RESEARCH ARTICLE

Leveraging Polygon Blockchain and Zero-Knowledge Proofs for Secure Electronic Evidence Storage in Court: A Novel Framework

Sarbeswar Behera[1] **and Suraj Arya**[1,*]

[1] Department of Computer Science & Information Technology, Central University of Haryana, Haryana 123031, India

Abstract

The increasing digitization of modern society has led to a growing volume of digital information—such as emails, photos, videos, and electronic records—being presented as evidence in legal proceedings. However, traditional storage systems are costly, slow, and prone to server failures. To address these challenges, we propose a system leveraging blockchain and IPFS to store electronic evidence securely and reliably. Existing Layer 1 blockchain-based systems face three key limitations: (1) scalability issues with slow throughput and high costs under heavy traffic; (2) privacy risks due to public visibility of transaction data; and (3) impracticality of on-chain storage for large evidentiary files. Our system utilizes Polygon, a faster and more cost-effective Layer 2 blockchain, combined with IPFS for off-chain storage of documents, images, and videos. Each file is assigned a cryptographic hash stored on-chain to ensure integrity and tamper resistance. Smart contracts enforce role-based access control, restricting actions to authorized participants such as police officers, lawyers, and judges. Experimental evaluation with 10,000 transactions achieved approximately 14.29 TPS on modest hardware, with the potential for significantly higher throughput in optimized environments. This demonstrates the system's viability for both high-volume urban courts and resource-constrained rural settings. Overall, the proposed solution offers enhanced security, controlled access, reduced storage costs, and improved efficiency for digital evidence management in judicial systems.

Keywords: blockchain, electronic evidence management, smart contracts, IPFS, polygon CDK, zkEVM, access control, decentralized storage, transaction per second.

1 Introduction

The digital revolution refers to the shift from analog and mechanical technologies to digital systems, fundamentally reshaping industries, communication, and daily life. Continuous innovation and responsible digital transformation will shape the future, making technology an even more integral part of human life.

In the field of evidence, the electronically stored

Submitted: 05 April 2026
Revised: 24 June 2026
Accepted: 08 July 2026
Published: 27 August 2026

Vol. 2, **No.** 3, 2026.
10.62762/JSE.2026.766488

***Corresponding author:**
Suraj Arya
surajarya@cuh.ac.in

Citation
Behera, S., & Arya, S. (2026). Leveraging Polygon Blockchain and Zero-Knowledge Proofs for Secure Electronic Evidence Storage in Court: A Novel Framework. *ICCK Journal of Software Engineering*, 2(3), 197–219.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

information is admitted as evidence at a trial or hearing, including electronic communications, such as emails, text messages, voice messages, and chat room communications; digital photographs; website content, including blog posters and social media postings; cookies; and computer stored records, such as electronic trading records, essential videos, communication records. Unfortunately, not all contents are properly protected. Evidence plays an important role across various fields, serving different purposes depending on the context and circumstances. In digital forensics and cybersecurity, investigators rely on data logs, malware traces, and digital footprints to solve cybercrime cases or unauthorized access. Klasén et al. [1] points out that electronic evidence plays a vital role in analyzing, solving cybercrimes by systematically collecting, preserving, and presenting electronic evidence.

In the criminal justice system, evidence is crucial for determining the truth in legal cases. It can include physical objects like weapons or fingerprints, digital records such as emails or surveillance footage, forensic findings like DNA, and witness testimonies. All these forms of evidence help law enforcement and courts decide whether a person is guilty or innocent, ensuring fair and just outcomes. Miller [2] points out how digital evidence is utilized in criminal cases and its impact on investigations and prosecutions. We can conclude in the above scenario electronic evidence is being used as a powerful tool in the ongoing digital evolution. However, storing the electronic evidence in a cost-effective manner while ensuring security and privacy presents significant challenges. In the early 2000s, law enforcement agencies started dealing with huge amounts of electronic evidence (like hard drives, phones, etc.) as the number of cases increased over time. Storing all this data safely and efficiently was hard. Davis et al. [3] proposed a Storage Area Network (SANs) - a kind of centralized storage system. Though these systems made it faster and easier to store, manage, and access digital evidence, these systems are expensive to deploy and maintain. If the network becomes unavailable, access can be delayed. Kumar et al. [4] proposed an Internet-of-Forensic (IoF) framework that leverages blockchain technology to address digital forensics challenges in IoT environments, providing tamper-proof evidence collection, secure storage, and chain-of-custody tracking through decentralized mechanisms. Centralized evidence storage systems provide tamper-resistant mechanisms in a centralized system via secure software, secure hardware, physical separation or hybrid strategies. These systems face the following challenges: (1) evidence tampering, in which evidence may be maliciously modified, removed or untraceable, (2) privacy leaks, which means private information like evidence content, evidence providers may be leaked. (3) the single point of failure, in which if it fails, causes the entire system to stop working or become invalid. (4) scalability issue, which arises if the amount of evidence is too large to store.

Therefore, changing the storage methods, enhancing credibility, and enabling the secure sharing of electronic evidence are urgent priorities. The emergence of blockchain technology offers a promising solution to the challenges associated with electronic evidence storage. Blockchain is an immutable, distributed ledger that facilitates the records of transaction in a decentralized network. When applied to the storage of electronic evidence, blockchain leverages its distributed nature to eliminate single points of failure. The use of public-private key cryptography and hash encryption ensures data integrity and prevents privacy leaks. The immutable nature of blockchain helps prevent evidence tampering. Despite these advantages, existing blockchain-based evidence management systems still face several practical limitations. The use of InterPlanetary file system (IPFS) can prevent single points of failure while accessing the electronic evidence, while it reduces cost of storage by enabling decentralized and efficient data distribution. Bonomi et al. [5] proposed a private blockchain based chain of custody evidence management system which eliminates single point of failure while ensuring data privacy and data integrity. Tian et al. [6] points out how hash encryption and multi-signature techniques help prevent evidence tampering.

Although blockchain Technology is an advanced solution of securing electronic evidence and utilized as a powerful tool when needed, it still faces several challenges:

- Existing systems are often built on public or private blockchains, which are typically Layer 1 networks. These networks can become slow as the number of transactions increases and may struggle to handle high traffic due to expensive transaction fees.
- Since many of these systems operate on public blockchains like Ethereum, all transaction data is visible to everyone, which poses a significant

privacy risk, especially when dealing with sensitive or confidential information.

- With the number of criminal cases continuously increasing, storing electronic evidence directly on the blockchain can become impractical and costly, as on-chain storage is limited and often expensive.

In response to these limitations, we propose a blockchain-based framework for the secure storage and management of electronic evidence collected during criminal investigations. This framework is built on the Polygon blockchain network—a Layer 2 scaling solution [7]—and utilizes an off-chain storage technique through IPFS (Inter Planetary File System) [8]. The primary objective is not only to secure electronic evidence but also to establish a secure and transparent system for court procedures. The framework incorporates smart contracts to enable secure authentication for case registration, evidence submission, and data verification, data sharing, permission management. Electronic evidence—such as images, videos, and forensic data—is collected and uploaded to IPFS (Inter Planetary File System) for decentralized storage. The associated Content Identifiers (CIDs) are then linked to smart contracts on the blockchain. The main contributions of this paper are summarized below.

- **Collection, Storage, and Abstraction of Hierarchical Authority:**
  Developed a framework for the systematic collection, secure storage, and hierarchical abstraction of electronic evidence and authority-related data.

- **Secure Evidence Storage Using IPFS:**
  Implemented a decentralized approach using the InterPlanetary File System (IPFS) to securely store digital evidence using Content Identifiers (CIDs).

- **Smart Contract Design for Role-Based Interaction:**
  Designed a secure and modular smart contract system that enables controlled abstraction and interaction among multiple stakeholders, such as plaintiffs, police officers, lawyers, judicial authorities, and forensic teams.

- **Blockchain Deployment Setup:**
  Deployed the proposed blockchain setup on a Polygon CDK (zkEVM) network on a local system using Docker-based containerized services for efficient testing and scalability.

The proposed framework addresses key challenges related to scalability, privacy, and secure evidence management. By leveraging Polygon's Layer 2 architecture, multiple transactions are aggregated into batches before being submitted to the Ethereum Layer 1 blockchain, thereby improving throughput and reducing transaction costs. Furthermore, zero-knowledge proofs (zk-proofs) enable transaction verification without revealing sensitive information, ensuring privacy while maintaining security. Electronic evidence is stored off-chain using IPFS, providing decentralized and tamper-resistant storage. Together, these features enhance transparency, integrity, and trust in judicial proceedings.

The rest of this paper is organized as follows: Section 2 provides a review of related work and discusses how blockchain is being used in court processes. Section 3 describes the research preliminaries used in this study. Section 4 explains the research methodology and the design of the proposed framework. Section 5 presents the experiments conducted and the results obtained. Section 6 provides a comparative discussion. Section 7 describes the limitations of the proposed framework. Finally, Section 8 concludes the paper.

2 Literature Review

Atlam et al. [9] presents a systematic literature review of blockchain forensics, comprehensively examining state-of-the-art techniques, applications, challenges, and future directions in the field. The review analyzes how blockchain's decentralization enhances security in forensic investigations while simultaneously complicating the identification and tracking of illegal activities, particularly in linking blockchain addresses to real-world identities. It highlights that while blockchain immutability protects evidence against tampering, it also introduces forensic challenges by preventing the modification or deletion of erroneous records. The review further emphasizes the need for privacy-preserving blockchain architectures that maintain evidentiary reliability. A key limitation identified is that most reviewed frameworks lack standardized interoperability mechanisms and legal recognition frameworks, which are critical for admissibility of blockchain-based evidence in court. Bonomi et al. [5] uses the IBFT (Istanbul Byzantine Fault Tolerance) consensus mechanism. This paper presented B-CoC, a blockchain-based architecture to dematerialize the CoC process in digital forensics. This paper uses the IBFT consensus mechanism instead of PoW, which consumes far less energy.

By using Private blockchain Framework Geth, this system eliminates the single point of failure. While this system adopts well under controlled situations, it may face challenges when handling a large number of evidence transactions under different jurisdictions. This system uses a Centralized Evidence database which will be too cost ineffective over time and it leads to a phase of destruction of all the evidence due to a single point of failure. So far, this system proposes fixed no of validators which is a key point of limitation. Tian et al. [6] adopts PBFT and Optimized PBFT. This paper presented a secure digital evidence framework, Block-DEF using blockchain technology that focuses on the security of digital evidence for file tampering. This research paper uses loose coupling structure which shows that evidence and the information are stored separately. Again, this paper adopts the multi-signature technique to support privacy. The limitation of this research paper is scalability issues. As the number of blocks increases overtime which requires all the nodes to maintain a full copy of the blockchain, leading to significant storage demands. It may lead to blockchain bloat, which occurs when large numbers of data are stored on the blockchain. Additionally, it assumes a single owner of evidence, in which ownerships can be shared. Brotsis et al. [10] uses Hyperledger Fabric. This Research paper proposes a Blockchain based cyber-trust platform which is based on Hyperledger fabric private blockchain. It relies on advanced intrusion detection tools to find out the malicious activity and enhances the IoT environments. We have identified some of the limitations of this paper. This paper proposes a Layer1 blockchain platform which may fail to handle in future due increase in transaction of Evidences. This paper highlights an evidence database which is centralized in nature that may cause destruction of all the evidence due to a single point of failure. Chen et al. [11] is based on a Consortium based Blockchain, Practical Byzantine Fault tolerance algorithm. This Research paper proposed a consortium based blockchain framework that offers a secure and efficient way to store, verify and examine the electronic evidence. This system uses elliptic curve cryptography for data encryption. This paper has certain limitations. As this paper proposed a consortium based blockchain built on Layer1 which can't handle the large amount of on chain transactions. Verma et al. [12] uses a Public Blockchain, POW consensus mechanism, IPFS. This paper proposes a tamper-proof secure evidence management system which uses IPFS (Inter planetary File system) as the off-chain data storage system for storage of the electronic evidence. This Research paper proposes to build it on public blockchain which is transparent in nature which may leak sensitive data. As it is working on a layer1 solution, it may not be able to handle the high volume of transaction load. It requires high costs to store transactions on blockchain which is too challenging in nature. Sun et al. [13] provides a comprehensive survey on zero-knowledge proof (ZKP) in blockchain, systematically reviewing the principles, constructions, and applications of ZKPs across various blockchain scenarios. The paper covers key ZKP schemes such as zk-SNARKs and zk-STARKs, and evaluates their applicability to privacy-preserving transactions, identity verification, and decentralized finance. This survey serves as a foundational reference for understanding how ZKPs can be integrated into blockchain systems to enable verification without disclosure of sensitive information. A key limitation identified is that most surveyed ZKP schemes involve significant computational overhead, and their efficient integration into high-throughput blockchain environments—such as those required for large-scale judicial evidence management—remains an open challenge.

Wang et al. [14] explores the evidentiary significance of blockchain records and the procedural implications of integrating blockchain technology into judicial systems. This paper conducts legislative analysis, comparative case law analysis, and technical examination of blockchain mechanics, and proposes a specialized consensus mechanism for standardizing blockchain evidence authentication. It argues that blockchain evidence may be classified as hearsay exceptions or non-hearsay depending on record characteristics, and outlines strategies to enhance trustworthiness of blockchain-based proof. The primary limitation of this work is that it remains largely theoretical and legal-oriented, without providing a concrete technical implementation for evidence storage or access control. Furthermore, it does not address scalability concerns or privacy-preserving mechanisms such as zero-knowledge proofs.

Brotsis et al. [15] utilizes Practical Byzantine Fault Tolerance consensus mechanism, Hyperledger fabric. This paper proposes an IoT based ecosystem that focuses on secure, tamper-proof digital evidence storage ecosystem. This paper focuses on implementing IPFS, MEC (multi-access edge computing that prevents blockchain bloat and efficient forensic data processing. This framework works on

Hyperledger fabric private blockchain which is a layer1 solution. As transaction loads increase, it may fail in efficient forensic data storage management.

Lewulis [16] deals with Social Media Evidence. This paper aims to supplement the discourse with information on how social media evidence is perceived by Polish civil litigation and criminal defense lawyers, based on a cross-sectional online survey (n = 278). Social media content is regarded as potentially useful in court because it provides important, legally relevant information. This paper introduces how social media evidence is useful in a wide range of cases including criminal and civil matters. The main challenge is establishing standardized methods for collecting and preserving such vast amounts of social media (SM) evidence which is increasing rapidly day by day. This underscores the broader need for tamper-proof and verifiable digital evidence management systems—such as the blockchain-based framework proposed in this work—capable of authenticating and preserving diverse forms of electronic evidence for judicial proceedings.

Malik & Sharma [17] employs Ethereum based public blockchain, POS, IPFS. This paper proposes a transparent chain of custody of digital evidence. This paper uses IPFS to store the digital evidence securely while keeping only a hash of the evidence on the blockchain. IPFS reduces storage cost and enhances security by preventing a single point of failure. This Research paper uses Ethereum based blockchain which is too costly to store the hash of the evidence. As Ethereum is public in nature so its transparent nature can expose the sensitive digital evidence.

Ratul et al. [18] proposes a blockchain-based solution for managing digital evidence in cybercrime cases within the judicial domain. The framework leverages a consortium blockchain to enable secure collaboration among judicial stakeholders—including investigators, prosecutors, and administrators—while ensuring data integrity and admissibility of evidence in court. It implements role-based access control with distinct permissions for each user group, and uses off-chain storage to manage evidence data, with only hashes recorded on-chain. The study also demonstrates the system's feasibility through implementation using smart contracts and a web interface. A notable limitation is that the system is built on a Layer 1 consortium blockchain and does not incorporate zero-knowledge proof mechanisms, leaving privacy-preserving verification of state transitions unaddressed. Scalability under high transaction loads in large-scale judicial environments also requires further evaluation.

Rani et al. [19] proposes a secure digital evidence preservation system for IoT-enabled smart environments, integrating IPFS, blockchain, and smart contracts to ensure the integrity, confidentiality, trust, and privacy of digital evidence throughout forensic investigations. The system employs role-based access control to authenticate users before any evidence insertion or deletion, and stores evidence files on IPFS while recording only cryptographic hashes on-chain, effectively preventing single points of failure and reducing storage costs. This framework closely aligns with the core technical approach of the present work. However, its primary limitation lies in its focus on general IoT environments rather than the judicial domain specifically, and it does not incorporate zero-knowledge proof mechanisms for privacy-preserving verification of state transitions.

Liu & Zheng [20] uses FISCO BCOS, Practical Byzantine Fault Tolerance (PBFT), IPFS. This paper proposes a blockchain based judicial evidence preservation system that significantly focuses on off-chain storage. This paper proposes it to build on the private blockchain in which storing the transaction is very costly. As the number of cases increases day by day, to store every transaction in a block is taking too much of the cost.

Tortola et al. [21] is based on Data Integrity proof, Validity proof, Fraud proof. This paper gives a systematic review on the proving scheme used in layer2 solution which is based on zero knowledge proofs. This paper emphasizes Data integrity proofs, validity proofs, fraud proofs are the building blocks of the technique while connecting layer2 to layer1. Despite its advantage, this paper lacks several points like (1) Data integrity proofs can't be verified without the original data. (2) Validity proofs are difficult to integrate due to zero knowledge proof complexity. (3) In the case of Fraud proof, it is very slow because people need to wait for fraud proofs and also participants need to stay online to detect fraud.

3 Research Preliminaries

3.1 Blockchain Technology

Since the introduction of Bitcoin in 2009, blockchain has gained recognition as a secure and decentralized way to keep track of transactions. At its core, a

Table 1. Comparison of consensus mechanisms and layer 2 scaling solutions.

Sl. No.	Mechanism	Throughput (TPS)	Finality Time	Security Model
1	PoW (Bitcoin)	Low (∼7–30)	Slow (10–60 min)	Economic + computational (mining power)
2	PoS (Ethereum)	Medium–High (30–1000+)	Fast (12 s – a few min)	Economic (staked tokens)
3	DPoS (EOS, TRON)	High (1000+)	Very fast (seconds)	Delegated validators elected by token holders
4	zkRollups (Layer 2)	Very high (thousands)	Fast to instant	Inherits Layer 1 security + zero-knowledge proofs

blockchain is a distributed ledger or database that is resistant to tampering. It allows for the recording and tracking of assets and transactions without the need for a central authority. Transactions are grouped together into blocks, each of which is secured using cryptographic methods. These blocks have a limited capacity for data, and as more transactions occur, new blocks are added in sequence, creating a continuous chain of records.

3.2 Consensus Mechanism

A consensus mechanism is a fundamental process in decentralized networks, particularly blockchains, ensuring that all participants agree on a single, consistent version of data or transaction history. In the absence of a central authority, consensus protocols establish trust and maintain uniformity across distributed nodes. Common consensus mechanisms include Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT), among others. These mechanisms are designed to validate transactions, prevent fraudulent activities, and enforce network rules. The choice of consensus protocol significantly influences the network's performance in terms of security, transaction speed, energy efficiency, and decentralization.

Layer 2 scaling solutions [7] are built atop Layer 1 blockchains to alleviate inherent scalability limitations. These solutions reduce the computational and storage burden on the main chain by handling transactions off-chain through techniques such as sidechains, rollups, validium, and off-chain computation. They enhance transaction throughput and reduce fees while preserving a high level of security. A comparative summary of selected consensus mechanisms is presented in Table 1.

Latency refers to the time delay between submitting a transaction and its final confirmation on the blockchain, reflecting system responsiveness.

Throughput, typically measured in transactions per second (TPS), denotes the rate at which a network processes transactions. It is a critical scalability metric that directly affects network congestion and transaction fees.

Security is a paramount requirement in digital evidence management systems, where sensitive data must be safeguarded against unauthorized access and tampering.

Encryption is a widely adopted cryptographic technique that transforms readable plaintext into an unreadable ciphertext, ensuring that only authorized entities can decrypt and access the original information.

Confidentiality ensures that sensitive information is accessible exclusively to authorized individuals, often enforced via role-based access control (RBAC). In blockchain-based systems, confidentiality is typically realized through cryptographic primitives, secure authentication protocols, and fine-grained permission mechanisms.

3.3 Polygon Blockchain

Polygon blockchain network [7] is a layer 2 blockchain network , which is a faster and cheaper version of Ethereum. Polygon was created in 2017 by Jaynti Kanani, Sandeep Nailwal, and Anurag Arjun to improve user experience and solve scalability issues like high gas fees and slow transaction speeds. It employs modern techniques such as Polygon PoS, zk-Rollups, and Optimistic Rollups. A comparison with other blockchain consensus mechanisms is presented in Table 1.

Polygon network can handle a large number of transactions at once, has very low fees, and is easy to use for developers already familiar with Ethereum. Polygon's fast execution technique helps prevent blockchain scalability issues. It uses a Proof-of-Stake system, which consumes less energy and is more environmentally friendly than older methods like Proof-of-Work. Major companies such as Adobe,

Stripe, and Starbucks are also exploring Polygon for things like digital identities, rewards programs, and blockchain-based payments. In the future, blockchain technology will be important for keeping digital systems safe and secure.

3.4 Cryptography

Cryptography is a fundamental technique for ensuring the security and privacy of information. It encompasses various primitives, including hashing, encryption, and digital signatures, to protect data from unauthorized access and tampering.

Zero-Knowledge Proof (ZKP) [13] is a cryptographic method in which one party, the prover, can convince another party, the verifier, that a given statement is true, without revealing any additional information beyond the validity of the statement itself. Recent advances in ZKP efficiency, such as the aggregation scheme proposed by Kuznetsov et al. [23], have significantly reduced proof size and computational overhead, making ZKPs more practical for large-scale blockchain applications. ZKPs have been widely applied in diverse domains, such as privacy-preserving cryptocurrency transactions, authentication systems, supply chain transparency, secure voting systems, decentralized identity verification, cross-border payments, and decentralized finance protocols.

A rollup is a Layer 2 scaling solution designed to improve the scalability and efficiency of blockchain networks. It operates by aggregating multiple off-chain transactions into a single batch and subsequently submitting this batch as one compressed transaction to the underlying Layer 1 blockchain. Among various rollup types, the Zero-Knowledge Rollup (zkRollup) [7] is a specific variant that employs Zero-Knowledge Proofs to certify the correctness of all transactions within the batch. Modular zk-rollup architectures, such as the one proposed by Lavaur et al. [24], enable multiple rollups to co-exist on the same smart contracts, significantly reducing deployment costs while maintaining the security guarantees of the underlying Layer 1 blockchain. The operational workflow of a zkRollup can be described as follows. First, the rollup aggregates numerous transactions into a batch. Second, a validity proof in the form of a zero-knowledge proof is generated, attesting to the correctness of every transaction in that batch. Third, both the batch data and the accompanying validity proof are submitted to the Layer 1 blockchain. Finally, the Layer 1 network verifies the submitted proof; if valid, the entire batch is accepted, thereby inheriting the security of the main chain while achieving significantly higher throughput.

A **digital signature** is a cryptographic mechanism used to verify the authenticity and integrity of digital data. It is generated using the sender's private key and can be validated by any recipient using the corresponding public key, thereby ensuring non-repudiation and data integrity.

3.5 Smart Contract

A smart contract is a self-executing program deployed on the blockchain that automatically executes the terms of an agreement when predetermined conditions are met [25]. These smart contracts can handle a wide range of functions, from simple payments between users to complex operations like managing decentralized applications (DApps), financial services (DeFi), supply chain logistics, and non-fungible tokens (NFTs). Smart contracts are supported by various blockchain platforms, including Ethereum and Polygon.

3.6 Block Scalability

Blockchain scalability [22] explains the ability of a system to handle the execution of a large number of transactions efficiently.

In some of the papers we reviewed, we found that Bitcoin, the foundational blockchain network operating on the Proof of Work (PoW) consensus mechanism, has relatively low transaction throughput, approximately 7–30 transactions per second (TPS). In contrast, the Proof of Stake (PoS) consensus mechanism offers higher throughput, around 30–1000+ TPS, while Delegated Proof of Stake (DPoS) achieves even higher throughput, exceeding 1000 TPS.

Sanka & Cheung [22] discuss several on-chain and off-chain scaling solutions. While on-chain scaling solutions directly address block capacity issues, they also increase the load on blockchain network nodes. Off-chain scaling solutions, such as ZK-Rollups, have emerged as critical techniques for improving scalability without overburdening the network.

Among all approaches, Layer 2 scaling solutions provide the most efficient method by conducting computations off-chain and submitting summarized results to the Layer 1 blockchain. This approach significantly enhances both efficiency and scalability, as illustrated in the transaction block diagram of Figure 1.

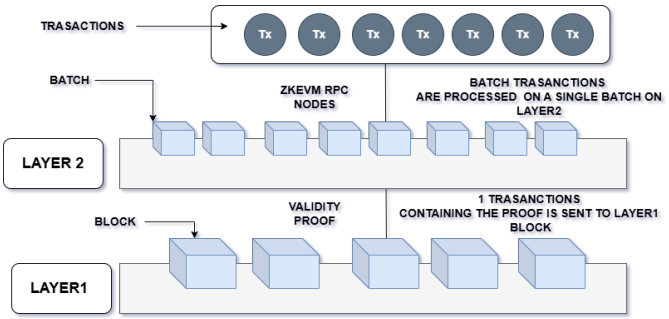


Figure 1. Transaction block diagram.

3.7 Interplanetary File System

InterPlanetary file system [8] is a decentralized file system that operates on the principle of content-based addressing. Most organizations store their data in the cloud. However, cloud technology relies on centralized databases managed by third-party providers, which can increase the risk of data loss or tampering. IPFS addresses the limitations of traditional centralized storage systems, where data can be lost, modified, or compromised by unauthorized access. Since files are distributed across multiple nodes in the network, IPFS reduces the risks commonly associated with centralized cloud servers.

IPFS utilizes technologies such as peer-to-peer (P2P) networks, cryptographic hashing, Distributed Hash Tables (DHTs), BitSwap protocols, Git-inspired versioning mechanisms, and the Self-Certifying File System (SFS) to provide a decentralized file storage solution. Unlike HTTP, which locates files based on their network location, IPFS retrieves files based on their content. This unique identifier is known as a Content Identifier (CID). In IPFS, all participating nodes are treated equally. To locate a file, users search for its unique hash value rather than accessing a specific server address.

When a file is uploaded to IPFS and exceeds 256 KB in size, it is divided into smaller chunks. IPFS then constructs a Merkle Directed Acyclic Graph (Merkle DAG) and creates an IPFS object that links all file chunks together, enabling efficient and secure content retrieval. The overall IPFS storage mechanism is illustrated in Figure 2.

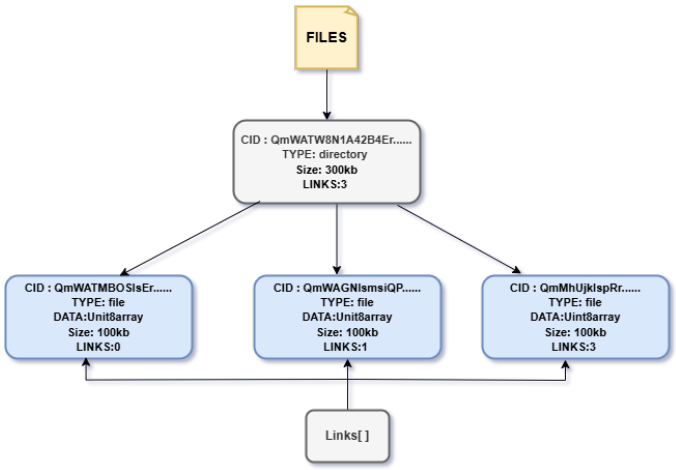


Figure 2. IPFS storage system.

4 Research Methodology and Framework Design

4.1 Research Approach

This research adopts an experimental and framework-oriented methodology for the design, implementation, and evaluation of a secure blockchain-based Electronic Evidence Storage System using Layer 2 blockchain network technology. The proposed framework addresses challenges in traditional judicial evidence management systems, including centralized storage dependency, evidence tampering risks, unauthorized access, and limited transparency.

To overcome these limitations, the study integrates Polygon CDK, smart contracts, cryptographic mechanisms, and InterPlanetary File System (IPFS) to establish a secure, transparent, and scalable judicial evidence storage environment.

The research methodology follows a systematic process consisting of requirement identification, framework design, blockchain deployment, smart contract implementation, IPFS integration. Initially, functional and non-functional requirements were identified based on the limitations of traditional judicial systems and the operational needs of secure evidence preservation. Subsequently, a blockchain-based framework was designed and implemented to support judicial operations such as authority registration, FIR submission, case registration, evidence verification, and secure information sharing.

Furthermore, the proposed framework was evaluated using performance metrics such as Transactions Per Second (TPS), latency, reliability, and scalability to assess system efficiency, responsiveness, and operational stability. Security and privacy considerations, including encryption, confidentiality, and controlled access, were incorporated throughout the framework design. The detailed framework architecture, implementation methodology, and evaluation process are discussed in the subsequent

sections.

4.2 Framework Requirements

Based on the challenges associated with traditional judicial evidence management systems, a set of functional and non-functional requirements was identified to guide the design and evaluation of the proposed framework. These requirements were derived to ensure secure evidence preservation, efficient judicial operations, and scalable blockchain performance. The theoretical background of the associated technologies and metrics is discussed in Research Preliminaries section.

4.2.1 Functional Requirements

The proposed framework must satisfy a set of core functional requirements to ensure its effectiveness in judicial evidence management. First, security is paramount to guarantee the safe preservation of judicial evidence and to protect against unauthorized modifications or tampering. Second, the system shall implement role-based access control to provide differentiated and controlled access for various judicial participants, including magistrates, police officers, lawyers, litigants, and forensic experts, ensuring that each actor can only perform actions permitted by their designated role. Third, the framework shall support decentralized evidence storage through integration with the InterPlanetary File System (IPFS), enabling secure, tamper-resistant, and resilient storage of digital evidence without relying on a single point of failure. Fourth, the system shall enable Zero-Knowledge Proof-based verification, allowing privacy-preserving validation of critical state transitions—such as authorization grants, case updates, and evidence submissions—without revealing sensitive underlying data. This is achieved by generating zero-knowledge proofs for each state change and anchoring corresponding commitments on-chain, thereby combining transparency with strong confidentiality guarantees.

4.2.2 Non-Functional Requirements

Beyond functional capabilities, the proposed framework must also satisfy several critical performance-oriented non-functional requirements. First, transactions per second (TPS) serves as a key metric to evaluate the system's transaction processing capability during routine judicial operations, reflecting its ability to handle concurrent requests in real-world scenarios. Second, latency is essential to measure the efficiency of transaction execution and confirmation, as timely processing is crucial in legal proceedings where delays may impact case progression. Third, reliability ensures that the framework delivers stable and consistent performance over time, minimizing downtime or unexpected failures that could compromise evidence availability or judicial integrity. Fourth, scalability is required to assess the system's capacity to support growing workloads—such as increasing numbers of cases, users, or evidentiary items—without experiencing significant performance degradation, thereby ensuring long-term viability as the judicial system expands.

4.3 General Design of The Framework

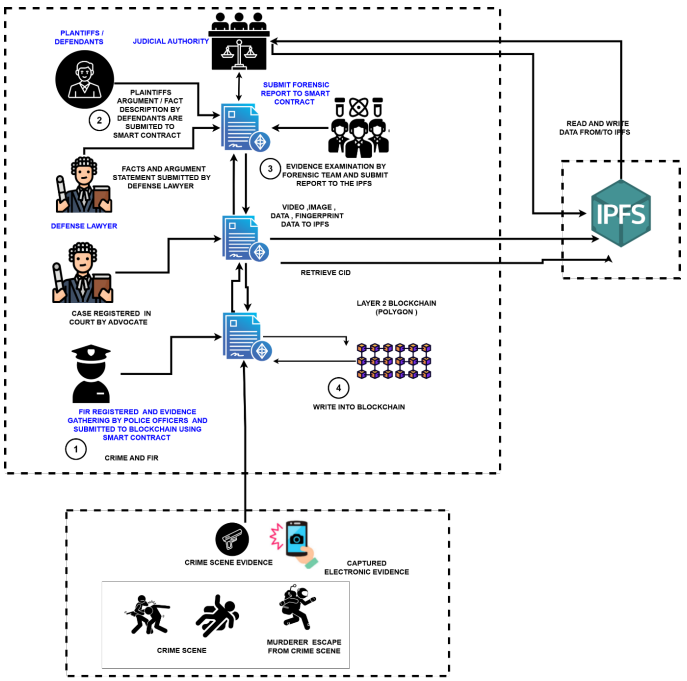


Figure 3. Workflow of the proposed blockchain-based judicial evidence management framework.

The proposed blockchain-based Electronic Evidence Storage System is designed to provide a secure, transparent, and scalable environment for managing judicial evidence using blockchain technology. The framework integrates Polygon CDK, smart contracts, role-based access control (RBAC), and IPFS to ensure secure evidence preservation, controlled access, and decentralized storage. The overall architecture of the framework is illustrated in Figure 3.

Our framework includes two parts: (1) Deployment of the Polygon blockchain network on a local system, and (2) An authentication framework that integrates the judicial process, enabling role-based access control, secure case registration, and management of digital evidence in a transparent yet privacy-preserving

manner.

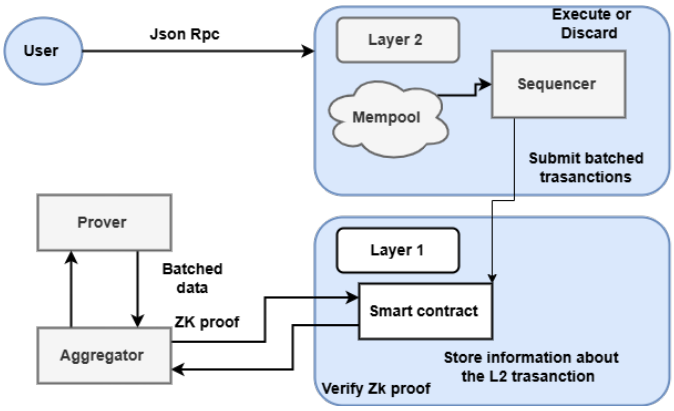


Figure 4. Polygon zkEVM CDK architecture.

4.4 Deployment of Polygon Blockchain

In this work, Polygon zkEVM CDK was employed to deploy a scalable Layer 2 blockchain environment fully compatible with the Ethereum Virtual Machine (EVM). The deployed blockchain infrastructure enables the secure execution of essential judicial workflows, including authority registration, First Information Report (FIR) submission, case registration, evidence verification, and controlled information sharing among authorized participants. Moreover, the underlying blockchain architecture ensures transparency, immutability, and accountability throughout the entire judicial process. The internal mechanism of the deployed system is illustrated in Figure 4.

The operational flow of the Polygon zkEVM CDK can be described as follows. Initially, sequencer nodes collect pending transactions from the mempool (i.e., the transaction waiting area), execute them locally, and group them into batches for further processing. Subsequently, an aggregator retrieves these batches and forwards them to a prover, which generates a Zero-Knowledge Proof (ZKP)—a cryptographic proof that attests to the validity of all transactions in the batch without disclosing any sensitive transaction details. The aggregator then submits both the batch data and the accompanying ZKP to the main Ethereum network (Layer 1). On the Ethereum main chain, a verifier contract checks the submitted proof; upon successful verification, the batch is accepted as a valid block, eliminating the need for the Layer 1 network to re-execute each individual transaction.

For secure evidence management in judicial contexts, the Polygon blockchain network serves as an efficient Layer 2 scaling solution. Polygon leverages a combination of sequencers, transaction pools, and aggregators to process transactions off-chain at Layer 2 and subsequently submits the aggregated results to Layer 1 in the form of zero-knowledge proofs. This approach substantially reduces both operational costs and confirmation latency while achieving significantly high transaction throughput. The detailed deployment configuration, experimental setup, and blockchain environment implementation are discussed in the Experiments and Results section.

4.5 Smart contract Development

The overall system architecture of the proposed framework, which integrates Polygon CDK (zkEVM), smart contracts, and IPFS for decentralized evidence storage, is illustrated in Figure 5. Smart contracts, which are self-executing programs deployed on the blockchain, establish a secure connection between the IPFS storage system and the blockchain network. These smart contracts store a digital fingerprint (hash) of each piece of evidence on the blockchain, thereby ensuring its authenticity, integrity, and immutability. The stored hash serves as proof that the evidence exists and has not been altered after submission.

The proposed framework incorporates several key processes, including authority and user registration, FIR filing, case registration, evidence submission, data verification, and secure information sharing. Before accessing the system, all judicial authorities and users, such as plaintiffs and defendants, are required to complete the registration process. Once registered, a plaintiff can file an FIR, which is subsequently reviewed and verified by the appropriate judicial authorities. Based on the approved FIR, a lawyer can register the corresponding case within the system. All activities performed during these stages are subject to verification and approval by the magistrate.

Police officers are responsible for submitting physical evidence collected from the crime scene, such as weapons and other relevant materials. Subsequently, forensic experts analyze the submitted evidence to identify critical clues, including DNA samples, fingerprints, and other forensic traces that may assist in solving the case. The smart contracts are integrated in such a way that all evidence is securely stored in IPFS, while only the corresponding cryptographic hashes are maintained on the blockchain.

The magistrate oversees and verifies each stage of the judicial process, ensuring the accuracy and legitimacy of all recorded activities. Furthermore, data sharing is conducted in a controlled and secure manner to

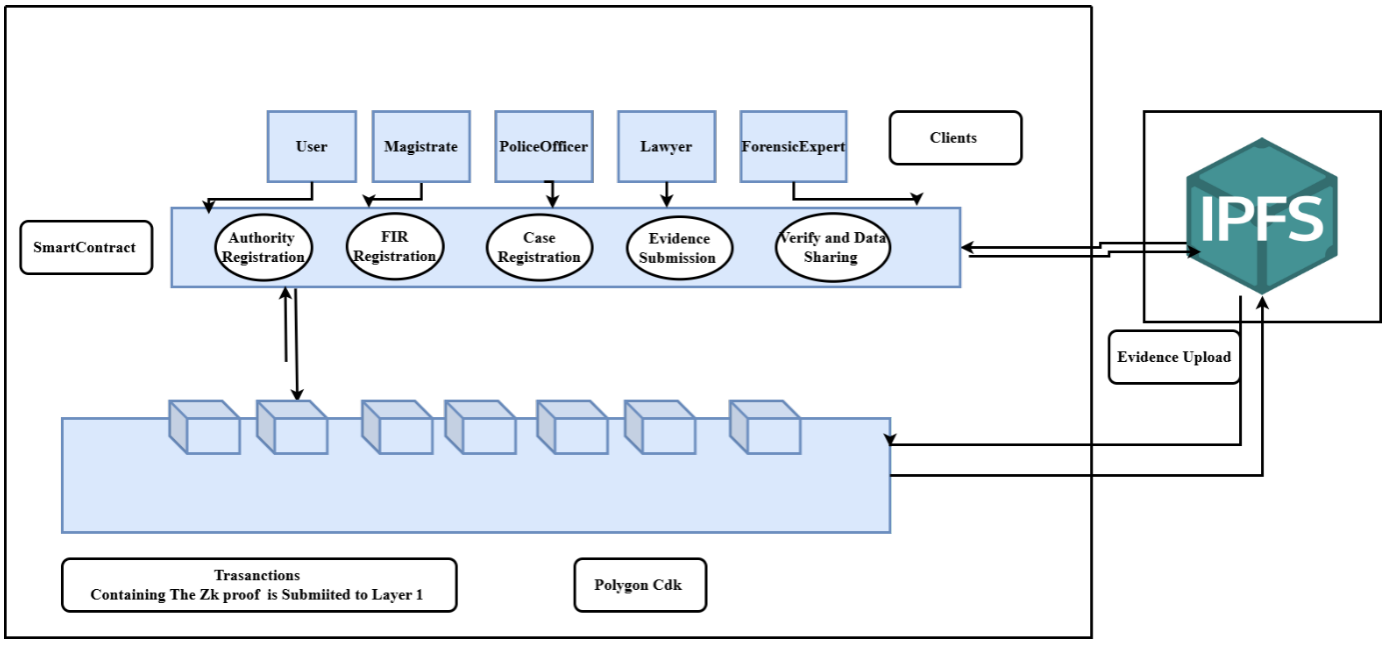


Figure 5. System architecture of the proposed judicial evidence management framework using Polygon CDK (zkEVM) and IPFS.

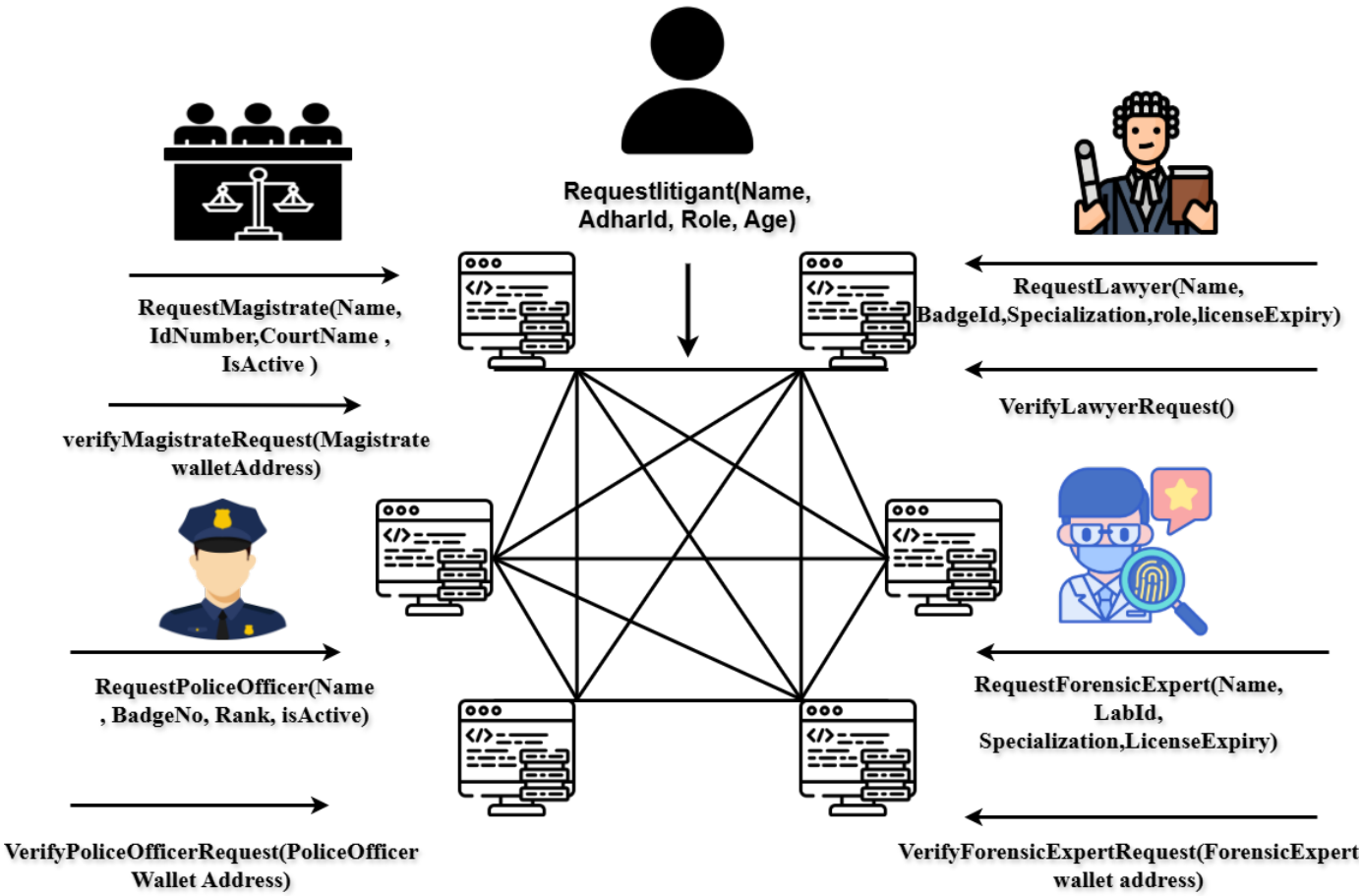


Figure 6. Judicial authorities involved in the proposed framework.

preserve confidentiality and privacy. For instance, only authorized portions of case-related information may be made accessible to defense lawyers and other relevant parties based on their access privileges. Thereafter, all legal proceedings, communications, and case-related activities are managed through the proposed system

until the case is formally resolved and closed.

4.5.1 Authority Registrations

The registration process involves adding judicial authorities and users (plaintiffs and defendants) to the blockchain through smart contracts. As illustrated in Figure 6, the framework supports multiple judicial authorities, including magistrates, lawyers, police officers, forensic experts, and litigants, each with distinct roles and permissions. These authorities submit registration requests through smart contract function calls [7, 25], as formalized in Algorithms 1 and 2 for magistrates. The registration and verification of lawyers follow the procedures defined in Algorithms 3 and 4. The registration requests of the magistrate and other authorities are reviewed and approved by the blockchain administrator. Police officer registration and verification are handled as described in Algorithms 5 and 6. Only verified authorities are granted access to specific smart contract functions according to their assigned roles and permissions. Each entity is associated with a unique wallet address linked to its identity, which serves as a digital signature for authentication and authorization purposes. Litigant registration is defined in Algorithm 7, while role verification for plaintiffs and defendants is detailed in Algorithms 8 and 9, respectively.

Algorithm 1: Request_Magistrate

Input : *name*, *idNumber*, *courtName*
Output: True if request submitted, False if already registered

if msg. sender *is already an active magistrate in magistrates* **then**
  return *False* ; `// Already Registered as a Magistrate`
end
Store the magistrate request in *Pendingmagistrates*[msg. sender];
return *True*;

4.5.2 Case Registration Contract

Forensic expert registration and verification follow the procedures in Algorithms 10 and 11. The Case Registration smart contract handles both FIR registration and case registration by a lawyer. First, verified users (litigants) can file an FIR by providing the details of the crime. This FIR is then checked and approved by authorized police officers. Once the FIR is approved, an authorized lawyer can register a case

Algorithm 2: Verify_Magistrate

Input : *magistrateAddress*
Output: True if verified

Ensure msg. sender has `ADMIN_ROLE`;
Retrieve data from *Pendingmagistrates*[msg. sender];
Move it to *magistrates*[*magistrateAddress*] and set *isActive* = true;
Emit `AuthorityRegistration` event;
Delete entry from *Pendingmagistrates*;
return *True*;

Algorithm 3: Request_Lawyer

Input : *name*, *barID*, *specialization*, *role*, *licenseExpiry*
Output: True if request submitted, False if already registered

if msg. sender *is already an active lawyer* **then**
  return *False* ; `// Already Registered as a Lawyer`
end
Store the lawyer request in *Pendinglawyers*[msg. sender];
return *True*;

Algorithm 4: Verify_Lawyer

Input : *lawyerAddress*
Output: True if verified

Ensure msg. sender has `ADMIN_ROLE`;
Retrieve data from *Pendinglawyers*[*lawyerAddress*];
Move it to *lawyers*[*lawyerAddress*] and set *isActive* = true;
Emit `AuthorityRegistration` event;
Delete entry from *Pendinglawyers*;
return *True*;

based on the FIR. After the case is submitted, it must be verified and approved by an authorized magistrate before it becomes part of the official record. FIR submission by a plaintiff is formalized in Algorithm 12, and the approval process by a police officer is described in Algorithm 13.

4.5.3 Evidence Submission Contract

Case registration by a lawyer and its subsequent approval by the magistrate are formalized in Algorithms 14 and 15, respectively. After a case is successfully registered, the police officer uploads the first evidence (collected from the crime scene) to IPFS

Algorithm 5: Request_Police_Officer

Input : *name*, *badgeNo*, *rank*
Output: True if request submitted, False if already registered

if msg. sender *is already an active PoliceOfficer* **then**
  return *False* ; `// Already Registered as a PoliceOfficer`
end
Store the Police Officer request in *PendingPoliceOfficers*[msg. sender];
return *True*;

Algorithm 6: Verify_Police_Officer

Input : *policeOfficerAddress*
Output: True if verified

Ensure msg. sender has `ADMIN_ROLE`;
Retrieve data from *PendingPoliceOfficers*[*policeOfficerAddress*];
Move it to *PoliceOfficers*[*policeOfficerAddress*] and set *isActive* = true;
Emit `AuthorityRegistration` event;
Delete entry from *PendingPoliceOfficers*;
return *True*;

Algorithm 7: Request_Litigant

Input : *name*, *adharID*, *role*, *Age*
Output: True if request registered, False if already active

if msg. sender *is already registered* **then**
  return *False* ; `// Already Registered as a Litigant`
end
Store directly in *litigants*[msg. sender] with *isActive* = true;
return *True*;

Algorithm 8: Check_Is_Litigant_Plaintiff

Input : *sender*
Output: True if role is Plaintiff

Check if *litigants*[*sender*] is active;
Compare *role* = "Plaintiff";
return *True*;

— a decentralized storage system. Then, a hash of that evidence is created using the keccak256 method (like a digital fingerprint of the data) and stored in the smart contract as plaintiffHash. Next, forensic experts

Algorithm 9: Check_Is_Litigant_Defendant

Input : *sender*
Output: True if role is Defendant

Check if *litigants*[*sender*] is active;
Compare *role* = "Defendant";
return *True*;

Algorithm 10: Request_Forensic_Expert

Input : *name*, *labID*, *specialization*, *licenseExpiry*
Output: True if request submitted, False if already registered

if msg. sender *is already registered* **then**
  return *False* ; `// Already Registered as a Forensic Expert`
end
Store in *PendingForensicExperts*[msg. sender];
return *True*;

Algorithm 11: Verify_Forensic_Expert

Input : *forensicExpertAddress*
Output: True if verified

Ensure msg. sender has `ADMIN_ROLE`;
Retrieve data from *PendingForensicExperts*[*forensicExpertAddress*];
Move it to *forensicExperts*[*forensicExpertAddress*] and set *isActive* = true;
Emit `AuthorityRegistration` event;
Delete entry from *PendingForensicExperts*;
return *True*;

examine the case and upload their findings — like clues or traces — to IPFS as well. All of these activities are recorded on the blockchain as transactions. Finally, the magistrate reviews and verifies the hash of the submitted evidence to confirm its authenticity. Evidence submission is formalized in Algorithm 16, and the hash verification and finalization process is defined in Algorithm 17 [7, 25].

4.5.4 Data Sharing Contract

The Data Sharing Contract enforces controlled access to case information, as implemented in Algorithms 18 and 19, restricting visibility to authorized parties only—specifically the defendant and their legal counsel. It verifies the caller's role before disclosing any case details, and grants access only to approved cases. Evidence files are stored off-chain in IPFS, with only their cryptographic hashes retained on-chain,

Algorithm 12: Submit_FIR_By_Plaintiff

Input : *complaintName*, *accusedName*, *crimeDetails*, *caseType*

Output: True if FIR submitted successfully, False if user is not a registered plaintiff

if msg. sender *is not a registered Plaintiff* (*via* `onlyPlaintiff` *modifier*) **then**
  return *False* ; `// You are not registered as a plaintiff`
end
Increment the FIR counter (*firIdCounter*);
Assign the new FIR ID:
_firId = *firIdCounter.current*();
Create a new FIR in *PendingFIRS*[*_firId*] with the following attributes:;
- *firId* = *_firId*;
- *complaintName* = *_complaintName*;
- *accusedName* = *_accusedName*;
- *crimeDetails* = *_crimeDetails*;
- *timestamp* = current block timestamp;
- *filedBy* = msg. sender;
- *firApprovedBy* = address(0);
- *caseType* = *_caseType*;
- *isApproved* = false;
- *FIRStatus* = FIRStatus.Pending;

Append *_firId* to *PendingFIRSIds* list;
return *True*;

Algorithm 13: Approve_FIR_By_PoliceOfficer

Input : *_firId*

Output: True if FIR approved and moved to *FIRS* mapping, False if not authorized or FIR already processed

if msg. sender *is not a Police Officer* (*via* `onlyPoliceOfficer` *modifier*) **then**
  return *False* ; `// You are not a police officer`
end
if *PendingFIRS*[*_firId*] *does not exist or* (*firApprovedBy* ≠ *address*(*0*) *or FIRStatus* ≠ *FIRStatus.Pending*) **then**
  return *False* ; `// FIR already verified or rejected`
end
Copy data from *PendingFIRS*[*_firId*] into a new entry in *FIRS*[*_firId*]:;
- Set *firApprovedBy* = msg. sender;
- Set *isApproved* = true;
- Set *Firststatus* = FIRStatus.Verified;

Emit `FirRegistered` event with:;
- *firId* = *_firId*;
- *caseType* = value from copied FIR;
- *firApprovedBy* = msg. sender;

Push *_firId* into *FIRIds* array;
Delete *PendingFIRS*[*_firId*] and remove *_firId* from *PendingFIRSIds* list;
return *True*;

thereby minimizing storage overhead while preserving data integrity. This design supports scalable, equitable, and secure information sharing suitable for both small and large judicial institutions.

4.6 IPFS Integration

To support decentralized evidence preservation, the framework integrates InterPlanetary File System (IPFS) for storing digital evidence files. Since storing large files directly on-chain can increase blockchain storage costs and reduce efficiency, evidence files are uploaded to IPFS, while only the corresponding content hashes are stored on the blockchain. The systematic architecture of evidence storage and access, illustrating how files are uploaded to IPFS and their hashes are recorded on-chain, is depicted in Figure 7.

This approach ensures tamper resistance, traceability, and efficient evidence management while preserving the integrity and authenticity of judicial evidence.

5 Experiment Result and Analysis

We developed a decentralized legal evidence storage system using Polygon CDK, a customizable Layer-2 blockchain framework. To prevent blockchain bloat caused by storing large media files, we integrated the InterPlanetary File System (IPFS) through Pinata Cloud for off-chain storage of electronic evidence.

5.1 Tools Used in the Framework

Docker: Docker is an open-source platform designed to simplify the development, deployment, and execution of applications through containerization. In this framework, Docker containers were used to deploy and run the Polygon CDK (zkEVM) environment locally.

Remix IDE: Remix IDE is an open-source integrated development environment used for the development, testing, and deployment of Ethereum smart contracts.

MetaMask: MetaMask is a cryptocurrency wallet that

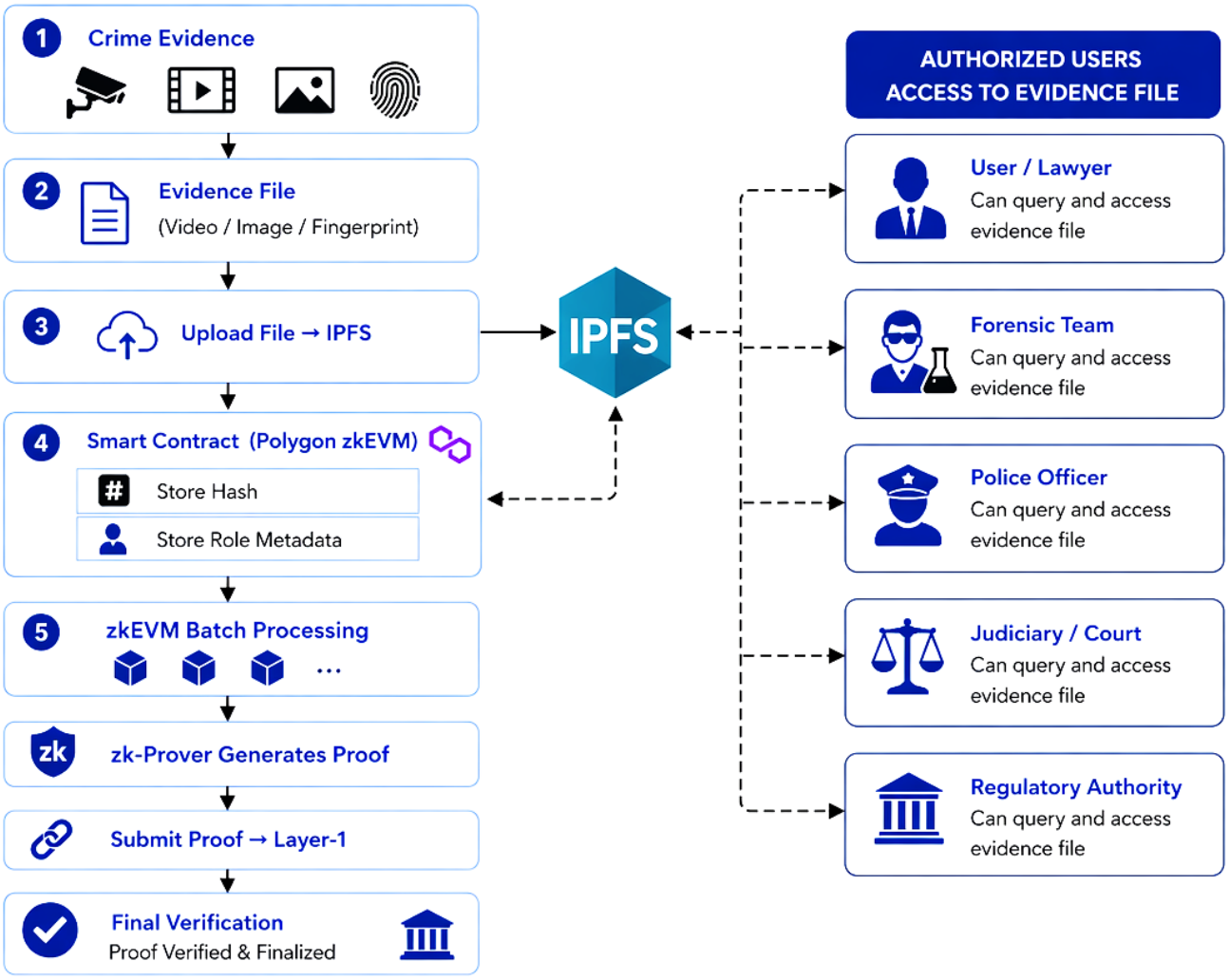


Figure 7. Systematic diagram of evidence storage and access architecture.

is used to manage the accounts of users and other participants within the proposed framework. It also facilitates interaction with the deployed blockchain network and smart contracts.

Polygon CDK (zkEVM) Toolkit: Polygon CDK (Chain Development Kit) is a comprehensive toolkit that provides all the necessary components for deploying, configuring, and operating a Polygon zkEVM-based blockchain network. In this framework, Polygon CDK (zkEVM) was deployed locally using Docker containers to establish and manage the Layer-2 blockchain environment. The implementation was carried out on a Linux system with 4 CPU cores and 12 GB of RAM. Initially, the Polygon zkEVM network was successfully deployed and run in the local environment.

The deployment and running status of the network are illustrated in Figure 8.

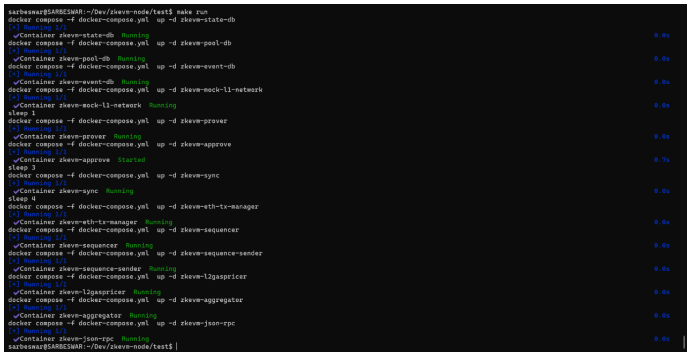


Figure 8. Successful deployment and execution of the Polygon CDK (zkEVM) environment.

To evaluate the performance of the proposed framework, a total of 10,000 transactions were generated and processed using the deployed blockchain environment. As presented in Table 2, the system completed all 10,000 transactions in approximately 700 seconds. Based on this measurement, the average transaction throughput

Algorithm 14: Case_Register

Input : *_firId* — ID of the approved FIR, *_ipfsHash* — IPFS hash of the plaintiff's evidence, msg. sender — the lawyer who is registering the case
Output: True if registration is successful, False otherwise

Ensure msg. sender is a registered lawyer via `authorityRegistry`;
if *FIR with ID = _firId is not approved* **then**
  return *False* ; `// FIR not approved`
end
Increment the internal *caseIdCounter* to get a new case ID (*_caseId*);
Create a new temporary `Case` struct in *TempCases*[*_caseId*] with the following attributes:;
- *caseId* = *_caseId*;
- *firId* = *_firId*;
- *caseFiledBy* = msg. sender;
- *caseApprovedBy* = address(0) ; `// Initial assignment of authority`
- *createdAt* = current block timestamp;
- *isCaseApproved* = false;
- *isClosed* = false;
- *plaintiffEvidenceHash* = *_ipfsHash*;
- *defendantEvidenceHash* = "No evidence till now";

return *True*;

Algorithm 15: Approve_Case_By_Magistrate

Input : *_caseId* — ID of the case
Output: True if approval is successful, False otherwise

Ensure msg. sender is a registered magistrate via `authorityRegistry`;
if *TempCases*[*_caseId*] *does not exist or isCaseApproved is true* **then**
  return *False* ; `// Case not found or already approved`
end
Copy the temporary case data from *TempCases*[*_caseId*] to the permanent *Cases*[*_caseId*] mapping;
Set *caseApprovedBy* = msg. sender;
Set *isCaseApproved* = true;
Emit `CaseRegistered` event;
Delete entry from *TempCases*;
return *True*;

Algorithm 16: Submit_Evidence

Input : *_firId* — FIR ID, *_evidenceUri* — Evidence URI (e.g., IPFS hash)
Output: Evidence stored in either *PlaintiffEvidences* or *DefendantEvidences*

Determine the role of msg. sender and set *partyType* accordingly:;
if msg. sender *is a Defendant's Lawyer* **then**
  Set *partyType* = "Defendant";
  Target *DefendantEvidences*[*_firId*];
else if msg. sender *is a Police Officer OR Lawyer OR Forensic Expert* **then**
  Set *partyType* = "Plaintiff";
  Target *PlaintiffEvidences*[*_firId*];
end
else
  return *False* ; `// Not authorized to submit evidence`
end
Increment the Evidence ID counter;
Assign new *evidenceId* = current counter value;
Create a new `Evidence` struct with the following attributes:;
- *evidenceId* = new evidenceId;
- *firId* = *_firId*;
- *evidenceUri* = *_evidenceUri*;
- *submittedBy* = msg. sender;
- *timestamp* = current block timestamp;
- *isApprovedEvidence* = false;

if *partyType* = *"Plaintiff"* **then**
  Push to *PlaintiffEvidences*[*_firId*];
end
else if *partyType* = *"Defendant"* **then**
  Push to *DefendantEvidences*[*_firId*];
end
Emit `EvidenceSubmitted`(msg. sender, *evidenceId*, *partyType*) event;
return *True*;

was calculated as follows:

$$\text{TPS} = \frac{10,000 \text{ transactions}}{700 \text{ seconds}} \approx 14.29 \text{ TPS} \quad (1)$$

This result reflects the system's performance under a basic local setup, with the trend of transaction processing time graphically illustrated in Figure 9. It is worth noting that Polygon zkEVM CDK chains are capable of achieving significantly higher throughput under optimized conditions; however, the observed

Algorithm 17: Verify_And_Finalize_Evidence [7, 25]

Input : *_caseId* — ID of the Case, *_firId* — ID of the FIR
Output: True if evidence hashes are generated and stored in *CaseRegistration*

if msg. sender *is not a registered Magistrate* **then**
  return *False* ; `// You are not a Magistrate`
end
Initialize empty byte arrays:;
  – *plaintiffData* = empty;
  – *defendantData* = empty;
foreach *evidence in PlaintiffEvidences*[*_firId*] **do**
  Append each *evidenceUri* to *plaintiffData*;
end
foreach *evidence in DefendantEvidences*[*_firId*] **do**
  Append each *evidenceUri* to *defendantData*;
end
Generate Keccak256 hashes:;
  – *plaintiffHash* = keccak256(*plaintiffData*);
  – *defendantHash* = keccak256(*defendantData*);
Convert both hashes to hex strings using `bytes32ToHex` function;
Call *CaseRegistration* contract to store:;
  – *caseId* = *_caseId*;
  – *firId* = *_firId*;
  – *plaintiffEvidenceHash* = *plaintiffHash*;
  – *defendantEvidenceHash* = *defendantHash*;
  – *isApprovedEvidence* = true;
  – *verifiedBy* = msg. sender;
  – *verificationTimestamp* = current block timestamp;
return *True*;

performance varies depending on the workload characteristics and the specific system configuration employed.

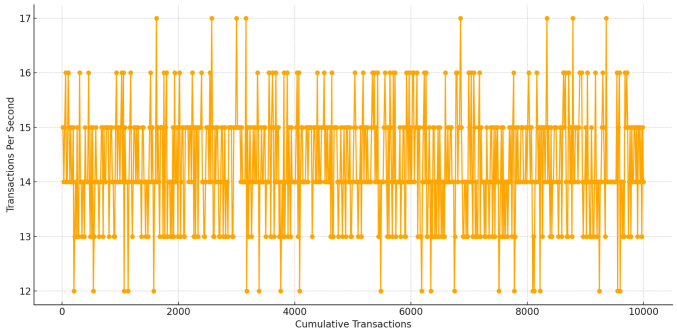


Figure 9. Transactions per second (TPS) achieved by the proposed framework.

Algorithm 18: Get_Case_Details_For_Defense

Input : *_caseId* — ID of the case
Output: FIR and Case details if authorized

if msg. sender *is not a registered Defendant or Defendant's Lawyer (via* `onlyDefendantOrLawyer` *modifier)* **then**
  return *False* ; `// Not authorized: Only defendant or their lawyer`
end
Retrieve the Case using `getCaseById`(*_caseId*) from the *CaseRegistration* contract;
if *case.isCaseApproved is not true* **then**
  return *False* ; `// Case not approved yet`
end
Get the corresponding FIR using `getFIRById`(*case.firId*);
Retrieve evidence hashes:;
  – *plaintiffEvidenceHash* = `getPlaintiffEvidenceHash`(*fir.firId*);
  – *defendantEvidenceHash* = `getDefendantEvidenceHash`(*fir.firId*);
return (*caseId, firId, complaintName, crimeDetails, firTimestamp, plaintiffEvidenceHash, defendantEvidenceHash, isCaseApproved, isClosed*);

Algorithm 19: Get_Case_Summary_For_Defense_Lawyer

Input : *_caseId* — ID of the case
Output: Case Summary if caller is defendant's lawyer

if msg. sender *is not a Defendant's Lawyer via* `authorityRegistry.isDefendantLawyer`(msg. sender) **then**
  return *False* ; `// Only defendant's lawyer can access`
end
Retrieve the Case using `getCaseById`(*_caseId*) from the *CaseRegistration* contract;
if *case.isCaseApproved is not true* **then**
  return *False* ; `// Case not approved yet`
end
Retrieve evidence hashes:;
  – *plaintiffEvidenceHash* = `getPlaintiffEvidenceHash`(*case.firId*);
  – *defendantEvidenceHash* = `getDefendantEvidenceHash`(*case.firId*);
return (*caseId, firId, caseFiledBy, caseApprovedBy, createdAt, plaintiffEvidenceHash, defendantEvidenceHash, isCaseApproved, isClosed*);

Table 2. Transaction processing performance.

No. of Transactions	Time Taken (s)	Avg. Time per Tx (ms)	Throughput (TPS)
1,000	72.1	72.1	13.9
2,000	145.4	72.7	13.8
3,000	218.6	72.9	13.7
4,000	291.9	73.0	13.7
5,000	365.2	73.0	13.7
6,000	438.4	73.1	13.7
7,000	511.7	73.1	13.7
8,000	584.9	73.1	13.7
9,000	658.2	73.1	13.7
10,000	700.0	70.0	14.29

Despite using modest hardware (4-core CPU, 12 GB RAM), the system achieved approximately 14.29 TPS with a maximum of 250 transactions per batch, as shown in Figure 10. With higher-spec infrastructure and optimized batching mechanisms, the system can achieve significantly higher throughput. This highlights the system's potential to scale horizontally and vertically, making it suitable for both large urban courts and smaller rural implementations.

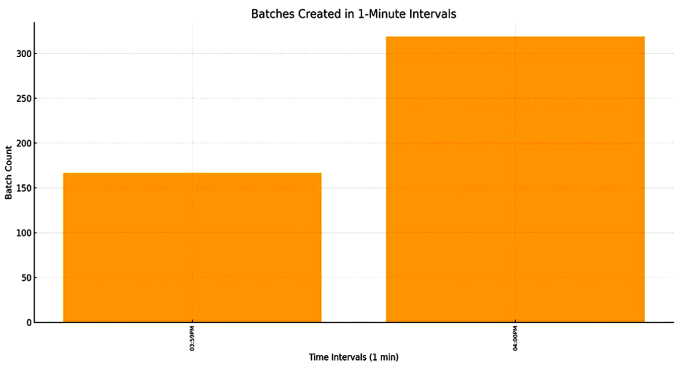


Figure 10. Batch processing performance measured over 1-minute time intervals.

To implement and deploy the proposed smart contracts, the Polygon zkEVM local network was configured in the MetaMask wallet for transaction submission. The Layer-2 network was configured with the network name Polygon zkEVM Local, RPC URL [1], and Chain ID 1001. In addition, a Layer-1 network, Geth Local, was configured as a local Ethereum test network. The Layer-1 network was set up with the network name Geth, RPC URL [2], and Chain ID 1337.

Several pre-funded account addresses containing test cryptocurrency were provided to facilitate smart contract deployment and transaction execution within the testing environment. For example, one such account address is `0xf39Fd6e51aad88F6F4ce6aB8827279cffFb92266`. These accounts were imported into the MetaMask wallet and used to deploy the smart contracts associated with the proposed framework. The successful deployment of the smart contracts is illustrated in Figure 11.

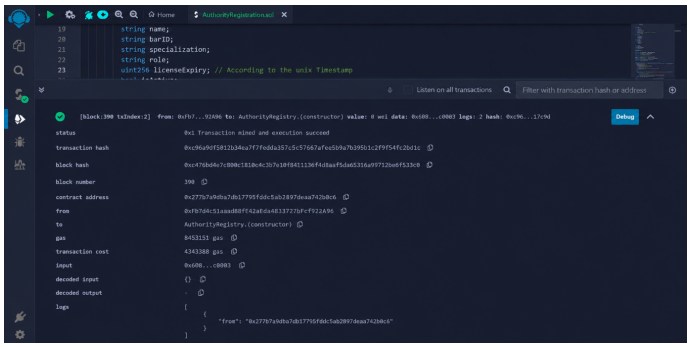


Figure 11. Successful deployment of the AuthorityRegistry smart contract, showing the generated contract address and transaction details in the Remix IDE environment.

These smart contracts which are mentioned above in smart contract design section 4.4 are deployed using and tested through Remix IDE. Once smart contract deployed through deployer account / owner account he will be Admin of this framework. The administrator role is enforced through the AccessControl smart contract, which provides robust and tamper-resistant role separation. Judicial Authorities can register themselves using functions, Admin will verify them using their KYC or details. Only verified authorities are permitted to participate in subsequent judicial activities. A plaintiff can submit an FIR through this framework, which remains in a pending state until verified. Once a police officer reviews the FIR details and approves it, as shown in Figure 12, a lawyer can register the corresponding case by referencing the approved FIR. All of the above actions are subject to verification and approval by the presiding magistrate. Evidence can be submitted by police officers, lawyers, and forensic experts during the case trials.

To evaluate the cost-effectiveness of the proposed system, both the operational benefits and associated expenses were examined. The transaction costs are listed in Table 3. The INR values are calculated based on an exchange rate of approximately 1 MATIC $\approx$ INR 73, applicable at the time of the experiment. The results demonstrate that the cost of every transaction in this framework is negligibly low. The main costs come from running the server and keeping the IPFS storage working. Even with these costs, the system is still more

[1] http://localhost:8123
[2] http://localhost:8545

Table 3. Transaction execution cost on polygon zkEVM local network.

Sl. No.	Function	Gas Consumed	Cost (MATIC)	Cost (INR ¤)
1	RequestMagistrate	98,128	0.00009813	0.0072
2	RequestLawyer	144,503	0.00014450	0.0106
3	RequestPoliceOfficer	97,926	0.00009793	0.0072
4	RequestLitigant	138,346	0.00013835	0.0102
5	VerifyPoliceOfficer	38,226	0.00003823	0.0028
6	VerifyMagistrate	37,892	0.00003789	0.0028
7	VerifyLawyer	38,012	0.00003801	0.0028
8	VerifyLitigant	38,076	0.00003808	0.0028
9	RequestForensicExpert	132,198	0.00013220	0.0097
10	VerifyForensicExpert	38,115	0.00003812	0.0028
11	RegisterFIR	174,263	0.00017426	0.0128
12	ApproveFIR	42,311	0.00004231	0.0031
13	RejectFIR	26,137	0.00002613	0.0019
14	RegisterCase	212,420	0.00021242	0.0156
15	ApproveCase	42,846	0.00004285	0.0032
16	RejectCase	26,603	0.00002660	0.0020
17	SubmitEvidence	166,198	0.00016620	0.0122
18	VerifyEvidence	48,311	0.00004831	0.0036
19	ShareEvidence	62,405	0.00006241	0.0046

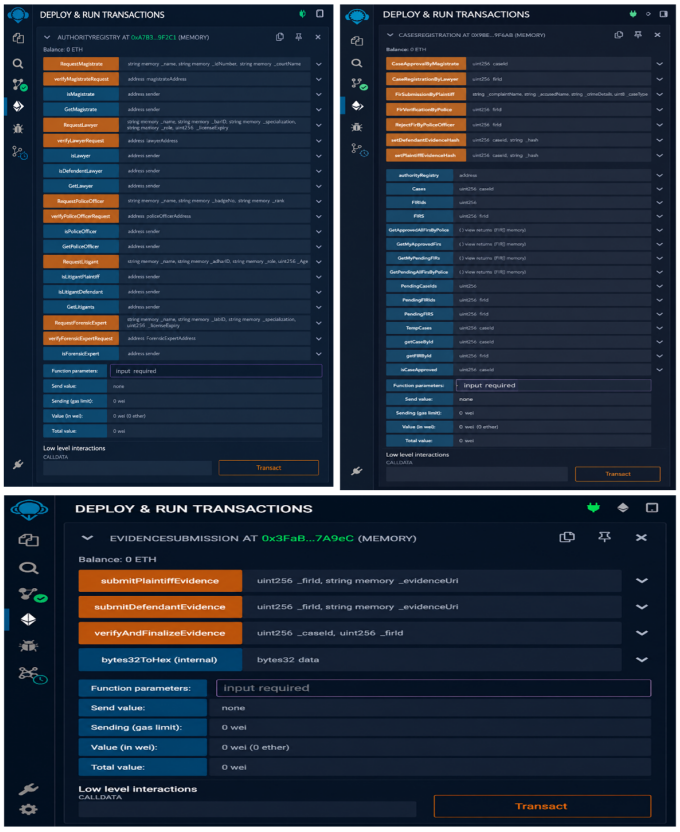


Figure 12. Interfaces of the deployed AuthorityRegistry, CaseRegistration, and EvidenceSubmission smart contracts, illustrating authority registration, case registration, evidence submission, and verification functions.

affordable than older methods. The gas consumption per transaction is also visualized in Figure 13, which highlights the cost distribution across different smart contract functions.

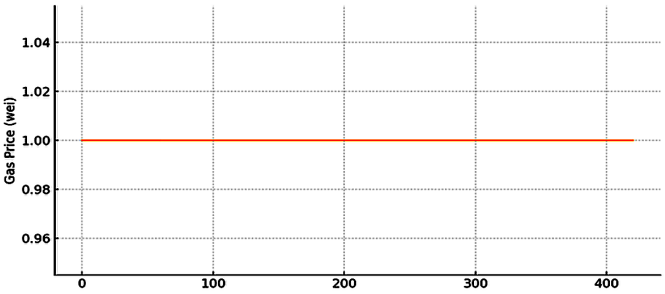


Figure 13. Gas consumption cost per transaction in the proposed framework.

We also tested how fast IPFS can upload and download different types of files. As expected, bigger files took longer to upload and download. Even for large files, the transfer speed remained sufficient for practical legal applications. The evidence file upload process to the IPFS network via Pinata Cloud is illustrated in Figure 14.

6 Discussion

This section presents a comparative discussion shown in Table 4. of the selected blockchain-based evidence-management frameworks, focusing on

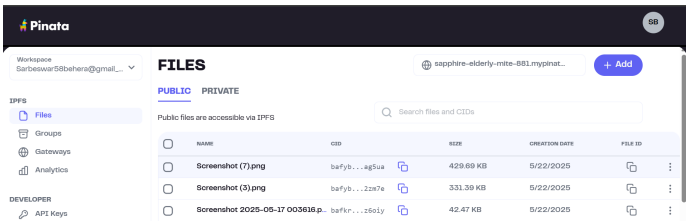


Figure 14. Evidence file upload and storage in the IPFS network via Pinata Cloud.

scalability, throughput, latency, privacy mechanisms, verification models, security, Reliability and cryptographic strength.

Scalability and Throughput (TPS): Most existing frameworks exhibit limited scalability; they are only suitable for small networks and do not support large-scale deployment.

Latency: Latency refers to time delay (how long it takes for a system to respond). Low latency means fast response and better performance, while high latency means slow response and poor performance. Latency may also depend on network conditions.

Privacy and Encryption: Earlier systems primarily relied on conventional encryption techniques for data protection, without incorporating advanced cryptographic verification mechanisms such as zero-knowledge proofs. In contrast, the proposed framework enhances privacy by combining zero-knowledge proof-based verification with decentralized storage using IPFS, enabling both confidentiality and verifiable correctness of operations.

Confidentiality: Confidentiality in the proposed framework is maintained using role-based access control using smart contracts and cryptographic security techniques implemented through smart contracts. Only authorized users, such as plaintiffs, lawyers, magistrates and forensic experts, are permitted to access specific evidence records.

Zero-Knowledge Proof-Based Verification: In the proposed system, zero-knowledge proofs are used to verify the correctness of state transitions related to case handling and evidence management, without revealing sensitive underlying inputs. In particular, the system allows validation of operations such as role-based authorization, case lifecycle updates, and evidence submission integrity. Sensitive information such as the actual evidence content is stored off-chain (e.g., IPFS) and represented on-chain using cryptographic commitments.

While evidence metadata such as evidence identifiers and transaction references may remain partially visible on-chain, sensitive attributes including the full evidence details and optionally the identity of the submitter are concealed depending on the access control policy. Only cryptographic commitments, proof verification results, and minimal transaction metadata are recorded on-chain to ensure integrity and auditability without exposing confidential information.

Security and Reliability: The proposed framework ensures security and reliability through the integration of Polygon zkEVM, smart contracts, cryptographic hashing, and decentralized storage. zkEVM enables validation of transaction correctness through zero-knowledge proofs without requiring full re-execution on Layer 1, thereby improving trust and system efficiency. Evidence integrity is preserved by storing actual files off-chain on IPFS and recording only their cryptographic hashes on-chain, ensuring tamper detection as any modification to the data results in a hash mismatch.

Reliability is maintained through immutable blockchain records and decentralized storage mechanisms. All judicial activities, including authority

Table 4. Comparison of the proposed framework with existing frameworks.

Authors	Scalability	TPS	Latency	ZK Proof	IPFS	Encryption	RBAC	Security	Reliability
Bonomi et al. [5]	Not scalable	Low	High	No	No	Basic	No	Basic	Low
Tian et al. [6]	Yes	Moderate	Low	No	No	Cryptographic	No	Moderate	Moderate
Brotsis et al. [10]	Not scalable	Low–Moderate	Medium	No	No	AES-like	No	Moderate	Low
Verma et al. [12]	Not scalable	Low	High	No	Yes	AES-style	Limited	Moderate	Moderate
Wang et al. [14]	Not scalable	Low	High	No	No	Basic	No	Moderate	Low
Rani et al. [19]	Scalable	Moderate	Low	No	Yes	Cryptographic	Yes	High	High
Liu & Zheng [20]	Scalable	Moderate	Low	No	Yes	Encrypted	Limited	High	High
Tortola et al. [21]	Potentially high	High	Low	Yes	No	ZK-based	No	High	Moderate
This paper	**Highly scalable**	**High**	**Low**	**Yes**	**Yes**	**ZK-based**	**Yes**	**High**	**High**

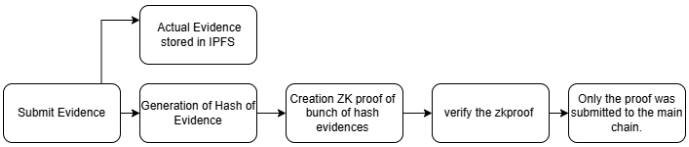


Figure 15. Proposed Evidence Submission and ZK-Proof Verification Process.

registration, FIR approval, case registration, evidence submission, and verification, are permanently recorded on the blockchain, thereby maintaining an unbroken chain of evidence. Furthermore, IPFS supports distributed storage using Content Identifiers (CIDs) [26], enabling evidence traceability and recovery in the event of storage failures or node unavailability. The complete evidence submission and zero-knowledge proof verification process is illustrated in Figure 15.

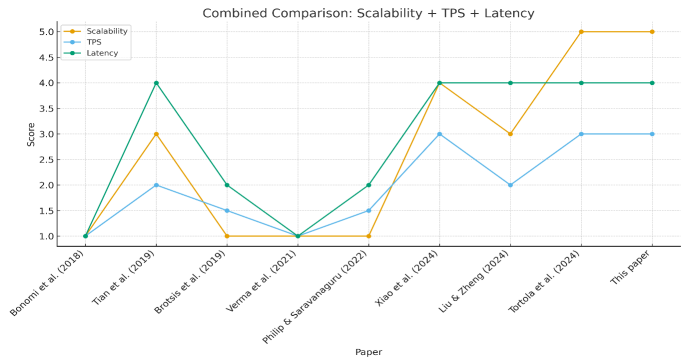


Figure 16. Comparison of scalability, TPS, and latency among different frameworks.

The comparative analysis presented in Figure 16 illustrates that the proposed framework offers an improved solution that successfully bridges the limitations observed in earlier research frameworks, achieving superior scalability, higher throughput, and lower latency.

7 Limitations

Despite the promising results of the proposed digital evidence management framework, certain limitations should be acknowledged.

First, the proposed system has been tested only in a controlled experimental environment and has not yet been deployed in real-world judicial or law enforcement settings. Therefore, practical challenges such as integration with existing legal workflows, network reliability, and user adoption have not yet been fully evaluated.

Second, the performance evaluation was conducted using simulated transactions rather than large-scale real-world workloads involving extensive user activity and large evidence files. As a result, scalability under high-concurrency and production-grade conditions requires further validation. In the experimental setup, the observed throughput was approximately 14.29 transactions per second (TPS), reflecting local execution constraints.

Third, the security of the system depends on proper key management by authorized users. Compromise or loss of private keys may lead to unauthorized access or loss of control over evidence-related operations. Additionally, risks related to decentralized storage, such as IPFS content identifier (CID) linkability and persistence assumptions, and potential misuse of administrative privileges, are acknowledged as part of the system's threat surface.

The performance evaluation of the proposed framework was conducted in a controlled local environment, where the observed throughput reached approximately 14.29 transactions per second (TPS). Although modern Layer-2 blockchain platforms such as Polygon zkEVM are capable of supporting significantly higher throughput under optimized and large-scale deployment conditions, such performance was not experimentally validated in this study.

8 Conclusion

In this paper, we presented a secure and scalable system for managing electronic evidence in judicial processes, built on Polygon zkEVM and IPFS. The system was deployed locally using Docker containers on a standard server with 4 CPU cores and 12 GB RAM, achieving approximately 14.29 TPS, demonstrating efficient performance even on limited hardware.

The proposed framework offers several key advantages. First, Zero-Knowledge Proofs ensure transaction validity without revealing internal computation details, while evidence hashes and access logs are stored on-chain and actual evidence content remains off-chain in IPFS. Second, the Layer 2 architecture provides high scalability, making it suitable for national-level court systems and large-scale legal networks. Third, Role-Based Access Control is enforced through smart contracts, ensuring that only authorized participants can perform specific actions. Fourth, IPFS integration enables efficient decentralized storage of large evidentiary files.

The system supports essential judicial functions, including uploading, reviewing, sharing, and

downloading digital evidence, and is designed to serve both large urban courts and smaller rural institutions.

Beyond judicial applications, the architecture can be extended to software engineering collaboration, commercial contract documentation, healthcare record management, supply chain tracking, and academic certificate issuance.

Future work will focus on optimizing the consensus mechanism and prover efficiency to further enhance throughput and robustness in large-scale deployments. Overall, this work contributes a practical and forward-looking solution for secure, transparent, and efficient evidence management using blockchain technology.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Klasén, L., Fock, N., & Forchheimer, R. (2024). The invisible evidence: Digital forensics as key to solving crimes in the digital age. *Forensic science international, 362*, 112133. [CrossRef]

[2] Miller, C. M. (2023). A survey of prosecutors and investigators using digital evidence: A starting point. *Forensic Science International: Synergy, 6*, 100296. [CrossRef]

[3] Davis, M., Manes, G., & Shenoi, S. (2005, February). A network-based architecture for storing digital evidence. In *IFIP International Conference on Digital Forensics* (pp. 33-42). Boston, MA: Springer US. [CrossRef]

[4] Kumar, G., Saha, R., Lal, C., & Conti, M. (2021). Internet-of-Forensic (IoF): A blockchain based digital forensics framework for IoT applications. *Future Generation Computer Systems, 120*, 13-25. [CrossRef]

[5] Bonomi, S., Casini, M., & Ciccotelli, C. (2018). B-coc: A blockchain-based chain of custody for evidences management in digital forensics. *arXiv preprint arXiv:1807.10359*. [CrossRef]

[6] Tian, Z., Li, M., Qiu, M., Sun, Y., & Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences, 491*, 151-165. [CrossRef]

[7] Polygon. (2026). *Polygon CDK: Private blockchains with public liquidity*. Polygon Developer Docs. Retrieved March 06, 2026, from https://docs.polygon.technology/zkEVM/overview/

[8] Trautwein, D., Raman, A., Tyson, G., Castro, I., Scott, W., Schubotz, M., ... & Psaras, Y. (2022, August). Design and evaluation of IPFS: a storage layer for the decentralized web. In *Proceedings of the ACM SIGCOMM 2022 Conference* (pp. 739-752). [CrossRef]

[9] Atlam, H. F., Ekuri, N., Azad, M. A., & Lallie, H. S. (2024). Blockchain forensics: A systematic literature review of techniques, applications, challenges, and future directions. *Electronics, 13*(17), 3568. [CrossRef]

[10] Brotsis, S., Kolokotronis, N., Limniotis, K., Shiaeles, S., Kavallieros, D., Bellini, E., & Pavué, C. (2019, June). Blockchain solutions for forensic evidence preservation in IoT environments. In *2019 IEEE conference on network softwarization* (*NetSoft*) (pp. 110-114). IEEE. [CrossRef]

[11] Chen, S., Zhao, C., Huang, L., Yuan, J., & Liu, M. (2020). Study and implementation on the application of blockchain in electronic evidence generation. *Forensic Science International: Digital Investigation, 35*, 301001. [CrossRef]

[12] Verma, A., Bhattacharya, P., Saraswat, D., & Tanwar, S. (2021). NyaYa: Blockchain-based electronic law record management scheme for judicial investigations. *Journal of Information Security and Applications, 63*, 103025. [CrossRef]

[13] Sun, X., Yu, F. R., Zhang, P., Sun, Z., Xie, W., & Peng, X. (2021). A survey on zero-knowledge proof in blockchain. *IEEE network, 35*(4), 198-205. [CrossRef]

[14] Wang, X., Wu, Y. C., & Ma, Z. (2024). Blockchain in the courtroom: Exploring its evidentiary significance and procedural implications in U.S. judicial processes. *Frontiers in Blockchain, 7*, 1306058. [CrossRef]

[15] Brotsis, S., Grammatikakis, K. P., Kavallieros, D., Mazilu, A. I., Kolokotronis, N., Limniotis, K., & Vassilakis, C. (2023). Blockchain meets Internet of Things (IoT) forensics: A unified framework for IoT ecosystems. *Internet of Things, 24*, 100968. [CrossRef]

[16] Lewulis, P. (2023). The use of social media evidence in the courtroom: A survey of lawyer's experiences in Poland. *Computer Law & Security Review, 51*, 105886. [CrossRef]

[17] Malik, A., & Sharma, A. K. (2023). Blockchain-based digital chain of custody multimedia evidence preservation framework for internet-of-things. *Journal of Information Security and Applications, 77*, 103579. [CrossRef]

[18] Ratul, M. H. A., Mollajafari, S., & Wynn, M. (2024). Managing digital evidence in cybercrime: Efforts towards a sustainable blockchain-based solution. *Sustainability, 16*(24), 10885. [CrossRef]

[19] Rani, D., Gill, N. S., Gulia, P., Yahya, M., Ahanger, T. A., Hassan, M. M., ... & Shukla, P. K. (2025). A secure digital evidence preservation system for an iot-enabled smart environment using ipfs, blockchain, and smart contracts. *Peer-to-Peer Networking and Applications, 18*(2), 5. [CrossRef]

[20] Liu, S., & Zheng, Q. (2024). A study of a blockchain-based judicial evidence preservation scheme. *Blockchain: Research and Applications, 5*(2), 100192. [CrossRef]

[21] Tortola, D., Lisi, A., Mori, P., & Ricci, L. (2024). Tethering Layer 2 solutions to the blockchain: A survey on proving schemes. *Computer Communications, 225*, 289-310. [CrossRef]

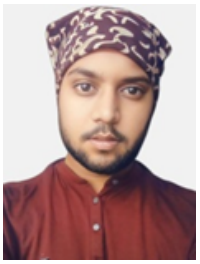
[22] Sanka, A. I., & Cheung, R. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications, 195*, 103232. [CrossRef]

[23] Kuznetsov, O., Rusnak, A., Yezhov, A., Kanonik, D., Kuznetsova, K., & Karashchuk, S. (2024). Enhanced security and efficiency in blockchain with aggregated zero-knowledge proof mechanisms. *IEEE Access, 12*, 49228-49248. [CrossRef]

[24] Lavaur, T., Detchart, J., Lacan, J., & Chanel, C. P. (2023). Modular zk-rollup on-demand. *Journal of Network and Computer Applications, 217*, 103678. [CrossRef]

[25] Iglesias, M. J. F. (2023). *Introduction to blockchain, smart contracts and decentralized applications*. https://www.researchgate.net/publication/370120309

[26] IPFS. (2026). *Content addressing*. IPFS Docs. Retrieved March 06, 2026, from https://docs.ipfs.tech/concepts/content-addressing/



Sarbeswar Behera is a MCA student in the Department of Computer Science and Information Technology at Central University of Haryana, India. He completed his Bachelor of Computer Applications (BCA) from Utkal University, Odisha. His research interests include blockchain technology, Web3 technologies, Layer 2 scaling solutions, cryptography, data structures and algorithms, and system design, distributed systems, and scalability in backend architectures. His current work focuses on decentralized systems and blockchain-based applications. (Email: sarbeswar58behera@gmail.com)

Suraj Arya currently working as Assistant Professor in the Department of Computer Science and Information Technology at Central University of Haryana, India. His academic qualifications are Ph.D.(Computer Science), M.Phil.(Computer Science) and M.Tech (Computer Science and Engineering). His research interests focus on Machine learning (ML), Deep Learning , Internet of things (IoT), Data warehousing and mining, System automation and patent writing. He has been granted and filed many patents. He has also published many research articles in international journals, book chapters, and conferences. (Email: surajarya@cuh.ac.in)