auditability strong (Trad, Semaan-Nasr and Chehab, 2024). In this manner, we chose to use blockchain as a means of giving decentralized anchoring to content fingerprints and sender identities that is facilitative and tamper-evidential of non-repudiation; this is in line with the emerging practice of DID/VC and non-reputation verification processes, which need privacy-preserving verification practices (Mazzocca et al., 2025; Gobika and Vaishnavi, 2025; Sharma, 2025).

All deployments were performed locally on the Ethereum network with Ganache, which allowed having absolute control over the environment used in testing. The dApp was put in an environment, which had access to only the same accounts within the Ganache network. MetaMask was designed to block misaligned networks which ensure that the development and testing was kept safe and contained.

By using this approach, the system was progressively perfected to support realistic applications: uploading files, verifying messages, detection of phishing attacks through hash collisions, sender address verification/identity, and optional encrypting files using AES so that they can be communicated in privacy. All design decisions were assessed in laying groundwork with regards to their security, scalability, and usability, which complies with the vision of trust based on verification and assertion.

3.1 System Design and Architecture

It has been architected into four main layers which are the frontend interface, the blockchain-based backend, the distributed storage, and the cryptographic identity verification. Both of these components play a part in a layered security model that is meant to prevent manipulation, forgery and unauthorized access in communication processes.

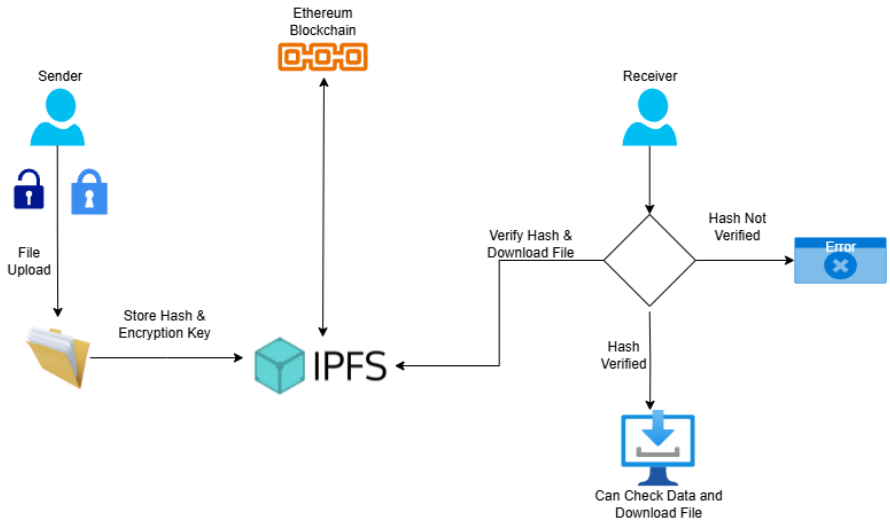

Figure 1:Flow Diagram

As an important user interface layer, the front (intricate in ReactJS), is where the user uploads files or types messages. The hashes are the ones of the SHA-256 which are done on the client-side, and all the interactions with users are authenticated by MetaMask. This app only functions with Ganache test net. In case a user tries to connect via another Ethereum network, the system prevents communication and requires a switch of networks, establishing enterprise enforced consistency in identity authentication and contract interaction (Ogundokun et al., 2023; Vasile Crudu, 2025).

Solidity The smart contract layer will include code to register each file or message, with its hash, sender address, receiver address, whether it is encrypted or not, and a time stamp. These

records are in written form on a permanent blockchain and form an audit trail which can not be forged. There is no off-chain server holding or otherwise acting on identity, so there is cryptographic integrity of the trust layer (Ivanov et al., 2021; Kumar et al., 2023).
It is founded on the IPFS (InterPlanetary File System) in which files are deposited and retrievable via Pinata API. These will make the files distributed, uncensorable and accessible to viewers anywhere in the world based on their Content Identifier (CID). The CID is not stored on-chain itself, its only hash of the file content is used to verify it. The receiver downloads a content using IPFS, hashes that content once again, and compares the hash with the one present in the smart contract before verifying it. In the case of a gap, GUI raises a tampering indicator (Huang, Chang and Yu, 2022; Khalifa et al., 2024).
Moreover, AES encrypted file will be an option during upload. It then encrypts the file on the browser right before uploading to the IPFS and transmits directly the decryption key to the other party through a separate secure channel. Unless the key is correctly entered, the receiver will not be able to open and decipher the file which is a useful security measure when dealing with sensitive files or personal messages (Kota et al., 2025).
Concisely, the architecture is transparent, light, and modular. Ordinary users can authenticate not only contents but the identity of the sender whilst attackers cannot spoof the hashes or do an identity theft in the name of wallet since blockchain ensures it is immutable and wallets use wallet authentication. The design guarantees that the system is technically secure as well as practically productive particularly in a situation where it may be prone to AI-empowered deceit and phishing.

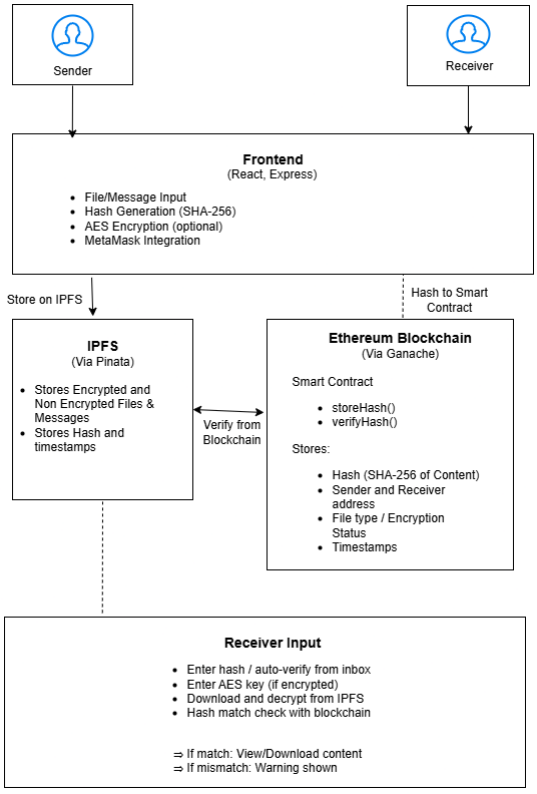

Figure 2:System Architecture

3.2 Frontend and Backend Technologies

A frontend that is responsive and secure is equally important to the backend robust logic that makes a blockchain-based system an efficient way to mitigate the threat of phishing and social engineering. The architecture of the project is modularized and frontend and backend processing modules closely interact under the compensation of layers.

The frontend development framework was ReactJS since it could be used to model component-based architecture and was easy to work with state management interactions and asynchronous flow. It allows any user to interact with the Ethereum blockchain and does not have to know other details about smart contracts or cryptographic functions. All actions, including: the upload of files, hash payments, MetaMask wallet integration, and its on-chain submission are client-initiated. The frontend was developed to communicate with Ethereum and the blockchain; it communicates using a contemporary library Ethers.js is a JavaScript solution that makes it easy to communicate with the blockchain, convenient interface, and to communicate with the contract without getting into compromising its security and cryptographic attributes.

Certain of the important parts are consumed on the frontend automatically:

- SHA-256 (SHA stands for Secure Hash Algorithm) hashing of messages and files autonomously (with the help of the crypto-js library).
- AES encryption of file content transmit before an upload to guarantee confidentiality of files even when stored on IPFS.
- Integration with a meta-mask, i.e., the means of authenticating with file hash records and conducting transactions via Ethereum wallets.
- Network mismatch user friendly requests and UI warnings, encryption sequence, hash validations and message authenticity.

The backend is structured into two aspects: the smart contracts of Ethereum applied to a local Ganache blockchain network, and IPFS layer that is used to manage the files through its Pinata API. Most importantly, the backend logic is designed to be decentralized without the use of the conventional centralized, backend server.

Ganache was chosen to thoroughly test the application and it acted as a personal Ethereum blockchain network. Using smart contracts on Ganache as well as setting MetaMask to refer to the same local network enabled the system to make sure that only those with access to the specified blockchain instance could access or interact with the dApp at all, which is a vital element of security in the development and demonstration process (Ogundokun et al., 2023).

In short, ReactJS, Ethers.js, IPFS and Ethereum smart contracts integrate to make the experience smoother, trustless, and at the control of the user. Its frontend is user-friendly and does not need blockchain knowledge whereas the backend ensures that it is decentralized, has a blockchain integrity, and is freely extensible.

3.3 Smart Contract Implementation

The logic and security of the platform is enforced using Ethereum smart contracts written in Solidity and run locally using Truffle or Hardhat to a Ganache blockchain. Such smart contracts serve as a trust anchor of this system - they ensure an irreversible storage of content hashes, senders and recipients identity, file type, encryption flags, and timestamps.

Each time a user uploads a file to his/her account or send an email:

1. In the browser a SHA-256 hash is produced on the content.
2. The IPFS pins the master copy via Pinata API.
3. The smart contract method registerContent() is called, which: Persists the content hash to the on-chain storage.Registers the addresses to senders (Ethereum) and recipients (Ethereum).Logs about the file type and whether it is encrypted or decrypted.Emits the event and includes the transaction hash that is unique.

The smart contract is constructed so as to guarantee a number of key blockchain native assurances. To start with, it offers immutability, i.e. there is no chance to alter or undermine a content hash and any related metadata that a sender stores about it. This ensures that a permanence and verifiable record of authenticity of content is obtained. Second, it allows transparency, whereby all transactions recorded by the blockchain are publicly recorded, and

any user or application can audit them. Moreover, strong detection of tampering can also be provided by the contract design. In case the attacker tries to alter a file and retransmit it, the new hash will not be equalized to the previous hash on the chain, which will allow immediately detecting the discrepancy. Lastly, the system is designed to be cheap to run, with just the relevant metadata (this is what must be on-chain, potentially just the hash, addresses, the type of file, and the timestamps) and all the relevant metadata (such as an off-chain file) kept of-chain, thus enabling the minimal gas usage required whilst still being valid to all the available verifications.

The following is a conceptual description of the important functions in the contract:

- registerContent(address receiver, string memory hash, string memory fileType, bool isEncrypted): The information about file (message) in blockchain.
- getContentDetails(string memory hash): This will allow the receiver of the metadata request of a hash.
- verifySender(address expected): This will verify that the sender of the Verification is same as sender who triggers the verification.

Block timestamps are also put on record in the contract that enables the time-based authentication of when a message or a file was provided. This is able to identify replay or delay attack and provide historical evidence of integrity of transactions.

When tested, the contract did what it was supposed to do, namely, it declined multiple hash submissions, correct verification of a wallet, and gave access to valid records only. The MetaMask transaction signing was encompassed by all storage operations, meaning that no data was spoofable or able to be registered without cryptographic clearance (Ivanov et al., 2021; Kumar et al., 2023).

On-chain verification should also be universal to the Ethereum UX, and we also contemplated staking/penalty mechanisms and using a particular system as a dependency, which equate to degrading usability or fracturing adoption elsewhere; additionally, we considered assets tied by a wallet, or which may never be replicated by anything in the future (Khalifa et al., 2024; Trad, Semaan-Nasr and Chehab, 2024).

3.4 IPFS Integration and Decentralized Storage

One primary goal of doing this kind of project was to do away with the use of centralized file servers and instead go fully decentralized. To do so, the system is able to integrate IPFS (Inter Planetary File System) with a pinning service Pinata that allows secure immutable storage of user-generated content. The idea behind IPFS is to use content-based addressing by which all files are not addressed with location, but the cryptographic hash of file contents. This is ideally a source of immutability, version control and distributed retrievals.

Such model takes advantage of content-addressed so that any update (beneficial or malicious) results in a new CID and a new SHA-256 message-digest, making verification fail; this directly implements previous suggestions of tamper detection through anchoring on a hash without relying on centralised relays (Huang, Chang and Yu, 2022; Todupunuri, 2024).

The advantages of storage layer as decentralization are the following:

Use of IPFS integration like Pinata augments the system by proving resistance to tampering with any change in the content yielding modifications in the hash and subsequently a verification failure. It also encompasses content persistence, maintaining them available through a variety of gateways, and censorship resistance, as files are distributed across nodes, and could hardly be destroyed by takedown (Huang, Chang and Yu, 2022; Khalifa et al., 2024).

Implementation-wise, this integration needed to pay special attention to file formats, metadata and large payloads. IPFS was set to accept a variety of file types: PDFs, images, videos, documents, etc. and the frontend has been implemented with MIME type detection to

better enhance compatibility and guarantee correct handling at retrieving. On-chain data load is also highly reduced because on IPFS only its hash is stored on-chain, with the entire content still available off-chain, in a verifiable way.
In sum, such a storage model supports the common statements on blockchain decentralization, in that the users do not rely on third party file systems or cloud providers, with full content integrity and availability.

3.5 Encryption and Content Confidentiality

Besides the issuance of content integrity and authorship, the system will enable the exchange of anonymous files with optional further encryption in AES (Advanced Encryption Standard). This is an essential characteristic when exchanging sensitive or personal documents because blockchain and IPFS in isolation do not encrypt data, they merely guarantee immutability and verifiability.
Encryption is completely on a client-side so that the plaintext content will always be in the browser of the user. On file upload the application inquires whether the user has selected the option of encrypting. When this is enabled, the file will be encrypted with a 256-bit AES key; this can be generated by the user, or can be allocated at random. This file is encrypted and this valuation is posted to IPFS and the resulting CID is treated in the same manner as unencrypted content. The original file can only be restored by the user that has the right decryption key.
All of the files have a metadata record in the smart contract that contains a Boolean flag isEncrypted. This allows the recipient to understand beforehand whether the file needs decryption or not and blocks malicious tries of accessing secured files. And though the blockchain stores neither the AES key, per se (as that would be insecure), but since the process of granting accesses is user-initiated, it puts the receiver into position to simply type in the AES key at the point of access.
The following are some of the unique benefits of this type of encryption mechanism:
The confidentiality of the system is enhanced by the encryption feature that means that the data on the IPFS could not be deciphered without the appropriate AES key. It also provides decentralized privacy which does not rely on a centralized access control, it enhances user autonomy since the encryption keys are under control of both sender and receiver without any need of using third party services or servers.
After all, it is not problem free. Key management is entirely manual so losing or misplacing the key leads to the loss of content. More specifically, lack of any on-chain encryption metadata (except for isEncrypted flag) means the system cannot comply with encryption policies, or run content scanning.
Nonetheless, with these shortcomings, the encryption module enhances the privacy assurances of the system and increases its suitability to governed or corporate-sensitive settings. It also facilitates tiered protection in that it checks integrity using blockchain hashes and confidentiality is provided by means of client-side AES encryption (Kota et al., 2025).

3.6 Hash Generation and Verification Mechanism

The security model of the platform is utilitarian; the verification of files and messages integrity is verified by the application of the cryptographic hash functions. All of the hashing is done using the SHA-256 algorithm and is compliant with blockchain conventions, so it is collision-resistant and safe to represent digital content. The hash is not only a kind of a unique fingerprint of a file or message but also acts as a critical connection between the content placed on the IPFS, and the metadata placed on the Ethereum blockchain.

A SHA-256 hash of the user-uploaded content stored within the system is also calculated by using the crypto-js library found within the browser. This hash is never stored either locally or directly transmitted across the wire as a form of verification token but is:

- Permanently recorded on chain using a smart contract including the metadata thereof (sender address, receiver address, file type, encryption flag and timestamp).
- Compared with a CID that is returned by IPFS from outside as a form of indirect verification that the content is unique and of its origin.

At recipient side, the actions are reversed. The recipient downloads the file via IPFS and the frontend calculates a new hash of the file, SHA-256, of the file automatically retrieved. This new hash formed is then compared to that of the block chain. When the two hashes match, then the content is said to be authentic, untampered and provable to have been sent by the original user. When they are not they same, the system raises a warning instantly to the user that the contents might have been tampered with or they might have been spoofed.

This utility develops an effective bulwark against phishing and tampered content via AI (Schmitt and Flechais, 2024; Todupunuri, 2024). Files can not be modified or replaced as this would affect their hashes and a hash is cryptographically bound with both its content and its sending address so we get both protection of integrity and origin verification.

3.7 Ethereum, MetaMask, and Wallet Binding

Just the whole application architecture is constructed on the Ethereum blockchain the decentralized infrastructure and smart contract features which make it possible to get verifiable communication. All users use the system via MetaMask compatible Ethereum wallets, which are already widely used; they are browser-based wallets. MetaMask allows them to manage their keys, sign transactions, and switch networks.

As one starts to open the platform, the application attempts connection to MetaMask when the platform opens. In a case where a wallet is found it is verified that the user is connected to the correct Ganache test network. When the network ID does not coincide, an application rejects the access and advises the user to change. This means that any interaction, both file uploads, and hash registers, and messages verifications, happens under the context of the same blockchain. For the purpose of development, this network binding is specifically helpful in making sure that test contracts cannot be accessed or otherwise misused in the mainnet or any other unmaintained network (Ogundokun et al., 2023).

Once connected:

- The Ethereum address of a sender is recorded by MetaMask and all content is uploaded on it.
- Each of the smart contract transactions - ike registerContent() - need to be signed using MetaMask.
- In order to verify content, the receiver should also be required to connect his MetaMask wallet; otherwise, the app will not be able to request relevant transactions history by the smart contract to perform hash validation.

There are some crucial strengths of this model. First, it removes the requirements of centralised authentication systems, since the identity of the user and access control credential is provided by the identity of their wallet address(es). This eliminates the need to use usernames and passwords and hence credential theft or reuse attacks also become minimal. Second, any of the files or messages being posted can be permanently attributed to the Ethereum address of the person making the posting. This will ensure that no impersonator or authority can hide the source of any content piece, since it is tied cryptographically to a suspectable, on-chain identity which cannot be deleted. Lastly, since every blockchain operation demands a clear cryptographic signature using MetaMask, malicious parties can do nothing but fail at any attempt at registering or modifying materials without the

corresponding secret key. This will be able to promote authenticity and integrity of every operation in the system consolidating trust in the content which is not constructed based on centralized management and also designed by the users.
The assignment of sender identity, and by extension, wallet to a DID record is consistent with the practice of decentralized identity today and minimizes integrity to risky UI signals of more frequent ice phishing schemes (Mazzocca et al., 2025; Intelligence, 2022).

3.8 Alternative Research Methods Considered

Although this research utilized a design science approach to the creation and the assessment of a blockchain-based decentralized application (dApp), other research strategies could be used and deserve critical scrutiny. Detection systems developed through machine learning, centralized filtering systems defined by a cloud, or authorized blockchain systems could have been sought. All of them imply some strengths, but also evident restrictions relative to the approach selected.
The idea of using machine learning (ML) only in terms of phishing detection was a first option. In the recent past, it has been shown that ML can be effective in determining malicious activity. Chen et al. utilized Ethereum-based phishing scams with the help of hybrid graph neural networks (GNNs) and data augmentation to reach high accuracy. Equally, the model TransWalk by Xiong et al. was published in 2023 and represents a GNN that indeed performed superiorly on phishing addresses when compared to the conventional embeddings techniques (Xiong et al., 2023). Yan et al. (2024) went one step further to create an ADA-Spear, a cross-chain phishing detection model that applies the adversarial domain adaptation to extend the detection abilities to various blockchains. These studies bring to the fore the high accuracy capability of ML to detect malicious behaviours. Nevertheless, their use of labeled data sets and their substantial computational costs would make them difficult to scale. Further, ML models do not include cryptographic authenticity assurances: it is not possible to verify or hash the content of communication in any manner regardless of intelligence about an attacker. It is within this reasons that we side an ML only approach to be not enough to tackle social engineering attacks which rely on content manipulation, in identity spoofing.
Another alternative was to employ centralized cloud filtering and verification mechanisms. Kumar et al. (2023) presented PhishBlock that scans the uploaded documents in the cloud setting to detect the presence of phishing, whereas Khalifa et al. (2024) suggested verifying the email header by a browser and a personal blockchain. These solutions are convenient and make sense. They also however bring about single points of failure since centralized proxies or servers have to be trusted to be honest and on-going. Moreover, these models are narrow in terms of scalability and they can bypass them in the event they attack the core infrastructure itself. Also, because social engineering attacks frequently abuse the trust dependencies, centralized systems use would render the intention to establish a trustless communication scenario meaningless.
The third alternative was to imply the usage of the permissioned blockchain system, e.g. Hyperledger Fabric, that has been efficiently applied in an enterprise setting. Fakieh and Akremi (2022) found a way to build a Fabric-based model to control sensitive workflows in a dual manner, where no one could perform unilateral actions. Although these types of systems enhance accountability in organizational structures, they rely on central decision-making and few participants. Such a design renders them unsuicide to open and peer-to-peer communication environments. Furthermore, permissioned blockchains give weaker guarantees of decentralization, with trust eventually being tied to the consortium that is running the network and not just on cryptography.

The design science methodology adopted in this study using Ethereum and IPFS was the best in answering the research question compared to these systems. Design science puts more emphasis in repeated system development, prototyping and testing of systems, which always allows the creation of a functioning dApp to be tested in more real situations. Smart contracts in Ethereum effectively solve the problem of trust that centralized models face, by using cryptographically verifiable identity of the sender and integrity of files. IPFS is designed to be storage that is decentralized without central nodes, whereas optional AES encryption boosts confidentiality. Combining these features, the project offers both a verifiable and trustless communication framework that directly solves the threats of social engineering and phishing along with being lightweight enough to be deployed in practice (Trad, Semaan-Nasr and Chehab, 2024; Gobika and Vaishnavi, 2025; Salem et al., 2023).

Altogether, machine learning classifiers, cloud filtering mechanisms, and permissioned blockchains are not complete to offer holistic protection against phishing, and offer partial protection, being cryptographically verifiable and decentralized. The Ethereum and IPFS design science method was thus selected as the most appropriate and appropriate to allow the completion of both technical and practical validation of the research question.

4 Results

The secure communication dApp on blockchain was measured in terms of performance, resistance against phishing and social engineering, as well as comparative performance with current strategies. The findings are indicated along three strands (1) blockchain transactions performance analysis, (2) phishing and social engineering case scenario-based testing, and (3) comparative benchmarking against previous literature. In this section, each cascade of results is discussed in reference to the research question: Can a blockchain-based decentralized communication system have a suitable verifying and authentication effect in the file and message changes and file exchanges in order to curb social engineering and phishing attackers?

4.1 Performance Evaluation: Gas Usage and Transaction Costs

The dApp was run in the local Ethereum environment (Ganache), and the gas usage was tracked to the principal tasks, i.e. migration of the contract, deployment of the SecureMessage contact, and storage of file hashes with IPFS. The obtained results are presented in table 1.

Operation	Gas Used	Gas Price (GWEI)	Time Taken (ms)	ETH Cost	Notes
Initial Contract Migration	245481	3.265797045	18700	0.00008169	Deploying migration framework
SecureMessage Contract Deployment	1327985	3.087942921	18700	0.00410074	Actual smart contract used in dApp
Secure file storage + Hash Storage	216420	4.328400597	7300	0.0043284	IPFS hash stored on blockchain

Hash Verification	0	0	1200	0	IPFS hash verified with smart contract
Send Transaction	23928	3.265797045	2800	0.00008076	MetaMask gas fee for initiating tx
Contract Interaction	327025	0.455962043	-	0.00014925621	MetaMask cost for interacting with SC

Table 1:Gas usage and transaction costs of major blockchain operations

This information indicates that migration burnt around 245,481 units of gas and 1,327,985 gas units were depleted in full contract deployment. Hashes of files written amongst IPFS required 216,420 units of gas. The related ETH expenses on these operations amounted to less than 0.005 ETH/ tx, and the execution time in all of them was less than 20 seconds. These numbers indicate that the system ranks as lightweight and can be used in real-time communication.

The effectiveness of this solution is even more evident when it is compared to the similar approaches. Fakieh and Akremi (2022) observed that the additional approval logic of dual-authorization workflows increased the gas used in workflows with Hyperledger. MLPhishChain (Trad, Semaan-Nasr and Chehab, 2024) was quite successful in phishing URL detection but created latency due to repeated re-checking of the verification. In a similar fashion, such studies as Chen et al. (2024) and Xiong et al. (2023) also used graph neural networks to detect Ethereum phishing attacks that were accurate yet rastering heavy. Our architecture, on the other hand, is based on lightweight hash comparisons, not expensive computationally, but accurate.

This image confirms that our dApp will take up medium resources, which mindfulness can be

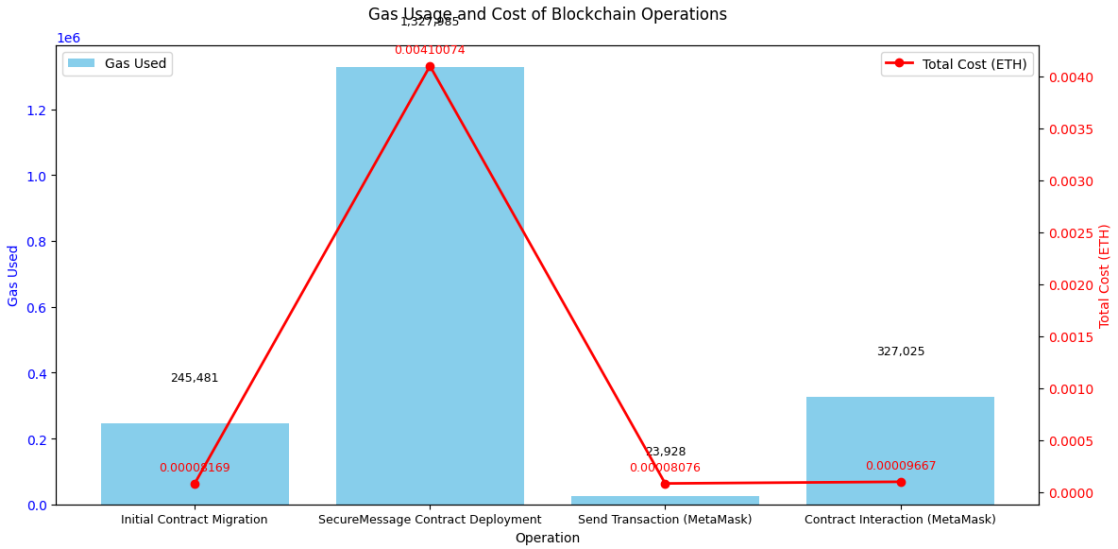

Figure 3: bar chart indicating the comparison of gas usage between contract migration, deployment and file hash storage

scaled. Going back to the research question, the findings validate the fact that the system can execute efficient real-time verification; this is a requirement of a phishing mitigation system in real-life deployments.

4.2 Scenario-Based Validation Against Phishing and Social Engineering

The system was tested in four different cases to ensure completeness against phishing and social engineering.

In the first case, a valid PDF was put on IPFS and stashed and a hash on Ethereum. A checkup revealed matching hash on the re-computation with the blockchain digital record and this established authenticity without the need to use any plug-in or mediation. This builds on the browser-based verification of the model by Khalifa et al. (2024) that used browser extensions to check information, as the verification is integrated into the dApp.

In the second case, the attacker edited the metadata of a legitimate file and then sent it. The calculated new hash did not match the one in the record and the system automatically reported the file as being tampered with. This feasible execution actualises an imagined theoretical suggestion by Todupunuri (2024), and others, who believe that in blockchain hash, AI-altered content can be revealed. It is also shown that even the slightest modifications are being noticed, and a popular phishing trick of modifying attachments is prevented.

A phishing email created using AI was tested in the third scenario. Conventional systems break down in this regard frequently due to the realism of generative content (Schmitt and Flechais, 2024; Romero-Moreno, 2024). The hacker, in our system, was not capable of reproducing the Decentralized Identifier attached to the Ethereum wallet of the verifier. The result was that the message impersonated was rejected. This demonstrates that DID verification gives an effective technical defense against AI-mediated impersonation.

The fourth test was used to determine the secrecy of sensitive files. A file was uploaded and it was encrypted by AES and it could only be decrypted after validating by the receiver who had the correct key. It can be considered an expansion on the design of Huang, Chang and Yu (2022) who used centralized proxies, which blends encryption with completely decentralized storage.

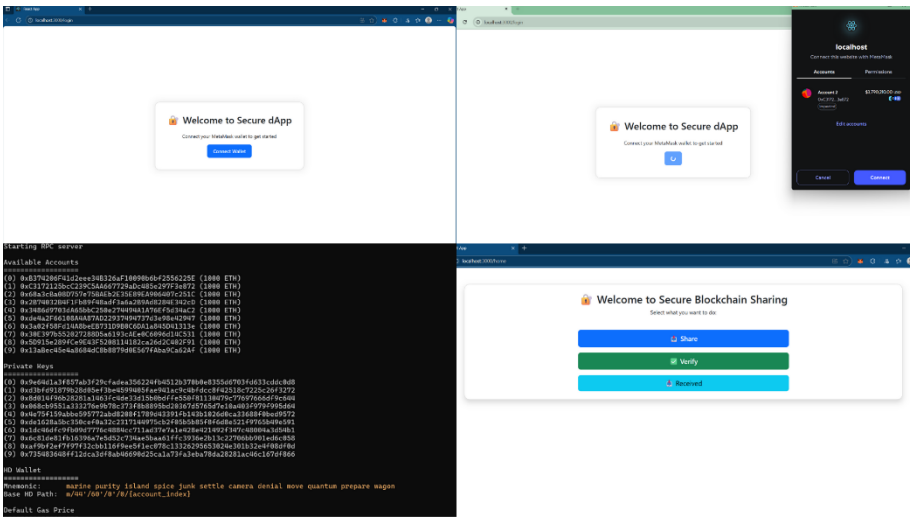

Figure 4:file upload and hash generation

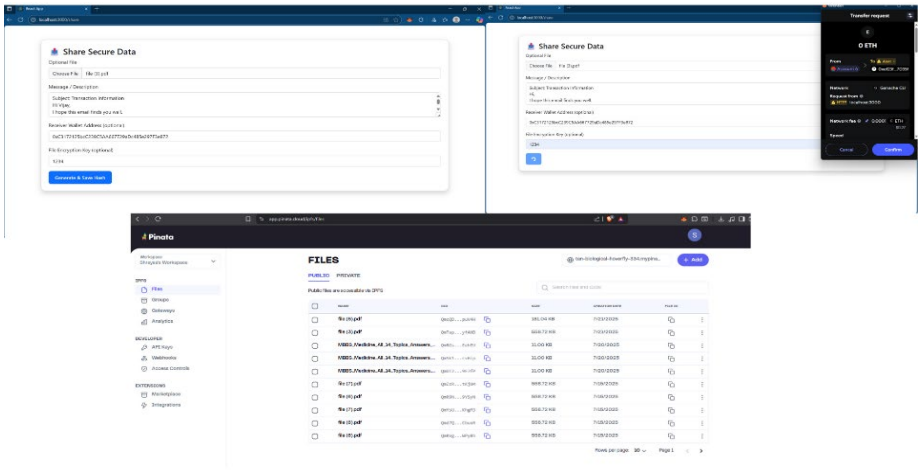

Figure 5:encrypted file upload and successful AES encryption

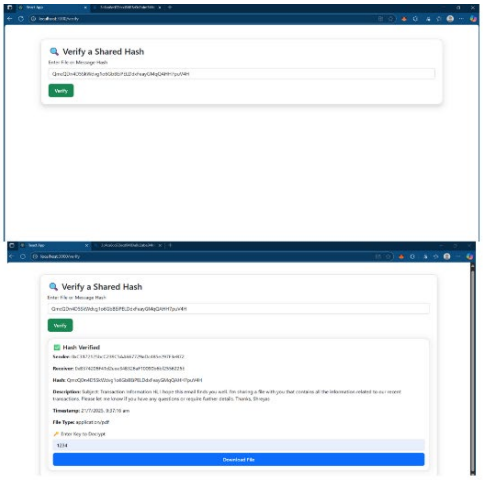

Figure 6:Hash verification results showing verified transaction

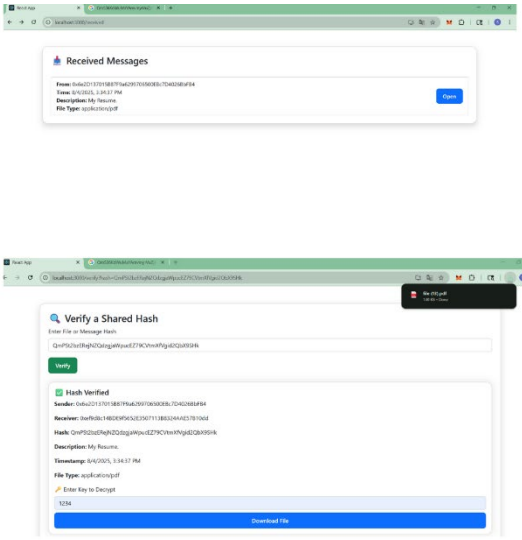

Figure 7:View Received Data & File Decryption

4.3 Comparative Benchmarking with Prior Literature

The comparison with research already done exposes the originality of this system. The proxy servers used by Huang, Chang and Yu (2022) compromised decentralization as opposed to our model, which erases middlemen with IPFS. Kumar et al. (2023) designed PhishBlock which is also constrained to the files of a cloud upload, we go further to support real-time file and message posted conditions. Khalifa et al. (2024) enhanced email integrity with blockchain headers with the necessity of user-side plugins, whereas our dApp provides an interface-level of transparency. Graph neural networks have been used both to identify blockchain phishing behaviours (Chen et al. 2024, Xiong et al. 2023 and Yan et al. 2024): their models are accurate but a computational burden and task specific. Our architecture is not ML based, so it is scalable and has extensive uses.

Lastly, the works on decentralized identity, including Gobika and Vaishnavi (2025), Sharma (2025), Salem et al. (2023), and Mazzocca et al. (2025) also emphasize the significance of DID and verifiable credentials in preventing phishing. These ideas are realised by us embedding DID verification directly into the dApp, so that the identity of senders is bound with Ethereum wallets. This turns theory into practice and provides genuineness in all transactions.

When considering our research question, the comparative benchmarking reveals that previous studies used to discuss pieces of the phishing issue, file integrity, email header, ML detection, or identity management, but our solution pulls them together to present a coherent platform. This merging shows that it is possible to use blockchain effectively to verify and authenticate communications and reduce poisoning attacks and social engineering, on a global level. To summarize, the findings indicate that the suggested dApp can operate effectively, provide strong scenario-oriented protection and trajectory in improvement to current procedures. The system explicitly answers the question offered by the research by identifying attempts of file tampering, preventing AI-generated unprofessional portrayals, and protecting encrypted confidentiality. Such observations indicate that blockchain can transcend theoretical potential to become a viable and implementable tool to counter phishing and social engineering messages by verifying and authenticating messages with blockchain benefit.

5 Conclusion

This project may be seen as an exhaustive solution to the problem of one of the most treacherous categories of cyber threats, which are phishing, social engineering, and AI-driven deception. We demonstrate that blockchain and cryptography are useful not only in a particularly sensitive area: financial transactions or data storage but also in restoring trust in digital communications, namely secure exchange of files and messages aided by a decentralized system. Content immutability, sender authenticity, and privacy controlled by users are at the heart of our system, which in their combination disrupt the hegemonies of centralized systems of security.

Technically connecting this vision entailed writing the smart contracts on Ethereum, IPFS through Pinata to store files distributedly, cryptographic integrity with the use of SHA-256 hashing and AES encryption when secure transport secrecy was required. Smart contracts are

universally readable registries which seal the hash of files and messages to a given Ethereum address and achieve spoof-resistant identity, as well as post-facto auditability. This takes file sharing out of the trust relationship and into the proof relationship by guaranteeing that what was received is exactly what the sender was willing try to send, and nobody has been able to modify in the process.

Moreover, through MetaMask and browsing frontend apps, it was possible to deal directly and securely with the blockchain Ethereum and continue to interact with it comfortably. Those design choices are guided by the necessity to balance between technical strength and usability, which is the key problem with implementing any decentralized system. In contrast to the previous proposals of academics which were either conceptual and/or overly technical, our system would provide a workable and accessible platform accessible to users either professionally, academically or governmentally.

It is also capable of providing a solution to a current issue that has been even more topical with the proliferation of generative AI to create effective phishing content. In contrast to reactive detection tools, or heuristic based classifiers, immutable proofs of content and origin are used by our approach, proved at the time of access. Such prospective architecture renders it resistant to impersonation perpetuated by AI and consistent with trends in cybersecurity contexts of achieving verifiable data integrity (Todupunuri, 2024; Rathod et al., 2024).

With this said, the system is however not limitless. AES encryption key exchange is done off-chain all the same exposing it to either a manual mistake or insecure back channel. The Ganache network to which the local personal computer is connected currently is acceptable when testing, but a public or Layer 2 chain will need to be used to test performance in a live environment.

Nevertheless, with these points of improvement, the project can be regarded as a huge advancement in blockchain-inspired cybersecurity. It validates the idea that the trust of the content should be made by cryptographic assurance, as opposed to wisdom or a central endorsement. It is a way ahead towards safe online communications in those cases where something deemed to be the truth is verifiable and that visual realisticness is not the building block of trust. The work therefore does not only give us a tool but also a template of the next phishing resistant communication system.

5.1 Discussion

The fact that a secure communications and file carrier platform built with blockchain technologies was a success proves that decentralized technologies show promise in countering an increasing number of cyber attacks that could be traced back to human tinkering, content scamming, and phishing made advanced by AI. Conventional methods of phishing protection usually involve using centrally located filters, black listing based systems and heuristic models that although are valuable enough are being caught up by the velocity and complexity of the latest attacks. Conversely, this project is redefining the problem in the context of verifiability and immutability-making sure that trust that the users who received

some content have to it, is cryptographically described and provable independent of any independent third parties.

The integrity model of this system is based on hash and the sender identity is verified through the use of blockchain anchored Ethereum addresses. The project is based on the modular design, where each message, or file, is delivered via multi-stage pipeline: hash is generated client-side, optionally the message is AES-ciphered, afterwards, the IPFS via Pinata decentralized storage is leveraged and, lastly, the hash is potentially anchored on Ethereum blockchain via a smart contract. This will guarantee that any form of modification, forgery or mixing of the data could be discovered not by means of probabilistic scores, but absolute and deterministic validation, a robust improvement and shift beyond traditional detection systems (Huang, Chang and Yu, 2022; Ogundokun et al., 2023).

One more thing to mention would be the seamless nature of the integration of cryptographic foundational block and decentralized infrastructure. Both integrity and confidentiality of the data are achieved as the system judiciously utilizes SHA-256 to hash and AES on the client as an encryption algorithm. Not only does the Ethereum layer of smart contract only have minimal metadata to make it tamper resistant and minimal on gas fees, but it also relates the submitted content to the wallet address of its sender and stands to replace traditional identifiers like names or email addresses with unchangeable blockchain identities (Khalifa et al., 2024). This can also provide non-repudiation this is a significant benefit in legal or audit case.

As far as real life applicability goes, this system is most applicable in those cases where the integrity and source of shared information is mission critical (e.g., law firms, research institutions, healthcare communication, or enterprise collaboration platforms). It also preconditions integration with decentralized identity (DID) systems and verifiable credentials (VCs) and, in the future, the possibility of introducing blockchain-native proof-of-origin workflows (Sharmin Akhter, Uddin Khan and Shikder, 2025; Vanna, 2025). Using DID extensions, people would be able to relate their blockchain identity to real-life profiles with a higher level of security without revealing confidential information.

It is worth noting that this platform also contains one of the most acute issues that can be used to describe the digital environment after 2020 the appearance of generative AI and deepfake attacks on their impersonation basis. Since according to Schmitt and Flechais (2024) and Romero-Moreno (2024), generative AI enables the creation of highly personalized, yet extremely large-scale scams, it can aid attackers both in performing them and, more importantly, getting away with them. In this case, the cryptographic authenticity which our system implements includes a scalable protection, that is, no matter how convincing an AI- generated file, video or message might appear, it will not be able to forge the initial on-chain hash and sender ID.

The greater meaning of this system, in short, is that it may move the discussion of trust in a whole new direction, away from an opinion-based model on trust, and algorithmic selection, and towards a cryptographic proof and wealth of analytics on an extendable digital provenance. A paradigm of this kind is particularly relevant as Web3 gains popularity and the extent to which human content has been replaced by machine-generated content blurs. The system will not just provide a resolution to phishing attack; it will provide a structure on how to ensure integrity of content, propagation of trust, and decentralized communications within the contemporary internet environment.

References:

Fakieh, A. and Akremi, A. (2022). An Effective Blockchain-Based Defense Model for Organizations against Vishing Attacks. *Applied Sciences*, 12(24), p.13020. doi:https://doi.org/10.3390/app122413020.

Gate.com (2024). *What is Social Engineering in the Crypto Space*. [online] Gate.com. Available at: https://www.gate.com/learn/articles/what-is-social-engineering-in-the-crypto-space/2463?utm_source.

Huang, H.-S., Chang, T.-S. and Yu, J.-Y. (2022). *A Secure File Sharing System Based on IPFS and Blockchain*. [online] Arxiv. Available at: https://arxiv.org/abs/2205.01728 [Accessed 21 Jul. 2025].

Intelligence, M.T. (2022). *'Ice phishing' on the blockchain*. [online] Microsoft Security Blog. Available at: https://www.microsoft.com/en-us/security/blog/2022/02/16/ice-phishing-on-the-blockchain/.

Ivanov, N., Lou, J., Chen, T., Li, J. and Yan, Q. (2021). Targeting the Weakest Link: Social Engineering Attacks in Ethereum Smart Contracts. *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. doi:https://doi.org/10.1145/3433210.3453085.

Ivanov, N. and Yan, Q. (2022). *Et tu, Blockchain? Outsmarting Smart Contracts via Social Engineering*. [online] arXiv.org. Available at: https://arxiv.org/abs/2209.08356 [Accessed 21 Jul. 2025].

Khalifa, O., Binti Tengku Nor Effendy, T.H.S., Zaharadeen Ahmed, M., El-Khazmi, E. and Nasser Esgiar, A. (2024). Blockchain Based Email Security to Mitigate Phishing Attack. *Asian Journal of Electrical and Electronic Engineering*, 4(2), pp.77–86. doi:https://doi.org/10.69955/ajoeee.2024.v4i2.73.

Kota, R.K., Hasthalamangali , S., Movva, D., Borugadda, S. and Shaik, S. (2025). Block-Chain Based Document Verification System Using IPFS . In: *International Journal of Creative Research Thoughts*.

Kumar, N., Goel, V., Ranjan, R., Majid Altuwairiqi, Hashem Alyami and Simon Atuah Asakipaam (2023). A Blockchain-Oriented Framework for Cloud-Assisted System to Countermeasure Phishing for Establishing Secure Smart City. *Security and Communication Networks*, 2023, pp.1–13. doi:https://doi.org/10.1155/2023/8168075.

Ogundokun, R.O., Arowolo, M.O., Damaševičius, R. and Misra, S. (2023). Phishing Detection in Blockchain Transaction Networks Using Ensemble Learning. *Telecom*, [online] 4(2), pp.279–297. doi:https://doi.org/10.3390/telecom4020017.

Pitre, V., Joshi, A. and Das, S.K. (2023). Blockchain and Machine Learning Based Approach to Prevent Phishing Attacks. *3rd Asian Conference on Innovation in Technology (ASIANCON)*. doi:https://doi.org/10.1109/asiancon58793.2023.10270720.

Rathod, T., Jadav, N.K., Tanwar, S., Alabdulatif, A., Garg, D. and Singh, A. (2024). A comprehensive survey on social engineering attacks, countermeasures, case study, and

research challenges. *Information Processing & Management*, [online] 62(1), p.103928. doi:https://doi.org/10.1016/j.ipm.2024.103928.

Romero-Moreno, F. (2024). Deepfake Fraud Detection: Safeguarding Trust in Generative Ai. *Deepfake Fraud Detection: Safeguarding Trust in Generative Ai*. [online] doi:https://doi.org/10.2139/ssrn.5031627.

Schmitt, M. and Flechais, I. (2024). Digital deception: generative artificial intelligence in social engineering and phishing. *Artificial Intelligence Review*, 57(12). doi:https://doi.org/10.1007/s10462-024-10973-2.

Sharmin Akhter, J., Uddin Khan, Md.I. and Shikder, A. (2025). BLOCKCHAIN-POWERED PROTECTION AGAINST SOCIAL ENGINEERING IN E-COMMERCE . *World Journal of Engineering Research and Technolology*, 11(5).

Tech, F. (2023). *A Tentative Study in Social Engineering Attacks in Blockchain Ecosystem*. [online] Medium. Available at: https://medium.com/coinmonks/a-tentative-study-in-social-engineering-attacks-in-blockchain-ecosystem-cd1c4c0a5638.

Todupunuri, A. (2024). Exploring the Use of Generative AI in Creating Deepfake Content and the risks it poses to Data Integrity, Digital Identities, and Security Systems. *European Journal of Advances in Engineering and Technology*, 11(10), pp.70–74.

Vanna, karthik (2025). *The Role of Blockchain in Preventing Phishing Scams*. [online] Insights2Techinfo. Available at: https://insights2techinfo.com/the-role-of-blockchain-in-preventing-phishing-scams/ [Accessed 21 Jul. 2025].

Vasile Crudu (2025). *Enhancing Security for Your Blockchain Application to Safeguard Against Phishing and Social Engineering Threats*. [online] Moldstud.com. Available at: https://moldstud.com/articles/p-enhancing-security-for-your-blockchain-application-to-safeguard-against-phishing-and-social-engineering-threats.

Chen, Z., Liu, S.-Z., Huang, J., Yu-Han Xiu, Zhang, H. and Long, H.-X. (2024). Ethereum Phishing Scam Detection Based on Data Augmentation Method and Hybrid Graph Neural Network Model. Sensors, 24(12), pp.4022–4022. doi:https://doi.org/10.3390/s24124022.

Gobika S. and Vaishnavi N. (2025). Blockchain Based Identity Management System. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11(2), pp.1413–1420. doi:https://doi.org/10.32628/cseit25112471.

Gomaa Salem, S.H., Hassan, A.Y., Moustafa, M. and Hassan, M.N. (2023). Blockchain-based biometric identity management. Springer. doi:https://doi.org/10.1007/s10586-023-04180-x(.

Mazzocca, C., Acar, A., Selcuk Uluagac, Montanari, R., Paolo Bellavista and Conti, M. (2025). A Survey on Decentralized Identifiers and Verifiable Credentials. IEEE CommunicationsSurveys&Tutorials,[online]pp.1–1. doi:https://doi.org/10.1109/comst.2025.3543197.

Sharma, V. (2025). Blockchain-Based Identity Management Systems For Financial Institutions. International Journal of Research and Analytical Reviews, 12(3). doi:https://doi.org/10.56975/ijrar.v12i3.317858.

Trad, F., Semaan-Nasr, E. and Chehab, A. (2024). MLPhishChain: a machine learning-based blockchain framework for reducing phishing threats. Frontiers in Blockchain, 7. doi:https://doi.org/10.3389/fbloc.2024.1484894.

Xiong, A., Tong, Y., Jiang, C., Guo, S., Shao, S., Huang, J., Wang, W. and Qi, B. (2023). Ethereum phishing detection based on graph neural networks. IET Blockchain, 4(3), pp.226–234. doi:https://doi.org/10.1049/blc2.12031.

Yan, C., Han, X., Zhu, Y., Du, D., Lu, Z. and Liu, Y. (2024). Phishing behavior detection on different blockchains via adversarial domain adaptation. Cybersecurity, 7(1). doi:https://doi.org/10.1186/s42400-024-00237-5.