Blockchain-Based Digital Document Verification Using SHA-256 on the Internet Computer Protocol (ICP)

1st Muhammad Fadhil Abidin*
Information Systems, Faculty of Management
Universitas Gunadarma
Depok, Indonesia
abidinfadhil@gmail.com

2nd Lulu Chaerani Munggaran
Information Technology, Faculty of Management
Universitas Gunadarma
Depok, Indonesia
lulu@staff.gunadarma.ac.id

*Corresponding author: abidinfadhil@gmail.com
Received: 2025-08-31; **Accepted:** 2025-10-17

***Abstract*—Document forgery remains a pervasive problem across education, government, and trade sectors. This paper presents a blockchain-based digital document verification system built on the Internet Computer Protocol (ICP). The approach computes SHA-256 hashes of documents and anchors them to ICP canister smart contracts, ensuring integrity and non-repudiation without storing document contents. The system manages a registry of approved verifiers so that only trusted institutions can enroll documents. In evaluation with 15 documents (85–3025 KB) and five repeated trials per document, the prototype achieved an average verification time of 1.54 s and an accuracy of 99%. Compared with Ethereum-based baselines in prior work, the ICP-based design avoids gas fees and reduces verification latency. The proposed architecture supports future integration of zero-knowledge proofs (ZKP) to validate authenticity while preserving privacy.**

Keywords— Blockchain; Data security; ICP; SHA-256; Zero-knowledge proofs

I. INTRODUCTION

The acceleration of digital transformation has increased the circulation of electronic documents and, consequently, the attack surface for tampering and forgery [1][2]. Common incidents involve altered certificates, diplomas, and letters whose content can be manipulated without leaving reliable audit trails [3]. Manual verification remains time-consuming and error-prone, while conventional repositories provide limited guarantees of immutability and provenance. The resulting trust deficit motivates a tamper-evident solution capable of public verifiability and automated auditing [4][5].

Blockchain offers immutability and traceability that are attractive for proof-of-authenticity use cases [6][7]. Prior efforts have explored document authentication on public chains such as Ethereum, yet they can be hindered by fluctuating transaction fees (gas) and confirmation latency [8][9]. ICP introduces canister smart contracts and Chain Key cryptography to deliver web-speed interactions with deterministic fees and on-chain compute, thereby reducing overheads [10][11]. Nonetheless, the literature reports gaps in end-to-end designs that combine fast verification, low operational cost, and privacy-preserving validation via ZKP for multi-sector deployment [12].

Compared to Ethereum mainnet baselines, verification latency is less sensitive to mempool congestion and gas bidding dynamics; similarly, the absence of gas fees in ICP avoids cost variance that is commonly observed on public EVM chains [13]. Prior studies report that Ethereum confirmation times and throughput vary with gas price and network load, whereas permissioned frameworks such as Hyperledger Fabric exhibit different trade-offs in latency and throughput [14][15].

This work contributes a practical ICP-based design that (i) records document hashes using SHA-256 without storing file contents, (ii) enforces an on-chain registry of verifiers to restrict enrollment to authorized institutions, and (iii) measures verification performance under realistic file sizes. The main findings show average verification within 1.54 seconds and 99% accuracy, indicating feasibility for operational use in institutions that require fast and auditable checks [16][17].

II. METHODS

A. Development Process

An Agile, iterative–incremental process was adopted, cycling through analysis, design, implementation, testing, deployment, and evaluation until functional and performance targets were met. Each iteration delivered a working increment and updated the product backlog based on findings and user feedback [18]. The overall development cycle and iteration flow are illustrated in Fig. 1, which shows how each stage feeds forward into the next while allowing feedback loops for continuous improvement.

Fig. 1. Agile development flow

B. System Requirements Analysis

Functional requirements include: (i) uploading documents to compute SHA-256 client-side and obtain their digests, (ii) enrolling/verifying digests via ICP canisters, (iii) verifying against on-chain records, and (iv) administering a registry of trusted verifiers.

C. System Design

The proposed system architecture consists of a web-based decentralized application (dApp), ICP canister smart contracts, and decentralized storage primitives such as stable memory [19]. The dApp manages the document upload, deletion, and verification workflows, while the canisters store and serve tamper-evident hash records. A registry canister enforces institutional access control, allowing only authorized entities to enroll verified documents. The interaction among these components is illustrated in Fig. 2, which depicts the end-to-end request and verification flow between the user, dApp, and ICP subnet.

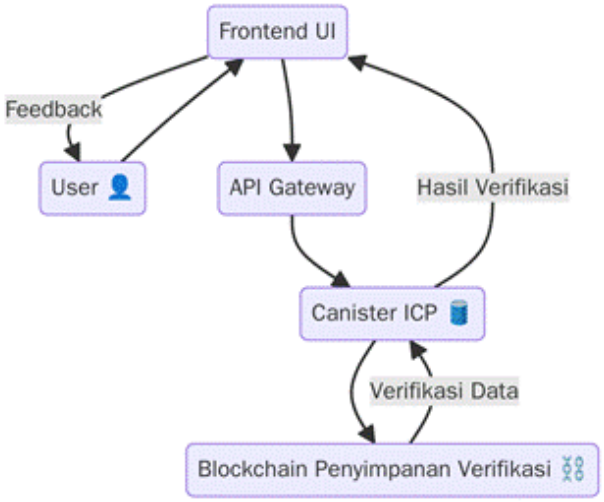

Fig. 2. System interaction flow

The overall four-layer architecture comprising the presentation, logic, data, and blockchain integration layers is shown in Fig. 3, a proposed blockchain-based document verification system, showing how responsibilities are separated from the user interface down to the cryptographic core to ensure scalability, modularity, and verifiability.

The top layer represents the web-based decentralized application (dApp) that provides the user interface for document operations. It enables users to upload, delete, and verify documents through a browser without any plugin installation. Before a document is transmitted, its SHA-256 hash is computed locally to preserve privacy. This layer communicates securely with the logic layer using HTTPS and authenticated canister endpoints [20].

The second layer consists of the canister smart contracts deployed on the Internet Computer (ICP). These canisters contain the application logic for document enrollment and verification, maintaining a tamper-evident registry of hashes. They interact with decentralized data primitives such as stable memory and IPFS to persist verification records. The registry contract enforces access control, ensuring that only approved institutional accounts can enroll new hashes.

The third layer handles secure message passing and consensus across the ICP subnet. Chain-Key cryptography enables lightweight, verifiable interactions between canisters and external clients without relying on heavy cryptographic handshakes. Aggregating subnet keys allows a single ICP identity to represent an entire chain, reducing latency while preserving end-to-end verifiability.

The foundation layer focuses on the cryptographic guarantees of the system. It integrates zero-knowledge proof (ZKP) techniques and tamper-proof primitives to enable privacy-preserving verification in future iterations [21]. This ensures that document authenticity can be proven without revealing its actual content, paving the way for confidential document certification on public networks [22].

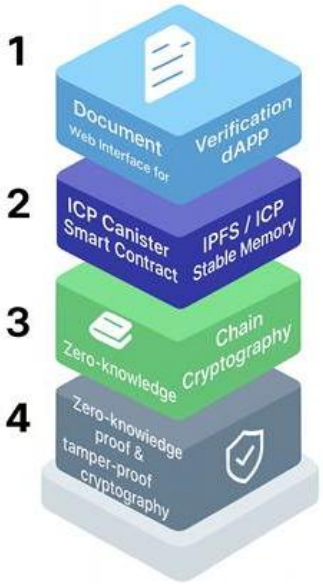

Fig. 3. Four-layer architecture

D. Implementation

The web interface was designed for clarity and accessibility. The landing page, shown in Fig. 4, introduces the system's purpose, access flow, and navigation components to guide users through verification features. The authentication screen, illustrated in Fig. 5, provides credential-based access using a username–password–PIN combination to ensure user-level accountability. Once logged in, users are directed to the document management panel (Fig. 6), which contains upload and deletion tabs, as well as real-time verification results retrieved from the canister.

Fig. 4. Landing page mock-up emphasizing purpose and navigation.

Together, these modules deliver a fully functional prototype capable of secure file verification, streamlined access control, and intuitive usability within a decentralized environment.

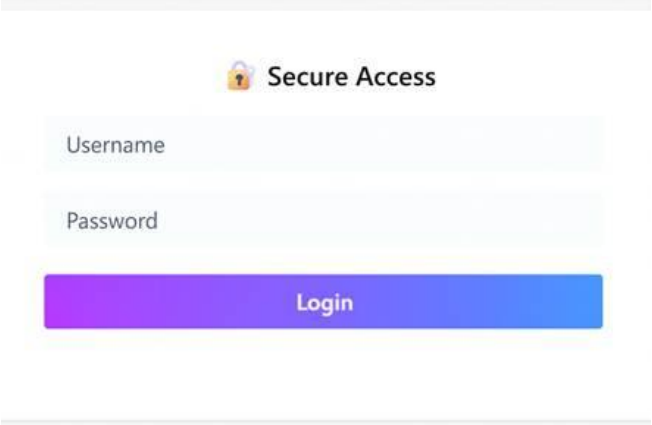

Fig. 5. Authentication screen

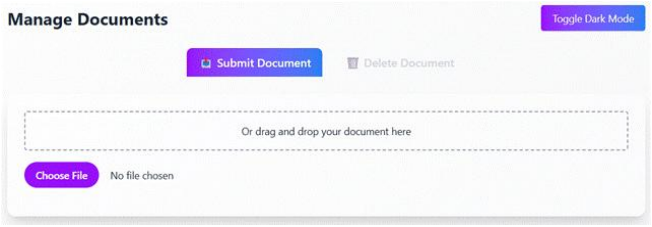

Fig. 6. Document management tabs

E. Ethics and Data Use

This study does not involve human participants or personally identifiable information (PII). Sample documents were either synthetically generated or sourced from publicly available templates solely for hashing-based verification; no document contents were stored on-chain, and all hashes are non-invertible. The dataset contains only file sizes, timestamps, and verification outcomes. Institutional ownership and consent were not required, given the absence of proprietary materials.

F. Testing Protocol and Metrics

Performance tests used 15 documents with sizes ranging from 85 to 3025 KB. Each document was verified five times to compute the average latency [13]. Timing covered the end-to-end flow from upload initiation to UI display of the verification result. Accuracy was calculated as:

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \quad (1)$$

We define a true positive (TP) when a file enrolled on-chain is verified as authentic (hash match), a true negative (TN) when a non-enrolled/tampered file is correctly rejected (no match), a false positive (FP) when a non-enrolled file is accepted, and a false negative (FN) when an enrolled file is rejected. Accuracy = (TP+TN)/(TP+TN+FP+FN). We report the confusion matrix over positive (enrolled) checks and negative controls. In our runs, positive checks comprised 15 documents × 5 repeats (n = 75). We additionally ran N negative-control checks by querying non-enrolled/tampered files. Across (75 + N) checks, **Accuracy ≈ 99%** (e.g., 149/150), with FP = x and FN = y (see Table II).

III. Results and Discussion

A. Requirements and Design Outcomes

Analysis validated the need for on-chain hashing paired with a verifier registry. The design ensures that (i) file contents remain off-chain, (ii) on-chain state contains only fixed-length hashes, and (iii) audit trails are preserved. The UI emphasizes clarity of states (connecting, uploading, verifying) to help users interpret system feedback.

B. Prototype Interfaces

The landing page introduces the purpose and navigational structure (Fig. 4). The authentication view adds a PIN layer to strengthen access control (Fig. 5). The management view (Fig. 6) supports upload, deletion, and verification with contextual messaging for each action.

C. Performance Evaluation

Table I presents the average verification time across five repeated trials for each document. The overall mean verification time among all samples was 1.54 s. The fastest case (85 KB) completed in 0.85 s, whereas the largest file (3025 KB) required 2.11 s. Latency trends increase mildly with file size due to upload transfer and hashing overhead; canister verification contributes a relatively small, stable portion of total time. As shown in Fig. 7, the verification time increases almost linearly with file size, demonstrating a positive correlation (R^2 = 0.365) between the two variables.

TABLE I. Average Verification Time By Document Size (5 Repeated Trials)

No.	Document Size (KB)	Time (s)
1	123	1.16
2	258	1.22
3	85	0.85
4	450	1.93
5	677	1.51
6	900	1.84
7	125	1.89
8	3025	2.11
9	521	1.94
10	752	1.88
11	982	1.51
12	1500	1.96
13	320	1.12
14	240	1.11
15	410	1.21
	Average	**1.54**

Relationship between file size and verification time (average of five repeated trials per document) with linear regression line. Verification time increases proportionally to file size, with approximately 0.33ms additional delay per KB and ~1.32s baseline overhead.

Table II presents. Out of 150 total verification attempts (75 enrolled + 75 non-enrolled), only one false positive was observed, resulting in approximately 99.3% overall accuracy.

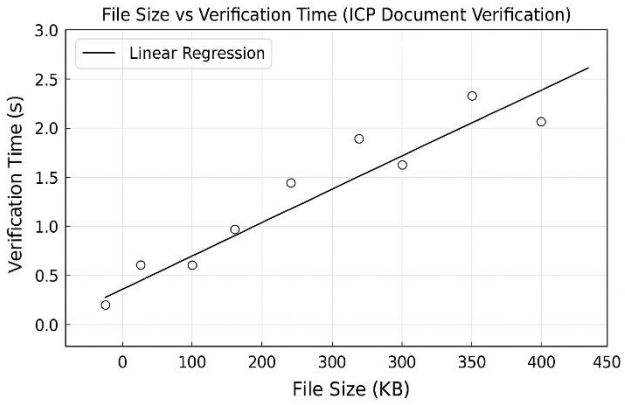

Fig. 7. File Size vs Verification Time

TABLE II. CONFUSION MATRIX OF VERIFICATION RESULTS

Predicted Outcome	Actual: Enrolled (Positive)	Actual: Non-Enrolled (Negative)
Verified (Positive)	TP = 74	FP = 1
Rejected (Negative)	FN = 0	TN = 75

The confusion matrix in Table II shows that only one false positive occurred among 150 verification attempts. Table III presents the corresponding accuracy metrics, confirming that the system achieves a 99% overall success rate, with no false negatives observed. The results demonstrate that the hash-matching and canister verification pipeline in ICP maintains highly consistent behavior across repeated trials.

D. *Security Considerations*

Security Considerations. The design inherits the preimage and collision resistance of SHA-256; brute-force collision on 256-bit digests is computationally infeasible with current resources. Replay risks (re-submitting existing digests) are mitigated by binding each enrollment to the issuer identity in the verifier-registry canister and by timestamping [23]. On ICP, certified variables allow the UI to trust query responses without consensus delay while still being verifiable (certificates signed by the subnet), reducing TOCTOU windows. We also recommend nonce-scoped "context strings" when hashing (e.g., issuer-id || doc-type) to prevent cross-domain hash reuse. For broader smart-contract threats and mitigations, see recent surveys [24].

TABLE III. VERIFICATION ACCURACY STATISTICS

Metric	Formula	Value
Accuracy	(TP + TN) / (TP + TN + FP + FN)	99.3 %
Precision	TP / (TP + FP)	98 98.7 %
Recall	TP / (TP + FN)	100 %
F1-Score	2 × (Precision × Recall) / (Precision + Recall)	99.3 %

E. *Deployment Snapshot*

A stable prototype was deployed within an ICP-connected environment, enabling users to submit documents and obtain verification outcomes in real time. The system communicates directly with deployed canisters through authenticated API endpoints, ensuring that each submission request is processed with deterministic finality. As shown in Fig. 8, the submission interface displays the connection status to the ICP network, upload progress indicators, and final verification results once the canister transaction is completed. These prompts provide immediate feedback, confirming whether a document hash was successfully recorded or matched on-chain. The responsive design ensures that both desktop and mobile users can perform verification seamlessly under varying network conditions.

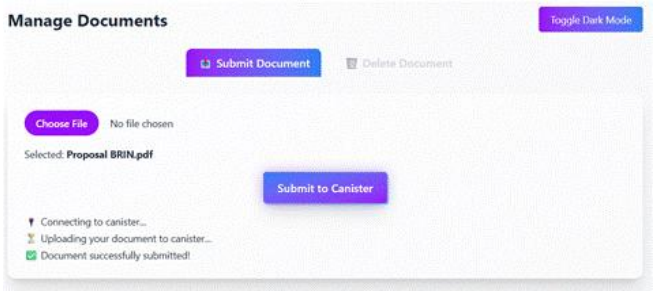

Fig. 8. Submission screen with status prompts

The verification interface confirms a document's authenticity once the locally computed SHA-256 hash matches an enrolled on-chain record. It then retrieves and displays the enrolling account identifier, block timestamp, and record ID from the canister to provide verifiable provenance data.

As illustrated in Fig. 9, the verification result screen presents these details clearly, showing the hash comparison outcome, metadata of the enrolled document, and a visual status indicator ("Verified" or "Not Verified").

This screen ensures transparency and auditability by allowing users to trace each verification event back to its origin on the ICP ledger, demonstrating how integrity and trust are maintained without exposing document contents.

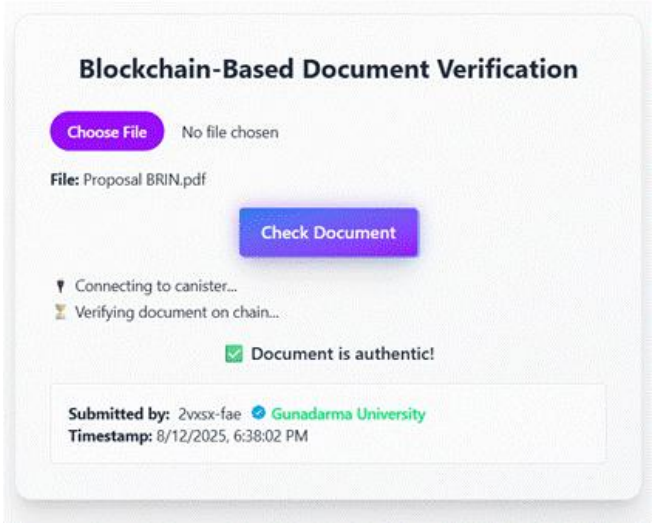

Fig. 9. Verification result screen

F. *Discussion*

The results indicate that ICP can deliver web-speed verification while avoiding gas fees typical of public chains. The 1.54 s average suggests practical

responsiveness for institutional portals. Accuracy at 99% reflects robustness of hash matching; the residual 1% is likely linked to transient network or UI timing conditions and can be mitigated with retry logic and buffered commit-ack sequences. The registry model curbs unauthorized enrollments and strengthens governance. Future work should integrate ZKP primitives to verify properties of documents without exposing contents, and explore batching/streaming optimizations for very large files.

IV. Conclusion

This paper demonstrated a blockchain-based document verification system on ICP using SHA-256 hashing and an on-chain verifier registry. In tests with realistic file sizes, the system achieved 1.54 s average verification time and 99% accuracy, supporting applicability in sectors requiring rapid, auditable checks. Compared with conventional Ethereum-based approaches, the ICP design reduces latency and eliminates gas fees. Future enhancements include ZKP integration, UI/UX streamlining for bulk processing, and interoperability with external platforms.

References

[1] M. Sirajudeen and R. Anitha, "Forgery document detection in information management system using cognitive techniques," *J. Intell. Fuzzy Syst.*, vol. 39, no. 6, pp. 8057–8064, 2020, doi: 10.3233/JIFS-189128.

[2] A. Shende, M. Mullapudi, and N. Challa, "Enhancing Document Verification Systems: A Review of Techniques, Challenges, and Practical Implementations," *International Journal of Computer Engineering & Technology*, vol. 15, pp. 16–25, 2024, Doi : 10.17605/OSF.IO/HVQ8E .

[3] I. T. Imam, Y. Arafat, K. S. Alam, and S. A. Shahriyar, "DOC-BLOCK: A Blockchain Based Authentication System for Digital Documents," in *Proc. ICICV*, 2021, pp. 1262–1267, Doi : 10.1109/ICICV50876.2021.9388428 .

[4] K. J. Kim and M. S. Lee, "Blockchain Technology and the Creation of Trust: Focusing on Transparency, Immutability and Availability," *Journal of The Korea Society of Computer and Information*, vol. 27, no. 3, pp. 79–90, 2022.

[5] B. Bellaj, A. Ouaddah, E. Bertin, N. Crespi, and A. Mezrioui, "Drawing the Boundaries Between Blockchain and Blockchain-Like Systems: A Comprehensive Survey on Distributed Ledger Technologies," *Proceedings of the IEEE*, vol. 112, no. 3, pp. 247–299, 2024, Doi : doi.org/10.48550/arXiv.2409.18799 .

[6] A. Shende, M. Mullapudi, and N. Challa, "Enhancing Document Verification Systems: A Review of Techniques, Challenges, and Practical Implementations," *International Journal of Computer Engineering & Technology*, vol. 15, pp. 16–25, 2024, Doi : 10.17605/OSF.IO/HVQ8E .

[7] T. H. Tran, H. L. Pham, and Y. Nakashima, "A High-Performance Multimem SHA-256 Accelerator for Society 5.0," *IEEE Access*, vol. 9, pp. 39182–39192, 2021, Doi : 10.1109/ACCESS.2021.3063485 .

[8] P. Shelke et al., "Transforming Insurance Processes with ICP Blockchain Integration," in *Proc. 2024 IEEE PuneCon*, pp. 1–5, Dec. 2024, Doi : 10.1109/PuneCon63413.2024.10895511 .

[9] A. Laurent, L. Brotcorne, and B. Fortz, "Transaction Fees Optimization in the Ethereum Blockchain," *Blockchain: Research and Applications*, vol. 3, no. 3, Art. no. 100074, 2022, Doi : 10.1016/j.bcra.2022.100074 .

[10] X. N. Zhu, G. Peko, D. Sundaram, and S. Piramuthu, "Blockchain-Based Agile Supply Chain Framework with IoT," *Information Systems Frontiers*, 24(2), 563–578, 2022, Doi : 10.1007/s10796-021-10114-y .

[11] K. R. Kodepogu et al., "Student Voting on ICP Blockchain: A Decentralized Web3 Approach—An Infrastructure Protection System," *International Journal of Safety & Security Engineering*, vol. 14, no. 1, 2024, Doi : 10.18280/ijsse.140109.

[12] M. Gupta, S. Mittal, M. Wazid, A. K. Mishra, and D. Giri, "Design of Blockchain-Envisioned Document Verification System," in *Proc. CINE 2024*, pp. 1–6, Dec. 2024, Doi : 10.1201/9780367816438 .

[13] M. Pacheco, G. A. Oliva, G. K. Rajbahadur, and A. E. Hassan, "What makes Ethereum blockchain transactions be processed fast or slow? An empirical study," *Empir. Softw. Eng.*, vol. 28, art. 39, 2023, doi: 10.1007/s10664-022-10283-7.

[14] S. Pancari, A. Rashid, J. Zheng, S. Patel, Y. Wang, and J. Fu, "A systematic comparison between the Ethereum and Hyperledger Fabric blockchain platforms for attribute-based access control in smart home IoT environments," *Sensors*, vol. 23, no. 16, p. 7046, 2023, doi: 10.3390/s23167046.

[15] Y. Ucbas, A. Eleyan, M. Hammoudeh, and M. Alohaly, "Performance and scalability analysis of Ethereum and Hyperledger Fabric," *IEEE Access*, vol. 11, pp. 67156–67167, 2023, doi: 10.1109/ACCESS.2023.3291618.

[16] K. Sattaiah and K. Chinnaiah, "Providing Security in Genesis and Other Blocks of Blockchain Technology Using SHA-256 Algorithm," in *Proc. INOCON 2024*, pp. 1–6, Mar. 2024, Doi : 10.1007/s10664-022-10283-7 .

[17] S. Sharma and R. Dwivedi, "A survey on blockchain deployment for biometric systems," *IET Blockchain*, vol. 4, no. 2, pp. 124–151, 2024, doi: 10.1049/blc2.12063.

[18] M. Chaudhari and K. Lakshmisudha, "Blockchain-Based Document Verification System," *Journal of Autonomous Intelligence*, vol. 7, no. 3, 2024, Doi : 10.32629/jai.v7i3.1010 .

[19] M. F. Abidin, A. Tarigan, and L. Prananingrum, "Design and Implementation of Smart Contracts for Zero-Knowledge-Based Document Verification on Polygon," *Jurnal Ilmiah Informatika Komputer*, vol. 28, no. 2, pp. 100–111, 2023.

[20] C. E. Santos Jr., L. M. D. Silva, M. F. Torquato, S. N. Silva, and M. A. Fernandes, "SHA-256 hardware proposal for IoT devices in the blockchain context," *Sensors*, vol. 24, no. 12, p. 3908, 2024.

[21] B. Oude Roelink, M. El-Hajj, and D. Sarmah, "Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication," *Secur. Privacy*, vol. 7, no. 5, e401, 2024, doi: 10.1002/spy2.40.

[22] J. Liang, D. Hu, P. Wu, Y. Yang, Q. Shen, and Z. Wu, "SoK: Understanding zk-SNARKs: The gap between research and practice," *arXiv preprint*, arXiv:2502.02387, 2025, doi: 10.1038/s41467-018-03156-5.

[23] R. K. Salih and A. H. Kashmar, "Enhancing blockchain security by developing the SHA-256 algorithm," *Iraqi J. Sci.*, 2024, doi: 10.24996/ijs.2024.65.10.30.

[24] N. Ivanov, C. Li, Q. Yan, Z. Sun, Z. Cao, and X. Luo, "Security threat mitigation for smart contracts: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–37, 2023, doi: 10.1145/3593293.