UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Comparative Study of Consensus Mechanisms for Digital Image Evidence Validation using Smart Contracts on Layer 2 Polygon Blockchain

Shameelah Mohammed Mahmoud[1*] & Mohd Fo'ad Rohani[2]
Faculty of Computing, Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
Email: mshmila04@gmail.com[1]; foad@utm[2]

Submitted: 4/9/2025. Revised edition: 21/1/2026. Accepted: 21/1/2026. Published online: 10/6/2026
DOI: https://doi.org/10.11113/ijic.v16n1.593

Abstract—Ensuring the authenticity, integrity, and reliability of digital image evidence is a persistent challenge in forensic and legal domains due to the vulnerabilities of centralized evidence management systems. This study compares three blockchain consensus mechanisms—Proof of Existence (PoE), Proof of Ownership (PoOW), and Zero-Knowledge Ethereum Virtual Machine (zkEVM)—to assess their effectiveness in securing and validating digital image evidence on the Layer 2 Polygon network. Forensic images were stored on the InterPlanetary File System (IPFS), with each consensus model registering tamper-evident Content Identifiers (CIDs) on-chain via dedicated smart contracts. The evaluation considered performance metrics including latency, gas usage, transaction fees, throughput, scalability, and privacy protection. The findings revealed that PoE demonstrated the best overall efficiency, achieving a latency of 3730 ms, a transaction fee of 0.001321 ETH, and a throughput of 0.176 TPS, making it well-suited for real-time applications such as timestamping and immediate evidence submission. PoOW, although more computationally demanding, achieved the highest gas-refund rate at 89%, making it ideal for ownership verification and traceability, such as copyright and asset provenance. Meanwhile, zkEVM provided a well-rounded performance profile with moderate transaction costs and latency. It is powerful for privacy-preserving applications that require cryptographic guarantees, especially in enterprise and regulatory settings. This comparative evaluation highlights the unique advantages and limitations of each approach, providing critical insights into selecting the most suitable blockchain-based consensus mechanism for the transparent and tamper-resistant validation of digital forensic evidence.

***Keywords*—Blockchain, Smart-contract, Image validation, Consensus mechanisms, Digital image evidence**

I. INTRODUCTION

Forensic investigations are increasingly challenged by the rapid growth of digital multimedia and the widespread use of image-based evidence. Although digital images are widely used in legal processes and legislation, ensuring the authentication, integrity, and traceability of such evidence is difficult because it can be easily manipulated, and secure verification structures are often lacking. With the continuing development of digital tampering techniques, the credibility and admissibility of image-based evidence in court have become significant concerns for forensic investigators and legal practitioners [1].

Despite their role in data protection, the centralized Digital Evidence Management System (DEMS) remains vulnerable to unauthorized access, insider threats, and data modification, creating critical weak points in the chain of custody and undermining trust in evidence storage and verification operations [2]. Existing approaches to image and audit-trail verification, such as cryptographic hashes, certificate-based techniques (PKI), and audit logs, offer only partial protection because they rely on centralized components and lack truly tamper-proof mechanisms, which introduces single points of failure and limits transparency [3]. Additionally, manual image authentication is time-consuming and complex, which increases the risk of human error, latency, and operational inefficiency. These limitations highlight the need for an automated, scalable, and secure method for validating digital image evidence [4].

This paper proposes a blockchain-based validation system for digital image evidence that integrates smart contracts, decentralized storage via the InterPlanetary File System (IPFS), and three distinct consensus models. The proposed solution is designed to prevent tampering, enable verification, and support the legal acceptability of digital records by ensuring that images are stored using IPES, which are then registered on the Polygon blockchain through smart contracts. The core contributions of this research are summarized as follows:

- A blockchain-based digital image validation system using IPFS for tamper-evident CID generation and smart contracts for immutable on-chain registration and verification.
- A comparison of three consensus mechanisms, PoE, PoOW, and zkEVM, by examining the most important performance forensics such as. Latency, gas consumption, transaction cost, and throughput.
- Planning and implementation of automated workflows based on Node.js and Hardhat. model and benchmark the cost, speed, and resilience of digital image validation transactions under realistic circumstances.
- A legally compliant framework of secure evidence management that law enforcement, forensics laboratories, and judicial security could use.

In contrast to traditional evidence management systems, the proposed solution automates the key forensic tasks, including timestamping, ownership verification, and privacy-preserving disclosure with the help of smart contracts. PoE enables the lightweight anchoring of evidentiary material, PoOW provides efficient verification of provenance, and zkEVM enables the provision of advanced privacy guarantees while maintaining verifiability. These consensus models are discussed together to determine the best trade-offs among performance, security, and forensic applicability. By facilitating the continuum between what blockchain science can do and what digital investigations require, the study provides a reliable and self-confident basis for the future generation of evidence validation apparatuses in the digital field.

II. RELATED WORK

Multiple research studies have been devoted to the topic of challenges and innovations related to the management, validation, and security of digital image evidence, particularly in the fields of forensics and law. Here, a detailed review of the literature on traditional evidence management systems, blockchain implementation in digital forensics, smart contract applications, and consensus mechanisms is provided, along with performance benchmarking on the modern Layer-2 environments.

A. Traditional Evidence Management System

DEMS are typically based on centralized systems for storing, tracking, and handling forensic materials. These systems utilize audit trails, cryptographic signatures, and controlled-access mechanisms to maintain the chain of custody [5]. Nevertheless, several reports indicate that the centralized repositories remain susceptible to unauthorized access, insider attacks, and system failures [6]. The following weaknesses cast doubt on the long-term reliability of digital evidence, particularly when public-key infrastructures or certificate authorities are compromised [7]. With the increasing volume and sensitivity of digital casework, researchers suggest considering decentralized models that minimize reliance on central trust anchors and offer a greater guarantee against evidence manipulation [8].

B. Blockchain Applications in Digital Forensics

Blockchain technologies have also become a topic of discussion for enhancing the quality, reliability, and transparency of digital forensics. Recent systematic reviews have indicated that blockchain technology enhances evidence integrity by generating unalterable and verifiable records, which significantly reduce the chances of manipulation at any point in the investigative lifecycle [9]. Previous research also emphasizes the practical advantages, which suggest that blockchain-based systems enhance the chain of custody by reducing reliance on centralized repositories, thereby allowing for more transparent and traceable audit trails of digital evidence [10]. More recently, studies have highlighted the critical importance of secure authentication and access control in blockchain forensic systems, demonstrating that forensic log data is protected by fine-grained authorisation controls to prevent unauthorised access [11]. At the same time, larger systematic reviews have found that the transparency and assurance benefits of blockchain do not eliminate scalability as one of the biggest obstacles. The resulting bottleneck has fueled increased attention to Layer-2 improvements, including rollups, Validium systems, and zkEVM architectures, which provide more scalable, evidence-intensive, and forensic workload management environments [12].

C. Smart Contract for Evidence Validation

Smart contracts have become a crucial tool for automating key processes within digital evidence management. They provide reliable operations, such as timestamping, registration, and controlled distribution of forensic records, by ensuring that every transaction is transparent and secure [13]. Recent research also highlights the benefits of utilizing smart contracts in conjunction with advanced cryptographic technologies. For instance, smart contract systems on zkEVM platforms allow privacy-preserving verification while maintaining high levels of auditability, which is crucial in controlled environments [14]. Additionally, experimental applications show that Layer-2 platforms can significantly improve the efficiency of innovative contract-based forensic procedures.

Prototypes on Polygon demonstrate their evidence management systems can authenticate and register digital evidence quickly and cost-effectively, while preserving the legal validity required for forensic procedures [15]. Other case studies support the idea that well-designed smart contract environments can maintain a secure and continuous chain of custody, making evidence verifiable throughout its entire lifecycle [16].

D. Consensus Mechanisms in Blockchain Forensics

The consensus mechanisms define the mechanisms through which blockchain networks verify and store transactions, especially digital evidence records. PoE provides a slim framework of evidence anchoring through an effective timestamping mechanism, which makes it highly applicable to high-volume forensic submissions [17]. To enhance this functionality, PoOW is a system used to attach digital artifacts to authenticated claimants, thereby guaranteeing provenance and legal responsibility [18]. Privacy is further strengthened by mechanisms based on zkEVM, which allows for verification without revealing the underlying information [19]. Comparative studies have shown that these mechanisms exhibit significant differences in throughput, latency, and computational cost variables, which directly determine their suitability for forensic workloads [20]. Modern studies demonstrate that traditional Layer-1 designs are limited in terms of scalability and energy efficiency, underscoring the need for more optimized designs [21]. To overcome these limitations, Layer-2 solutions such as roll-ups, Polygon, and zkEVM-based systems have been proposed to enhance network performance by increasing data availability, reducing operational costs, and facilitating faster transaction processing [22].

E. Benchmarking Frameworks and Performance Metrics

Blockchain system benchmarking requires periodic and controlled evaluation processes to ensure a fair comparison across platforms. Recent studies highlight the importance of standardized testing conditions, including fixed gas settings, consistent performance settings, and repeatable workloads, to accurately measure the throughput, latency, and transaction costs of different blockchain layers [23]. With the widespread adoption of Layer-2 technologies, the latest studies suggest that performance requirements should extend beyond processing speed. An extensive review of roll-up architectures reveals that data availability, proof-generation latency, and off-chain storage policies have a significant impact on overall scalability and system reliability [24]. At the same time, zkEVM design analyses have shown that constraint-level architectural trade-offs, including circuit complexity, proof compression, and prover-hardware requirements, are directly proportional to verification cost and operational efficiency [25]. The existence of empirical benchmarking for modern ZK-roll-ups also confirms that such systems can bring substantial performance improvements, as demonstrated by significant throughput, finality time, and gas cost improvements compared to traditional Layer-1 configurations [26]. The reinforcement of the roll-up architecture through complementary studies of TEE-based solutions also demonstrates that the introduction of heterogeneous trusted-execution environments can further reduce latency and off-chain processing overhead, offering another path towards optimizing blockchain applications with high demand [27].

III. METHODOLOGY

This section describes the methodological systematization that was followed in developing and benchmarking a framework for digital image evidence using blockchain. The framework consisted of three phases: (1) Secure Image Evidence Preparation using IPFS, (2) Consensus Mechanism Implementation on Polygon Amoy TestNet, and (3) Performance Evaluation and Comparative Analysis. All phases incorporated decentralized technologies, cryptography, integrity verification, and automated benchmarking to enable a reproducible, scalable, and tamper-evident digital forensic workflow. The structure and flow of these phases are illustrated in Fig. 1.

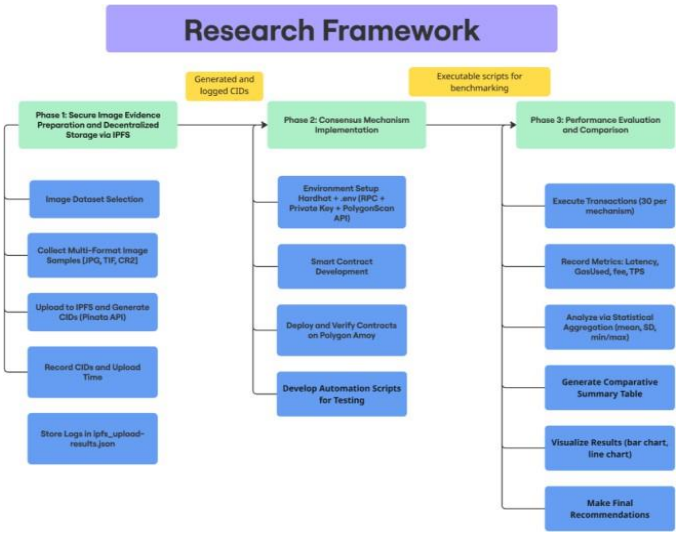

Fig. 1. Research Framework

A. Phase 1: Secure Image Evidence Preparation via IPFS

In the preliminary phase, an exceptionally selected dataset of thirty forensic image samples was extracted from the ShoeCase repository. The three file formats available in the choice, JPG, TIF, and CR2, were balanced to reflect the actual heterogeneity of forensic images in the real world, and to comprehensively evaluate the impact of file format on the quality of content recognition and upload latency.

To ensure decentralized, tamper-evident storage, we utilized the InterPlanetary File System (IPFS) to store each image through the Pinata Application Programming Interface (API), written in a Node.js environment. The transmission routine utilized the fs, form-data, and axios modules to automate the process of submitting every image file. IPFS then generated a unique Content Identifier (CID) for each object, serving as a cryptographic signature of the file's contents. Any alteration to the file would produce a different CID, hence guaranteeing both immutability and verifiability.

The script captured important metadata during the upload process, including the filename, file format, CID, and the duration of the upload. These were stored on a structured JSON file to make further analysis and benchmarking. With the use of CIDs, large files of any image can be stored off-chain without requiring on-chain memory, but can only be referenced on-chain, making them verifiable off-chain.

B. Phase 2: Consensus Mechanism Implementation

The second phase involves setting up three smart-based consensus systems: PoE, PoOW, zkEVM. All models were created on Solidity with the Hardhat framework and deployed on the Polygon Amoy TestNet. These systems were designed to record and authenticate the CIDs created during Phase 1, enabling the on-chain referencing of digital image evidence. Each smart contract was designed to accept CID as input and record it in the chain using a logging mechanism that cannot be modified.

- PoE records the CID with a timestamp, proving the existence of the content at a given moment.
- PoOW associates the CID with the sender's wallet address, establishing a verifiable ownership claim.
- zkEVM simulates a privacy-preserving model by including a hashed placeholder to accommodate zero-knowledge proof extensions.

Deployment was carried out using Hardhat scripts in Node.js. Network credentials, including private keys and API keys, were securely managed through a .env file. The contracts were written in Solidity version 0.8.x, with the optimizer enabled to reduce gas consumption. The deployed configurations included:

- Chain ID: 80002 (Polygon Amoy TestNet)
- RPC Provider: Alchemy
- Deployment Tool: Ethers.js (via Hardhat Runtime Environment)

C. Phase 3: Performance Benchmarking and Evaluation

The third and final step of the methodology analyzes the performance of the three smart contracts deployed on PoE, PoOW, and zkEVM in standardized settings. This benchmarking was conducted to evaluate the relative strengths and weaknesses of each consensus mechanism in validating digital image evidence on-chain. To test 30 transactions per contract, I wrote a test script based on Node.js using Hardhat and Ethers.js, which utilized the generated CIDS that I generated in Phase 1. Per every transaction, they were done using:

- Gas Price: 30 Gwei
- Buffered Gas Limit: 1,500,000 units
- Network: Polygon Amoy TestNet

Transaction details and performance metrics were captured in a structured JSON file for further analysis. Each contract was evaluated using the following metrics:

- Transaction Latency (ms): Time between transaction submission and confirmation.
- GasUsed: Extracted from the transaction receipt to measure computational effort.
- Transaction Fee (ETH): Calculated as GasUsed × GasPrice.
- Gas Efficiency (%): (GasUsed / GasLimit) × 100, indicating resource utilization.
- Transactions Per Second (TPS): 30 / Total Batch Time.
    - Theoretical Max TPS: 1000 / Minimum Latency.
    - TPS Efficiency (%): (Actual TPS / Theoretical Max TPS) × 100

To ensure that performance results were not influenced by extrinsic network variability, benchmarking was conducted in controlled blockchain environments. All the tests were performed when the Polygon Amoy TestNet was not overloaded, and only one stable RPC endpoint (Alchemy) was used to eliminate variability related to switching between providers. The automated script was used in a sequence with fixed delays to avoid accidental mempool spikes, and a constant gas price of 30 Gwei was applied to remove the effect of competition on fees. No parallel deployments or background processes were running during the experiment; thus, the covered latency, throughput, and gas-related measures were only indicative of the inherent behavior of the smart contracts, not temporary network congestion

The collection of performance metrics was analyzed using Python in a Jupyter notebook to ensure statistical rigor and interpretability. The mean, median, and standard deviations were part of the analysis to determine the central tendency and variation of the results, along with minimum and maximum values, to highlight the extremes of performance in the tested mechanisms of consensus. Visualization was key in simplifying the data pattern. Bar charts were used to compare the average values of PoE, PoOW, and zkEVM, and box plots were used to show the distribution and spread of data, allowing outliers to be identified and the level of consistency to be determined. Moreover, the line graphs were created to illustrate changes in performance over time, which implies an understanding of stability and trends over the succession of transactions. All those visual tools helped to see the comparison between the tested blockchains more straightforwardly, particularly in terms of cost efficiency, latency, and scalability.

IV. IMPLEMENTATION DESIGN

This section gives the application of the proposed blockchain-based validation framework in digital image evidence. The implementation is designed into five main modules, namely: System Architecture, IPFS integration, Smart Contract Development, Comparative Contract Analysis, and Deployment on Polygon Amoy TestNet. All the components convert formal designs of the methodology into a working setup that can benchmark performance, as well as verify the verifiability, traceability, and integrity of forensic image records. The architecture utilizes decentralized content-addressed storage via IPFS and supports smart contract logic in three piecemeal consent systems: PoE, PoOW, and zkEVM on the Polygon Amoy TestNet. The development tools Node.js, Hardhat, and MetaMask assist automatic deployment and performance tracking.

III. *System Architecture and Workflow*

The forensic images can be securely and modularly verified through the overall architecture, Fig. 2. This workflow involves uploading forensic image samples to IPFS, with JPG, TIF, and CR2 as three of the supported file types, each assigned a CID every time. Combined with timestamps, every one of these CIDs is logged in a well-structured JSON file. The system will then switch to the smart contract deployment using the Hardhat framework, which is contract-specific to the PoE, PoOW, and zkEVM consensus models. Under every scheme of validation, CIDs are permanently recorded on-chain. All transactions on the smart contract are automatically recorded in JSON files and determined in an environment created on Jupyter Notebook.

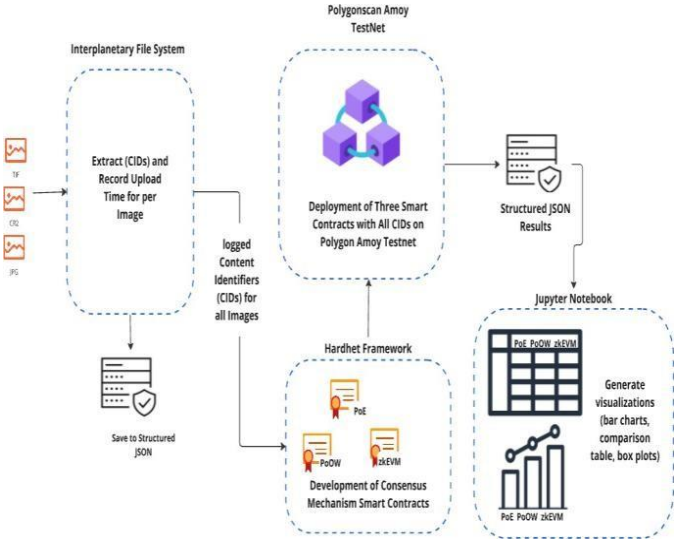

Fig. 2. System Architecture

IV. *IPFS Integration and CID Generation*

The version of IPFS that is implemented serves as the basis for the decentralized evidence store, as it provides tamper-evident and content-addressable unique identifiers for files in the forensic images. The section reports on the preparation of the dataset, the automatic upload mechanism, and structured CID logging, which form the basis of digital traceability of image evidence.

- Dataset Preparation

To recreate real-life forensic scenarios, the implementation utilized publicly accessible Shoe Case data provided by the same image-sharing repository at Iowa State University. The sample was chosen among 10 samples of JPG, 10 samples of TIF, and 10 samples of CR2 formats. These were selected as having opposite features: JPG used lossy compression, TIF lossless compression, and CR2 sensor data (the raw output of the camera sensor), allowing a highly informative assessment of practiced upload behavior and the resulting CID generation.

- Upload Script Implementation

A fully automated Node.js script, developed using modern ES6 module syntax, was implemented to facilitate the upload of forensic image files to IPFS via the Pinata API. The script integrates several packages to streamline its functionality. dotenv securely manages the JWT token for authentication, while fs, path, and fileURLToPath handle file system navigation and path resolution. Additionally, axios and form-data enable the construction and transmission of multipart HTTP POST requests. The script iterates through

image subdirectories categorized by format (JPG, CR2, TIF), uploading each file to IPFS and capturing critical metadata, including the returned CID, upload duration in seconds, file name, and file format. These details are logged into a structured output file named ipfs_upload_results.json, which subsequently involves blockchain-based CID registration and verification.

- CID Logging and Output File

The metadata associated with each successful upload of a forensic image to IPFS is generated with information that includes the original filename (file), image format (format), returned content identifier (IPFS hash), and upload time in seconds (upload time in seconds). Such entries are logged to an indexed JSON file and become a cryptographic record of each image, referencing its individual CID. This log is part of the blockchain validation system, as it ensures that any alteration of an image will result in a different CID, thereby guaranteeing immutability, traceability, and non-repudiation of the original digital evidence throughout the entire system.

V. *Smart Contract Development and Deployment*

The part describes how three smart contracts, PoE, PoOW, and zkEVM, implemented in this project as the central part of the blockchain validation mechanism, have been created, designed, and deployed on-chain. Each of the contracts was coded in Solidity (v0.8.x) and deployed using the Hardhat framework on the Polygon Amoy TestNet to provide an effective real-life simulation of consensus-based digital evidence anchoring.

- Development Environment Setup

Its implementation using a smart contract was done within a carefully crafted, safe, and modular development environment, which supports the creation of efficient compilation, testing, and deployment processes. A few essential tools were utilized: Hardhat was employed to simulate a local blockchain and deploy contracts; Node.js was used to automate deployment tasks by writing our own ES6 scripts. The ethers.js framework was used to work with on-chain operations (i.e., running transactions), and the dotenv package ensured that crucial sensitive information (such as API keys or personal addresses) was managed securely. The PolygonScan API was also established to automate the verification of source code, making it more transparent and traceable. The setup of the environment consisted of the hardhat.config.js file, namely the Amoy TestNet chain ID (80002), the RPC address, and the deployer private key, respectively, loaded safely in the hardhat.config.js file using the .env file so that the credentials could not be exposed in the code.

- Contract Design and Logic

PoE is lightweight and timestamp verification-oriented. It receives a content hash (usually an IPFS CID), verifies whether it is already recorded, and, if not, records the current block timestamp. As illustrated in Fig. 3, the timestamp query getTimestamp() can later be used by the user to obtain an easily verifiable proof of file existence; however, it is unnecessary for tracking ownership or file metadata.

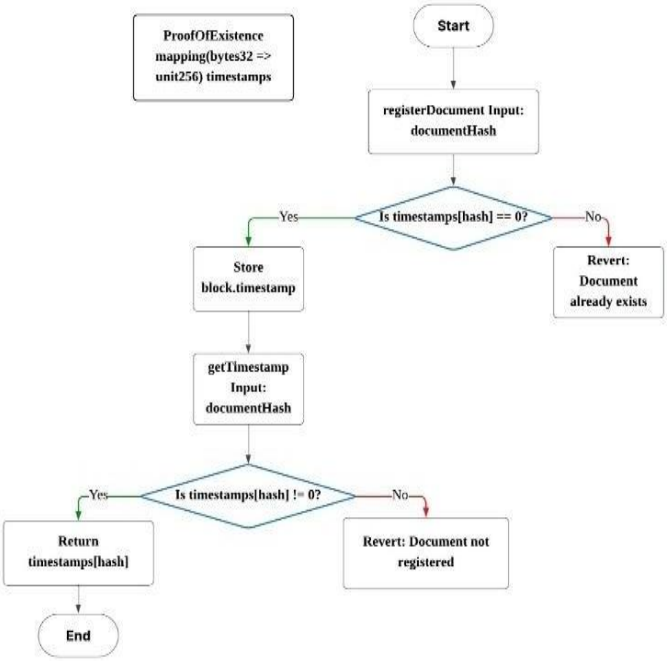

Fig. 3. PoE Flowchart Diagram

As an extension of PoE, PoOW enables the storage of a CID, the owner's Ethereum address, and a timestamp, providing cryptographic proof of ownership. AS shown in Fig. 4, this agreement would be most appropriate where copyrights or the use of law is of importance. In the case of an existing CID, it will not be overwritten, and it will maintain an immutable record.

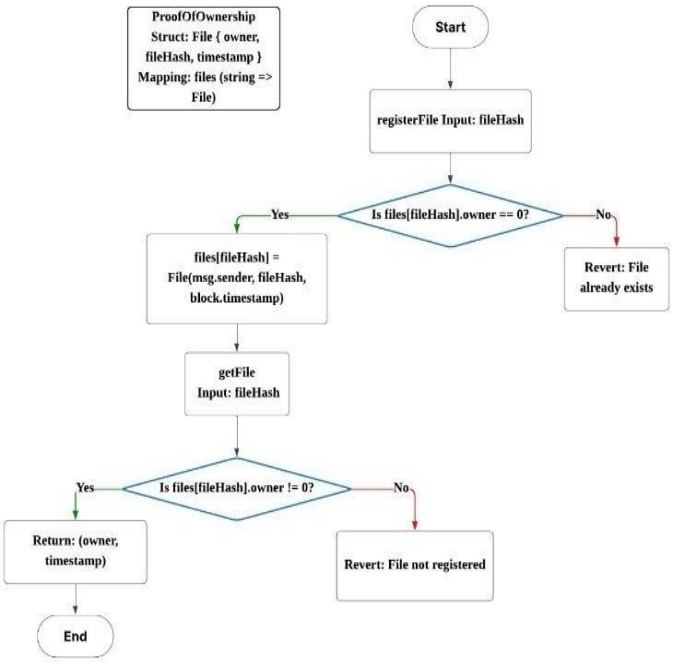

Fig. 4. PoOW Flowchart Diagram

The zkEVM contract allows privacy zk-SNARKs-based CID verification through the insertion of ECDSA signature validation. Once a user has filed a proofHash, an imageHash, and a signature, the contract first checks for uniqueness and then verifies the signature using the cryptographic functions of OpenZeppelin. As shown in Fig. 5, the design is expected to be used in the future together with a zk-SNARK or zk-STARK framework.

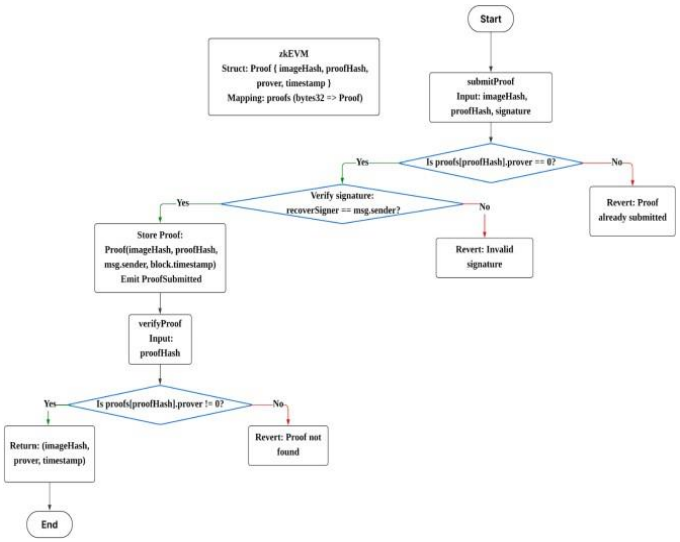

Fig. 5. zkEVM Flowchart Diagram

- Comparative Features of the Smart Contract

To compare the trade-offs of the different mechanisms for consensus, a side-by-side comparison was conducted based on input requirements, stored data, logic, and the intended use case. The most gas-efficient and simplest is PoE. PoOW brings in the ownership context. zkEVM provides the most sophisticated security layer, featuring signature-based proof handling.

VI. *Deployment to Polygon Amoy TestNet*

The smart contracts PoE, PoOW, and zkEVM were deployed using a systematic process within the Hardhat development framework, aiming for the Polygon Amoy TestNet. Every contract was written using Solidity (version 0.8. x), optimised to fortify the size of the bytecode and optimise its gas consumption. Custom Node.js scripts were used to automate the deployment pipeline, providing repeatable deployments and consistent backend results by communicating with the Hardhat Runtime Environment (HRE). Necessary credentials, such as private keys and RPC endpoints, were securely stored in environment variables through the .env file and were never hard-coded in the source files. After the collection, the contracts were executed in order, and those ready to execute were called by each transaction and transmitted to ensure their finality on the network. Once deployed, the contracts were thoroughly checked on PolygonScan via the platform's official API, which enabled them to be publicly accessible, trackable, and viewable using the explorer. The used contract addresses were tracked and utilized as endpoints during the benchmark phase. This systematic deployment is then used to provide a secure, transparent, and reproducible base for evaluating the performance of all three consensus mechanisms.

V. RESULTS

This section presents a detailed analysis of the experimental evaluation of the three implemented smart contracts, PoE, PoOW, and zkEVM, which were deployed on the Polygon Amoy TestNet. The performance assessment was conducted using thirty transactions per contract under controlled conditions. Key metrics, including transaction latency, gas usage, transaction fees, throughput, and efficiency, were examined to compare their suitability for blockchain-based image evidence validation.

A. Smart Contract Performance Summary

All contracts were launched through the same execution environment with a gas price of 30 GWei and a gas limit of 150000 units. Table I summarizes the entire benchmarking results by recording the average latency, batch processing time, gas consumption, transaction cost, and throughput efficiency of each consensus model.

TABLE I. PERFORMANCE COMPARISON OF SMART CONTRACTS

Metric	PoE	PoOW	zkEVM
avgLatency (ms)	**3730**	**5180**	**4500**
totalBatchTime(ms)	**170,232**	**213,754**	**193,608**
avgGasUsed	**44,040**	**134,833**	**118,795**
gasEfficiency	**29%**	**89%**	**79%**
avgTransactionFee (ETH)	**0.001321**	**0.004045**	**0.003564**
actualTPS	**0.176**	**0.140**	**0.155**
theoreticalMaxTPS	**0.578**	**0.577**	**0.580**
tpsEfficiency (%)	**30.4%**	**24.3%**	**26.7%**

B. Latency Evaluation and Batch Processing Time

Transaction latency, which is a significant aspect in real-time forensic systems, was considered first. As shown in Fig. 6, PoE had the least average delay (3730 ms), zkEVM (4500 ms), and PoOW had the highest delay (5180 ms). The latency indicates the computational complexity of each model, with the least logic in PoE allowing for a faster implementation. In contrast, in PoOW, the execution is slow due to its cryptographic operations.

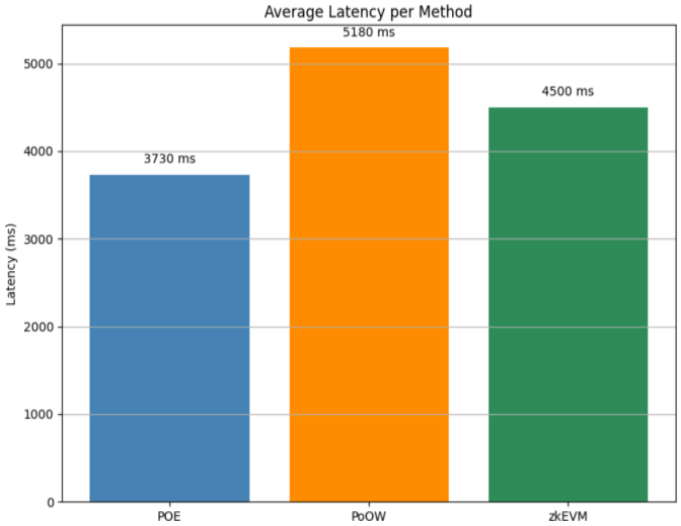

Fig. 6. Average Transaction Latency Across Smart Contract Models

The overall batch duration also followed the latency patterns. PoE processed its batch in 170,232 ms, which is quicker than zkEVM (193,608 ms) and PoOW (213,754 ms), demonstrating that PoE is more efficient in time-sensitive applications.

B. Gas Usage and Efficiency Evaluation

The use of gas provides insight into the computational impact of any smart contract. The average gas consumption and efficiency of every model are presented in Fig. 7. PoE had the lowest gas consumption (44,040 units) because the logic

behind the hash registration in this chain is simple. PoOW, on the other hand, incurred the highest gas (134,833 units) consumption due to its cryptographic check. zkEVM used 118,795 units, which was a balance between verification complexity and resource control.

In terms of efficiency, PoE was the least efficient with an average gas efficiency (29%), second was zkEVM (79%), and lastly PoOW (89%). Nevertheless, very efficient does not necessarily mean cost-effective PoE; even though it is less efficient, it is the most gas-economical as it carries a light computational load.

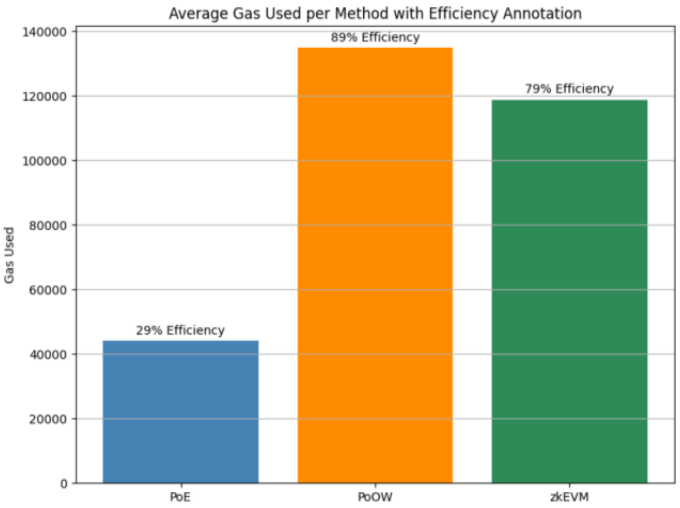

Fig. 7. Average Transaction Gas Efficiency Across Smart Contracts Models

C. Transaction Fee Comparison

The transaction fee for each model was calculated using the fixed gas price. Fig. 8 illustrates that PoE had the lowest cost (0.001321 ETH), confirming its economic advantage. PoOW incurred the highest cost (0.004045 ETH) due to intensive computation, while zkEVM offered a middle ground at 0.003564 ETH, delivering cryptographic privacy without excessive overhead.

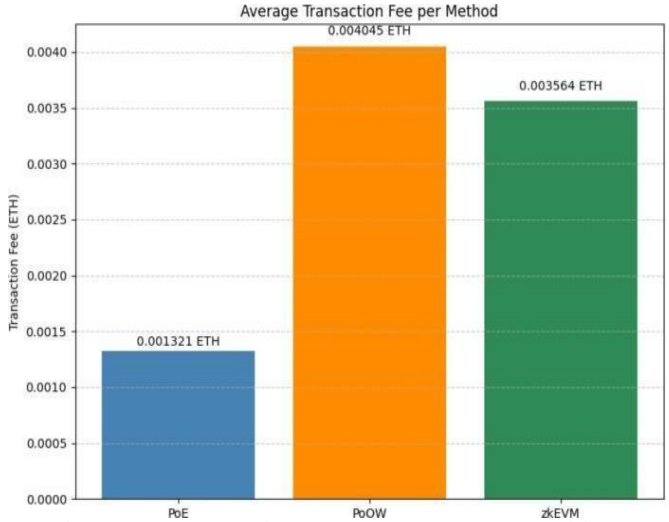

Fig. 8. Average Transaction Fee Across Smart Contract Models

D. Throughput and TPS Efficiency Evaluation

The efficiency of TPS was computed using the actual TPS and the theoretical TPS. As shown in Fig. 9, the highest throughput (0.176 TPS) and efficiency (30.4%) were achieved with PoE, demonstrating its suitability for high-volume workloads. zkEVM came second.0.155 TPS and 26.7 per cent efficiency with PoOW having as low as 0.140 TPS and 24.3 per cent performance because of processing overhead.

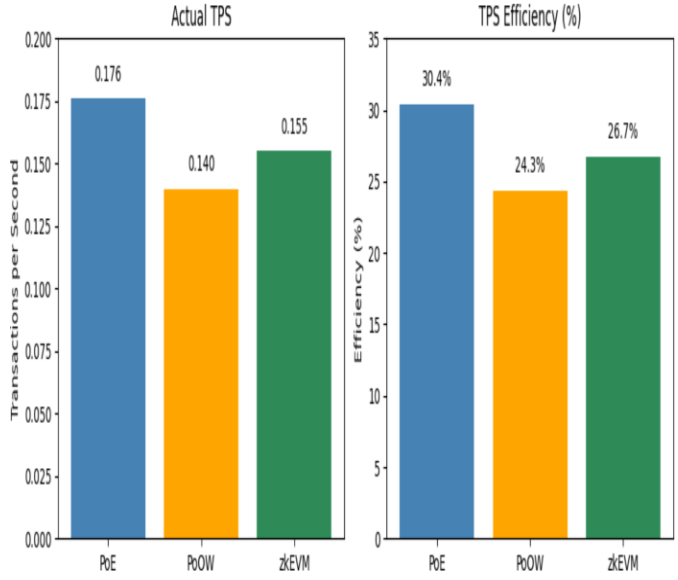

Fig. 9. Comparative Analysis of TPS and Efficiency Across Contract Model

E. Recommendation and Use Case Insights

Based on empirical results, the following recommendations are made:

- PoE is best for time-sensitive and cost-conscious scenarios, such as rapid digital evidence registration.
- PoOW is suited for legal or ownership-sensitive use cases requiring strong traceability and verification.
- zkEVM is ideal for confidential data validation where privacy and security are critical but balanced with reasonable performance.

Scalability-wise, PoE is preferred for mass deployments, while zkEVM is more appropriate for secure, enterprise-level applications.

VI. CONCLUSION

The current paper introduced a thorough analysis of a blockchain-based solution to the problem of authenticating digital image evidence based on three architecture options available to smart contracts: PoE, PoW, and Zero-Knowledge zkEVM. These smart contracts were rolled out

to the Polygon Amoy TestNet with calibrated operating conditions that utilized an identical set of transactions (30 per contract), each based on image-based Content Identifiers (CID)s leveraged on IPFS. The main objective was to evaluate the level to which each of the models is suitable for secure, scalable, and efficient validation under decentralized forensic systems.

A clear performance difference emerged in the experimental results. PoE recorded the lowest average latency (3730 ms), transaction cost (0.001321 ETH), and highest throughput (0.176 TPS), which affirms its usefulness in low-latency and cost-sensitive applications. PoOW was the most gas-consuming, but also the most gas-efficient (89%) protocol due to its own firm logic ownership verification, making it suitable for use in security-sensitive applications. The zkEVM struck a balance between performance and privacy, as it incorporated an advanced cryptographic component while maintaining reasonable latency (4,500 MS) and transaction fees (0.003564 ETH). The results of the research confirm the effectiveness of automated logging, statistical benchmarking, and organized visualization in making contract/application choices contingent on a given operational requirement, thereby contributing to versatile and compliant blockchain-based validation systems.

VII. LIMITATIONS AND FUTURE WORK

This study also has its limitations, despite the positive outcomes. The experimental tests were limited to 30 transactions per contract and, therefore, may not be representative of full-scale operational behavior in the presence of heavy or real-world traffic. The testing was performed only on the Polygon Amoy TestNet; therefore, the results may vary if the network is exposed to congestion, miner prioritization, or gas dynamics on the Ethereum mainnet or other production networks. Moreover, the zkEVM implementation used against libraries that would be developed would impact cryptographic stability. The research was also more centered around performance indicators, such as latency, gas consumption, and TPS, with no extensive work being done on security resilience, privacy assurance, or legal scenario scalability.

The next step in research should be scalability testing, where the number of transactions performed is vastly expanded, and submissions are piled on top of one another to see how the system behaves under stress. A deployment on production networks, such as Ethereum's mainnet, Arbitrum, or Optimism, would provide more realistic details on the variability of gas and the reliability of transactions. Comparative analysis of the benefits of advanced zk-protocols, such as zk-SNARKs and zkzk-STARKs, is advocated to reduce further proof size, verification period and, privacy tradtrade-offsture work would need to conduct studies further on the real-time intelligent contract monitoring based on interactive dashboards under legal practitioners and give extensions to the framework through the study of cross-chain interoperability that would provide the framework applicability and greater access and validity of forensic verification within the distributed ecosystems.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Faculty of Computing, Universiti Teknologi Malaysia (UTM) for providing the resources and supportive research environment necessary to complete this study. We also wish to extend our appreciation to the reviewers for their insightful feedback and constructive suggestions, which significantly helped improve the quality and clarity of this paper.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Albeshri, A. (2021). An image hashing-based authentication and secure group communication scheme for IoT-enabled MANETs. *Future Internet*, *13*(7), 166. https://doi.org/10.3390/fi13070166.

[2] Loffi, L., Camillo, G. L., De Souza, C. A., Westphall, C. M., & Westphall, C. B. (2025). Management of the chain of custody of digital evidence using blockchain and self-sovereign identities: A systematic review. https://www.researchgate.net/publication/390742953.

[3] Borse, Y., Patole, D., Chawhan, G., Kukreja, G., Parekh, H., & Jain, R. (2021). Advantages of blockchain in digital forensic evidence management. In *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST 2021)*. SSRN. https://doi.org/10.2139/ssrn.3866889.

[4] Custers, B., & Stevens, L. (2021). The use of data as evidence in Dutch criminal courts. *European Journal of Crime, Criminal Law and Criminal Justice*, *29*(1), 25–46. https://doi.org/10.1163/15718174-bja10015.

[5] Custers, B. H. M., & Stevens, L. (2024). Data as evidence in criminal courts: Comparing legal frameworks and actual practices. In S. Gless & H. Whalen-Bridge (Eds.), *Human-robot interaction in law and its narratives* (pp. 221–251). Cambridge University Press. https://doi.org/10.1017/9781009431453.014.

[6] Szabo, J., Bernard, C., & Philip, L. (2024). Legal implications and challenges of blockchain technology and smart contracts. *Computer Life*, *12*(2), 6–10. https://doi.org/10.54097/ztn2w848.

[7] S. G., Narendhran, V., D. K., A. M., & K. K. (2025). Blockchain-based evidence tracking system for forensic integrity and secure chain of custody. In *2025 1st International Conference on Radio Frequency Communication and Networks (RFCoN)* (pp. 1–6). IEEE. https://doi.org/10.1109/RFCoN62306.2025.11085167.

[8] Negi, S., Kumar, A., Pandey, S., Yamsani, N., Singh, R., & Balyan, R. (2023). The preservation of digital evidences through blockchain technology. In *2023 IEEE World*

Conference on Applied Intelligence and Computing (AIC) (pp. 954–958). IEEE. https://doi.org/10.1109/AIC57670.2023.10263968.

[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. https://doi.org/10.3390/electronics13173568

[10] Borse, Y. (2021). Advantages of blockchain in digital forensic evidence management. In *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST 2021)*. SSRN. https://doi.org/10.2139/ssrn.3866889.

[11] Kim, S., Lee, H., & Park, J. (2023). User authentication and access control to blockchain-based forensic log data. *EURASIP Journal on Information Security*, *2023*(7). https://doi.org/10.1186/s13635-023-00142-3.

[12] Andrews, K., Ngo, L. B., & Amiruzzaman, M. (2025). A detailed comparative analysis of blockchain consensus mechanisms (Version 2). *arXiv*. https://arxiv.org/abs/2511.15730.

[13] Santamaría, P., Tobarra, L., Pastor-Vargas, R., & Robles-Gómez, A. (2023). Smart contracts for managing the chain-of-custody of digital evidence: A practical case study. *Smart Cities*, *6*(2), 770–777. https://doi.org/10.3390/smartcities6020039.

[14] Fekete, D. L., & Kiss, A. (2023). Toward building smart contract-based higher education systems using zero-knowledge Ethereum Virtual Machine. *Electronics*, *12*(3), 664. https://doi.org/10.3390/electronics12030664.

[15] Rana, S. K., Rana, A. K., Rana, S. K., Sharma, V., Lilhore, U. K., & Khalaf, O. I. (2023). Decentralized model to protect digital evidence via smart contracts using Layer 2 Polygon blockchain. *IEEE Access*, *11*, 83289–83300. https://doi.org/10.1109/ACCESS.2023.3302771.

[16] Cable, N. R. (2025). *Standardizing blockchain layer 2 benchmarking* (Master's thesis, Lehigh University). Lehigh Preserve. https://preserve.lehigh.edu/.

[17] Kumar, R., Sharma, N., & Gupta, P. (2025). A systematic literature review of blockchain technology and energy efficiency based on consensus mechanisms, architectural innovations and sustainable solutions. *Journal of Intelligent & Fuzzy Systems*, *39*(4), 5571–5590. https://doi.org/10.1007/s44257-025-00041-6.

[18] Zhu, D., Tong, X., Wang, Z., & Zhang, M. (2022). A novel lightweight block encryption algorithm based on combined chaotic system. *Journal of Information Security and Applications*, *69*, 103289. https://doi.org/10.1016/j.jisa.2022.103289.

[19] Ghaffar, R., Ali, S., & Khan, M. (2024). A survey on data availability in Layer-2 blockchain rollups: Open challenges and future improvements. *IEEE Access*. https://www.researchgate.net/publication/383532885_A_Survey_on_Data_Availability_in_Layer_2_Blockchain_Rollups_Open_Challenges_and_Future_Improvements.

[20] Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., & Rimba, P. (2021). A comprehensive review of blockchain consensus mechanisms. *ResearchGate*. https://www.researchgate.net/publication/350031088_A_Comprehensive_Review_of_Blockchain_Consensus_Mechanisms.

[21] Bin Saif, M., Migliorini, S., & Spoto, F. (2024). A survey on data availability in Layer 2 blockchain rollups: Open challenges and future improvements. *Future Internet*, *16*(9), 315. https://doi.org/10.3390/fi16090315.

[22] Hassanzadeh-Nazarabadi, Y., & Taheri-Boshrooyeh, S. (2025). Constraint-level design of zkEVMs: Architectures, trade-offs, and evolution. *arXiv*. https://arxiv.org/abs/2510.05376.

[23] Chaliasos, S., Reif, I., Torralba-Agell, A., Ernstberger, J., Kattis, A., & Livshits, B. (2024). Analyzing and benchmarking ZK-rollups. *ACM*. https://doi.org/10.4230/LIPIcs.AFT.2024.6.

[24] Wen, X., Feng, Q., Lyu, H., Niu, J., Zhang, Y., & Feng, C. (2025). TeeRollup: Efficient rollup design using heterogeneous TEE (Version 2). *arXiv*. https://doi.org/10.48550/arXiv.2409.14647.

[25] Giménez, C., Ahmed, L., Benhaim, F., Coronado, S., & Perales, P. (2025). Analyzing performance bottlenecks in zero-knowledge rollups. *arXiv*. https://arxiv.org/abs/2503.22709.

[26] Torralba-Agell, A., Chaliasos, S., Reif, I., & Vasquez, D. (2025). Constraint-level design of zkEVMs: Architectures, trade-offs and performance. *arXiv*. https://arxiv.org/abs/2510.05376.

[27] Cable, N. R. (2025). *Standardizing blockchain layer 2 benchmarking* (Master's thesis, Lehigh University). Lehigh Preserve. https://preserve.lehigh.edu/.