Data Sharing and Verification for Certificates of Origin Based on Blockchain and Zero-Knowledge Proof

Huan Wang[1]*

[1]Yunnan University of Finance and Economics, Information Center, South Campus, Yunnan University of Finance and Economics, Wuhua District, Kunming City, Yunnan Province, China

Keywords
Zk-snark; Blockchain; Data Sharing; Decentralized Verification

Correspondence Email:
202302110719@stu.ynufe.edu.cn

Abstract
Blockchain technology, as a representative paradigm of distributed ledgers, has been widely applied in cross-border trade and finance. However, in cross-border trade, issues such as privacy leakage, forgery, and data misuse during document data sharing hinder the receiving parties from verifying the authenticity of shared data. To address these problems, this paper proposes a decentralized and trustworthy data verification scheme based on the PLONK protocol in zk-SNARKs. Leveraging PLONK's universal trusted setup, the proposed system performs off-chain computation and on-chain verification. Sensitive trade data are processed with the Poseidon hash to generate commitments, which are then recorded on the blockchain for integrity verification. Arithmetic circuits are constructed to ensure that the committed data satisfy pre-defined validity constraints. Meanwhile, trusted institutions conduct off-chain audits to realize a minimal disclosure mechanism, ensuring privacy-preserving verification. Based on an analysis of verification requirements in certificate-of-origin scenarios, the system architecture and arithmetic circuits are designed and implemented to achieve secure, privacy-preserving, and verifiable data sharing in cross-border trade.

1. Introduction

Under the deepening trend of globalization, the complexity and scale of cross-border trade have increased significantly, involving diverse participants such as customs authorities, financial institutions, logistics providers, and multi-national regulators. As a result, ensuring the authenticity and integrity of trade documents has become a major challenge. Existing verification processes remain highly dependent on centralized institutions, leading to issues such as information asymmetry, data monopolies, and slow, manual authentication procedures that hinder digital trade efficiency. Blockchain technology offers a new path for building decentralized trust due to its immutability and traceability. However, simple "hash-on-chain" methods cannot verify the correctness of document content, while uploading plaintext raises privacy concerns. Zero-Knowledge Proofs (ZKPs) provide a solution by enabling verification without revealing sensitive information. In particular, universal zk-SNARK schemes such as PLONK allow trade participants to generate proofs off-chain while keeping only commitments and verification logic on-chain, achieving minimal disclosure and efficient validation. Current research on trusted document sharing mainly follows two approaches: (1) blockchain combined with standardized trade data models, and (2) blockchain integrated

with privacy-preserving computation, including ZKP, MPC, and TEE. Proof systems such as PLONK have gained attention due to their universal CRS, efficiency, and engineering practicality, and have been adopted in prototypes for credential verification and selective disclosure. Despite these advances, challenges remain in cross-border standardization, cross-chain interoperability, circuit universality, and the coordination of on-1 chain/off-chain computation. To address these limitations, this paper proposes a decentralized document verification mechanism that integrates universal ZKP circuits, institutional signature validation, and interpretable access control models. The system aims to provide practical engineering feasibility while maintaining privacy, auditability, and compliance with complex cross-border regulatory requirements.

1.1 Development of Zero-Knowledge Proofs

The concept of Zero-Knowledge Proofs (ZKPs) was introduced by Goldwasser et al. as a method for proving the correctness of a statement without revealing sensitive information. Blum et al. later proposed Non-Interactive Zero-Knowledge (NIZK) proofs based on a Common Reference String (CRS), enabling single-message verification and applications in signatures and CCA-secure encryption. With the development of zk-SNARKs, Groth16 became a milestone by reducing proofs to three group elements and achieving fast verification through pairing-friendly elliptic curves. Its main limitation, however, is the requirement for circuit-specific trusted setup. Subsequent work focused on universal and updatable CRS. Sonic introduced polynomial-commitment-based proofs with reusable setup, and PLONK further improved efficiency through a refined commitment scheme and constraint system. PLONK achieves significantly faster proving time and has become a mainstream zk-SNARK due to its practicality [1-8].

Beyond SNARKs, zk-STARKs provide transparency and post-quantum security by eliminating trusted setup using hash-based assumptions, but their large proof sizes and high verification cost limit on-chain usage. Bulletproofs offer logarithmic proof sizes without trusted setup and are widely used for range proofs, though their comparatively slower verification makes them less suitable for high-frequency blockchain applications [8-10].

1.2 KZG Polynomial Commitment

The KZG Polynomial Commitment (KZG-PC) is an efficient polynomial commitment scheme based on the Knowledge of Exponent Assumption in bilinear pairings. Its core idea is to utilize structured group elements generated during a trusted setup phase to enable verifiable commitments to polynomials and their evaluation values [15]. The first step in constructing a commitment is the trusted setup, during which random parameters are generated to build the Structured Reference String (SRS) as follows:

(G0, G1, G2, ..., G(d-1), H0, H1) = (G, chi*G, chi^2*G, ..., chi^(d-1)*G, H, chi*H)

Here, the secret scalar chi is referred to as the toxic waste produced during the trusted setup. It must be securely destroyed after setup by a trusted third party, ensuring that chi remains completely hidden within the base vectors. Even with knowledge of chi*G, it is infeasible to derive chi. G1 and G2 are generators of the two pairing groups, with a bilinear mapping e: G1 x G2 -> GT that satisfies the bilinearity property e(aG1, bG2) = e(bG1, aG2) = e(abG1, G2) = e(G1, aG2)^b = e(G1, G2)^ab. For example, let a polynomial over a finite field be f(x) = a0 + a1*x + a2*x^2 + ... + a(n-1)*x^(n-1). Then the commitment is defined as C = f(chi)*G = a0*G + a1*chi*G + ... + a(n-1)*chi^(n-1)*G. To prove the correctness of the polynomial's evaluation at a random point zeta, the prover constructs an opening proof. Since (x - zeta) must be a factor of the polynomial f(x) - y, there exists a quotient polynomial q(x) such that f(x) - y = (x - zeta) * q(x). The prover computes the commitment to q(x) and provides it as the opening proof. The verifier then checks the following bilinear equation to verify correctness: e((f(x) + zeta * q(x) - y), 1) = e(C, x).

2. Research Methods

2.1 Requirement Analysis

With the intensification of the digital economy, information technology has already penetrated every corner of various industries [10], and data has become an important asset of the modern economy. At the same time,

numerous challenges concerning data security and privacy have emerged. Incidents of declining corporate credibility, economic losses, and personal privacy breaches caused by data leakage have become increasingly common. At present, user information is mainly stored on the servers of third-party service providers. Both service providers and malicious attackers who obtain data through illegal means can analyze and extract users' personalized characteristics by collecting relevant data, thereby conducting precise advertising, providing targeted personalized services, and so on. This brings users a series of problems in daily life, such as massive spam messages and marketing phone calls.

To address the above issues, this paper analyzes practical application scenarios and, by integrating modular zk-SNARK-based NIZK techniques with blockchain, transforms the document verification problem in international trade into a non-interactive proof statement. In this way, verifiers are able to confirm the correctness and compliance of trade documents without accessing the underlying data. The overall process of transforming real-world document verification requirements into a NIZK-verifiable formulation is illustrated in Figure 1 at the appropriate position.

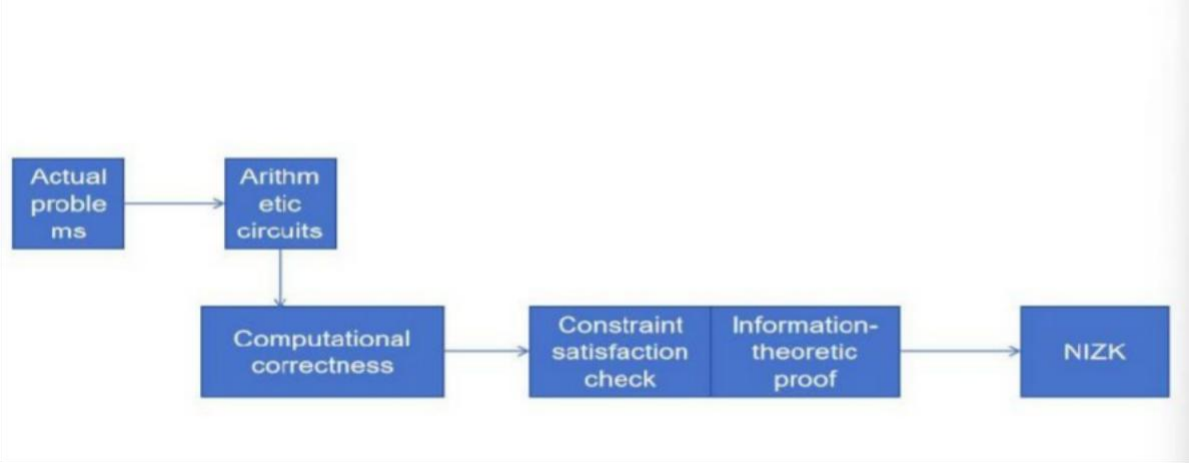

Fig. 1 Practical problem conversion NIZK process

2.2 System Model

The system proposed in this paper establishes a privacy-preserving verification framework for cross-border trade documents based on Zero-Knowledge Proof (ZKP) and Decentralized Identifier (DID) technologies. The framework adopts the universal zk-SNARK protocol PLONK as the core verification mechanism, enabling privacy protection and data verifiability in a multi-party collaborative environment. As illustrated in Figure 2, the system model consists of five entity roles: Trusted Party, User, Institution, Blockchain, and Verifier. The Trusted Party generates the trusted setup parameters of the PLONK protocol and securely destroys all randomness after initialization. The User conducts constraint checks on its private data, produces commitments, and generates zero-knowledge proofs. The Institution verifies the authenticity of the committed data within its business scope and issues digital signatures. The Blockchain records consensus parameters, manages credential indexing, and performs decentralized proof verification. The Verifier validates the credentials and associated zero-knowledge proofs to assess the correctness and compliance of the data.

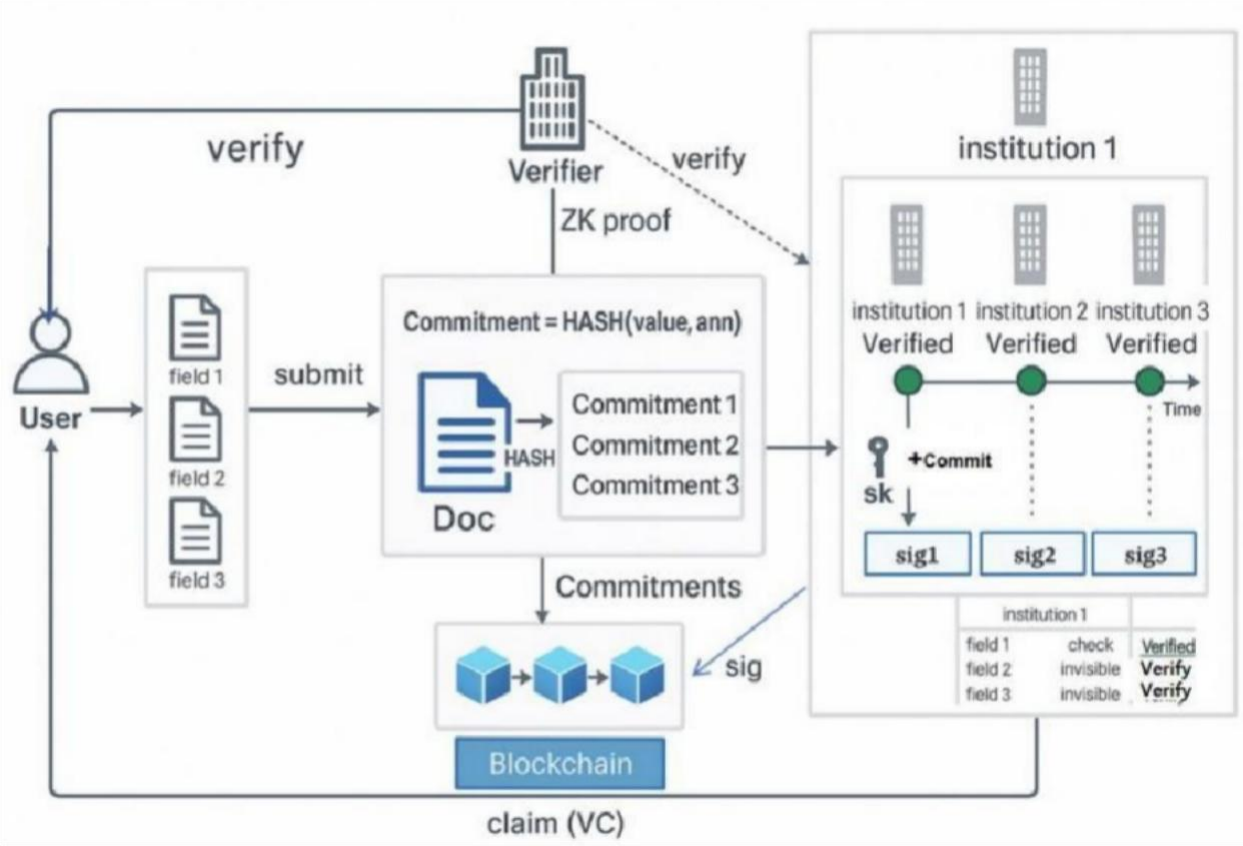

Fig. 2 System Model

2.3 Commitment Generation

In the context of cross-border trade document processing, the data structure typically includes multidimensional information such as enterprise identifiers, transaction amounts, country-of-origin ratios, and product codes. Some fields, such as document numbers and exporting countries, need to be publicly disclosed for regulatory inspection, while other fields - such as transportation details and company-origin material information - are privacy-sensitive and should only be verified by authorized institutions. To achieve differentiated disclosure control, the system adopts a Hash Commitment Scheme to generate commitments for each piece of metadata containing private information. The pseudocode for the commitment generation process is illustrated in the corresponding figure. Let the extracted user privacy input vector be denoted as: V
= {v1, v2, v3, ..., vn}. For each element in V, the user generates a random factor r, and the output commitment C is computed as: C = {c1, c2, c3, ..., cn}, ci = H_poseidon(vi, r1, r2). The commitment scheme satisfies two essential properties [14]: Hiding and Binding. Hiding ensures that external entities cannot infer the original value from the commitment, while Binding guarantees that once a commitment is generated, the corresponding original value cannot be altered.

2.4 Circuit Construction

This section describes the design and construction of data verification circuits. To verify trade document data under cryptographic constraints, the system transforms logical conditions into Arithmetic Circuits. The pseudocode for circuit construction is shown in the corresponding figure. The circuit consists of three components: a Commitment Verification Subcircuit, which verifies the consistency between the commitment C and the private input V; a Constraint Verification Subcircuit, which checks trade compliance rules such as Regional Value Content (RVC); and an Access Control Subcircuit, which enforces verification authorization

based on the Linear Secret Sharing Scheme (LSSS). Let the commitment verification subcircuit be denoted as f(V, r), where V represents private input values and r are two random factors (nonces) uniformly selected from a finite field. The subcircuit should satisfy the consistency condition between the commitment C and the private input V.

In trade document processing, many certificates include customized constraint conditions. For instance, in the preferential trade certificate FORM E under the China-ASEAN Free Trade Agreement, the rule requires that the Regional Value Content (RVC) within China-ASEAN territories be greater than or equal to 40%. The arithmetic representation of this constraint can be expressed as: RVC = (FOB - VNM) / FOB * 100% >= 40%, where FOB refers to Free On Board (the export product's ex-factory price), and VNM refers to the Value of Non-Originating Materials. Considering that division operations are computationally expensive in zero-knowledge circuits, this constraint can be equivalently transformed into: (FOB - VNM) * 100 >= FOB * 40. Assume that in the user's private data array Values[], Values[1] and Values[3] represent the value of materials originating from China-ASEAN, while Values[5] corresponds to the value of non-originating materials. Then, the second part of the circuit (the constraint verification subcircuit) can be expressed as enforcing arithmetic constraints that ensure the above relationship holds, thereby verifying whether the origin data provided by the user satisfies the preferential trade condition RVC >= 40%.

In the access-control component, the system defines an access policy using a Linear Secret Sharing Scheme (LSSS), represented by an access matrix M and a role vector rho. During proof generation, the circuit computes M * s and verifies whether the result equals a predefined secret s. When this equality holds, the role vector is considered to satisfy the access policy, indicating that the prover possesses the institutional roles required for this document. This mechanism ensures that the prover can generate a valid proof only under the correct role configuration, achieving policy enforcement with minimal disclosure and strong privacy protection.

2.5 Proof Generation and Verification Framework

Data Verification Process. The entire system consists of three main modules: the Client, the Proof System, and the Blockchain. In the client stage, the user first performs local data input and commitment generation. Using the Poseidon hash function, the user mixes private data with random factors (nonces) to compute the Commitment, generating an input vector Values and a commitment vector Commits. These commitments serve both as the public inputs for subsequent proof circuits and as the foundation for privacy protection. In the proof system stage, the client invokes the PLONK circuit (WASM) together with the SNARKJS toolchain to generate a Zero-Knowledge Proof (ZKP). The circuit logic corresponds to the constraint design described previously: the Commitment Verification Subcircuit is used to verify the correctness of the hash commitment; the Constraint Verification Subcircuit validates whether the private inputs satisfy preferential trade conditions (for example, RVC >= 40%); and the Access Control Subcircuit embeds an LSSS (Linear Secret Sharing Scheme) structure to control verifier access policies.

The system generates the zero-knowledge proof (Proof) off-chain using the witness, and subsequently serializes the proof along with the public inputs to derive an on-chain data structure (callData). In the blockchain stage, the generated callData is submitted to an on-chain smart contract (EVM/Smart Contract). The system follows an off-chain computation, on-chain verification architecture: the off-chain module is responsible for constructing the zero-knowledge proof, while the on-chain smart contract executes the verification function Verify(C, pi) = (false, true). If the verification result returns true, the corresponding trade document data is deemed to satisfy the established constraint conditions.

2.6 Trust Model Based on DID

To ensure the authenticity of identities and the verifiability of credentials, the system introduces a Decentralized Identifier (DID) framework to construct the foundational trust layer. During the registration phase, the user locally generates a DID along with a corresponding DID Document, which contains metadata

such as the public key, controller address, and service endpoints. The document is encrypted and uploaded to the distributed storage system IPFS, obtaining a unique Content Identifier (CID). Subsequently, the system registers the binding relationship between the CID and the DID on-chain through the DID Registry smart contract, thereby establishing the mapping DID -> CID -> Document.

During the verification phase, the verifier retrieves the DID included in the credential and queries the on-chain DID Registry to obtain the corresponding CID. The verifier then accesses IPFS to fetch the DID Document. The public key contained in the document is used to verify the institution's digital signature and the authenticity of the credential, achieving decentralized verification of both identity and data. This hybrid architecture combines on-chain registration with off-chain storage: the on-chain component ensures immutability of indexing, while the off-chain component provides scalability and privacy of data.

Fig. 4 DID Registration and Resolution Workflow

After completing identity registration and resolution based on DID, the system further implements the processes of commitment verification and credential issuance. The main objective of this phase is to ensure that the user's submitted commitment information can be securely and verifiably issued and utilized in the form of a zero-knowledge proof after being verified and signed by multiple trusted institutions. Specifically, the user first generates commitment data on the client side and submits it to the corresponding Institutions. These institutions, according to their respective responsibilities, review the submitted metadata and verify the correctness of the computations. Upon successful verification, they generate digital signatures for the validated commitments. Certain institutions with final authorization qualifications append a Final Authority Seal to the credential, forming a multi-signature credential structure.

After the credential is generated, the system integrates the commitment results and the associated signatures into a Verifiable Credential (VC), which includes both the Zero-Knowledge Proof (ZKP) and institutional signature information. The credential is then processed through the DID/IPFS Registry module for signature verification and storage indexing, ensuring the authenticity and traceability of the signature sources. Subsequently, a ZKP Verifier smart contract is deployed on the blockchain to verify the correctness of the zero-knowledge proof, ensuring that the committed data satisfies the required constraints without revealing the underlying values. Meanwhile, the verifier obtains the institution's public key from the corresponding DID Document to validate the accompanying signatures, confirming that the commitment is authenticated by authorized entities. This design enhances the system's verifiability and trust scalability while maintaining robust privacy protection.

3. Experimental Results and Discussion

This experiment was conducted in a local Hardhat blockchain testing environment, simulating the system's operation under real deployment conditions. The experiment focused on evaluating the performance of three core stages: (1) circuit compilation and deployment; (2) zero-knowledge proof (ZKP) generation; and (3) proof verification within the smart contract. By comparing computational time and gas consumption at different stages, the overall computational efficiency and privacy-preserving performance of the system were analyzed. During the circuit construction phase, the system compiled and generated 4,791 PLONK constraints, including 2,052 linear constraints and 1,649 nonlinear constraints, with 6 public inputs and 18 private inputs. After compilation, the circuit produced the WebAssembly (WASM) file and the verification key (zkey), which were deployed using the snarkjs toolchain. The entire compilation and deployment process took approximately 7.2 seconds. Although this step is only required once during initial deployment, its complexity directly affects the runtime efficiency of subsequent proof generation and verification.

Figure 6 (left) illustrates the distribution of gas consumption for major on-chain operations. The ZKP verification phase consumed the highest amount of gas, reaching 385,457 gas, accounting for more than half of the total cost. This stage involves elliptic curve pairing computations and polynomial verifications, making it the most computation-intensive part of the system. The Commit Post phase consumed 311,018 gas, mainly due to data writing and state initialization operations in the contract. The Commit Verify phase consumed

only 45,863 gas, which mainly involves lightweight operations such as hash verification and state queries, resulting in relatively low resource usage. Overall, by separating heavy cryptographic computations from lightweight on-chain storage operations, the system achieved effective resource utilization and hierarchical efficiency.

Figure 6 (right) presents the time distribution across different operational stages. Experimental results show that proof generation took approximately 4.8 seconds, primarily reflecting the computational complexity of circuit processing and random number generation on the client side. On-chain verification took about 10.9 seconds, including transaction submission, block confirmation, and the execution of smart contract verification logic. The total end-to-end process - from input submission to the return of the verification result - was completed in under 20 seconds. This result demonstrates that although proof generation introduces certain computational overhead, the verification latency remains within an acceptable range, making privacy-preserving real-time verification feasible.

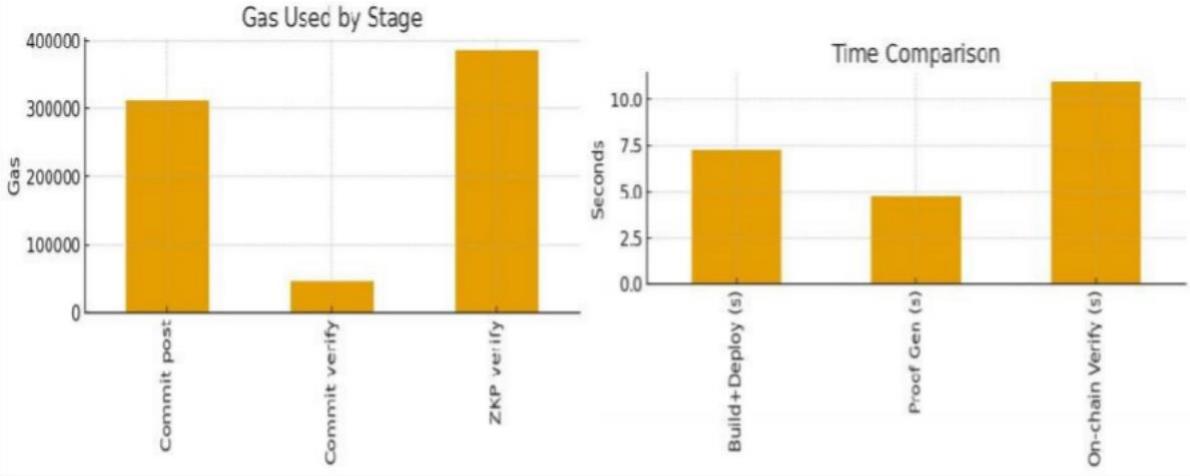

Fig. 3 System Performance Data

From a performance distribution perspective, the system achieved reasonable results in both gas consumption and execution time. The ZKP verification stage remains the primary performance bottleneck; however, its resource cost is still within an acceptable range on Ethereum-compatible chains. Compared with traditional plaintext verification mechanisms, the proposed approach maintains good on-chain verification efficiency while ensuring strong privacy protection. In addition, the generated zero-knowledge proof size is approximately 800 bytes, resulting in low communication overhead. The proof generation time scales almost linearly with the number of circuit constraints, indicating that further improvements in scalability can be achieved through constraint structure optimization and the adoption of recursive proofs. Overall, the experimental results validate the feasibility and performance advantages of the PLONK-based zk-SNARK framework in decentralized identity (DID) and credential verification scenarios. The system's layered design - combining off-chain computation with on-chain verification - enables highly reliable verification without exposing private data, demonstrating excellent deployability and scalability potential.

4. Conclusions

This paper constructs and validates a decentralized identity verification framework that integrates zero-knowledge proofs (ZKP) with blockchain technology, realizing a complete workflow from DID registration and commitment submission to proof generation and verification. The proposed system enables on-chain verifiability of user identities and credentials without exposing original data, effectively balancing privacy preservation and auditability. Experimental results demonstrate that the system completes circuit construction and deployment within 7.2 seconds, generates proofs in 4.8 seconds, and finalizes verification

10.9 seconds. The gas consumption across the three stages - Commit Post, Commit Verify, and ZKP Verification - was 311,018, 45,863, and 385,457 gas, respectively, with the verification phase incurring the highest but still acceptable cost. The proof size is approximately 800 bytes, and the verification cost remains below 400,000 gas, indicating the feasibility of deployment on Ethereum-compatible blockchains.

These findings confirm that the proposed PLONK-based zk-SNARK system possesses strong practical value in decentralized credential verification scenarios. Its architecture decouples computation from verification, offloading intensive computations to the off-chain environment while performing lightweight verification through smart contracts on-chain, thereby significantly reducing on-chain resource consumption. Meanwhile, the compact proof size and stable verification performance provide a solid technical foundation for large-scale data sharing and privacy-compliant inter-organizational verification. Future work can focus on two key optimization directions: (1) reducing circuit complexity through constraint structure optimization and constant folding to shorten proof generation time; and (2) exploring batch verification and recursive proof mechanisms to further lower on-chain verification gas costs. In summary, this research demonstrates the application potential of zero-knowledge proofs in privacy-preserving and blockchain-based trusted computation. It provides both experimental validation and theoretical support for the development of distributed trust systems that comply with privacy and regulatory requirements.

5. References

[1] Facebook. Libra White Paper. https://libra.org/en-US/white-paper/

[2] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems[M]//Providing sound foundations for cryptography: On the work of Shafi Goldwasser and Silvio Micali. 2019: 203-225.

[3] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems[C]//Conference on the theory and application of cryptographic techniques. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986: 186-194.

[4] Reyzin L, Yakoubov S. Efficient asynchronous accumulators for distributed PKI[C]//Security and Cryptography for Networks: 10th International Conference, SCN 2016, Amalfi, Italy, August 31-September 2, 2016, Proceedings 10. Springer International Publishing, 2016: 292-309.

[5] Al-Bassam M. SCPKI: A smart contract-based PKI and identity system[C]//Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts. 2017.

[6] Nikolaenko V, Ragsdale S, Bonneau J, et al. Powers-of-Tau to the People: Decentralizing Setup Ceremonies[J]. Cryptology ePrint Archive, 2022: 279-295.

[7] Ben-Or M, Goldreich O, Goldwasser S, et al. Everything provable is provable in zero-knowledge[C]//Advances in Cryptology-CRYPTO'88: Proceedings 8. Springer New York, 1990.

[8] Partala J, Nguyen T H, Pirttikangas S. Non-interactive zero-knowledge for blockchain: A survey[J]. IEEE Access, 2020, 8: 227945-227961.

[9] Goldwasser S, Micali S, and Rackoff C. The knowledge complexity of interactive proof-systems[C]//Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85). 1985: 291-304.

[10] Arora S, Safra S. Probabilistic checking of proofs: A new characterization of NP[J]. Journal of the ACM (JACM), 1998, 45(1): 70-122.

[11] Micali S. Computationally sound proofs[J]. SIAM Journal on Computing, 2000, 30(4): 1253-1298.

[12] Bos J W, Costello C, Longa P, et al. Selecting elliptic curves for cryptography: an efficiency and security analysis[J]. Journal of Cryptographic Engineering, 2016, 6: 259-286.

[13] Schnorr C P. Efficient signature generation by smart cards[J]. Journal of Cryptology, 1991, 4: 161-174.

[14] Krawczyk H. SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols[C]//Crypto. 2003, 2729: 400-425.

[15] Gabizon A, Williamson Z J, and Ciobotaru O. PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge[J]. Cryptology ePrint Archive, Report 2019/953. https://eprint.iacr.org/2019/953.