MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 1, pp. 342–354 (2025)

Zero-knowledge proof framework for privacy-preserving financial compliance

Solomka I. R., Liubinskyi B. B.

Lviv Polytechnic National University,
12 S. Bandera Str., 79013, Lviv, Ukraine

(Received 20 February 2025; Revised 22 March 2025; Accepted 25 March 2025)

This article presents a minimal viable product (MVP) architecture and proof-of-concept implementation that leverages zero-knowledge proofs to conduct essential KYC checks on a blockchain network without disclosing sensitive user information. The design employs a trusted off-chain KYC provider to validate user credentials, then uses succinct cryptographic proofs, compiled and verified with Groth16, Circom, and snarkjs, to guarantee compliance on-chain. A single smart contract deployed on a test network (Sepolia) verifies these proofs while insulating personal data from public exposure. The article outlines a practical off-chain/on-chain data flow, discusses essential performance metrics such as proof generation time and gas costs, and describes limited user testing for qualitative feedback. By integrating regulated AML checks with privacy-oriented ZKP protocols, this work demonstrates that decentralized applications can satisfy stringent compliance standards while upholding the confidentiality of user identities.

Keywords: *blockchain; zero-knowledge proofs; know your customer (KYC); anti-money laundering (AML); off-chain processing; on-chain verification.*

2010 MSC: 94-04, 94A05, 94A60 **DOI:** 10.23939/mmc2025.01.342

1. Introduction

Conventional blockchain solutions face a persistent challenge in reconciling anti-money-laundering (AML) and know-your-customer (KYC) requirements with the strict privacy needs of end users. On one hand, financial regulations demand that service providers verify user identities and ensure that transactions do not violate sanctions or other compliance directives. On the other hand, exposing detailed personal information on a public ledger poses significant risks for both individuals and institutions, including data leaks and unauthorized profiling. The need therefore arises for a framework that enforces KYC and AML checks in a trustless manner while preserving user confidentiality

Financial institutions today face a paradox: the need to comply with rigorous anti-money laundering (AML) and know-your-customer (KYC) regulations while also protecting sensitive customer information. Zero-knowledge proofs (ZKPs) offer a cryptographic paradigm where one party can prove the validity of a statement without revealing any underlying details. This report reviews the evolution of ZKPs — from their theoretical inception to modern practical instantiations such as zk-SNARKs, zk-STARKs, and Bulletproofs — and surveys current research that explores their application in privacy-preserving financial compliance. We discuss how ZKPs can enable selective disclosure of transactional attributes, reconcile privacy with regulatory oversight, and address challenges like scalability and trusted setups. Finally, we outline future research directions needed to fully integrate these cryptographic techniques into financial systems.

Regulatory frameworks such as AML, KYC, and counter-terrorism financing (CTF) impose strict reporting and verification obligations on financial institutions. Simultaneously, customers and institutions are increasingly aware of the risks associated with exposing sensitive transactional and identity data. This inherent tension has spurred interest in cryptographic methods that can offer strong privacy guarantees without sacrificing compliance.

2. KYC, AML and zero-knowledge proofs

The concepts of "Know Your Customer" (KYC) and "Anti-Money Laundering" (AML) are cornerstones of modern financial regulation, designed to protect financial systems against illicit activities such as money laundering, terrorist financing, and various forms of fraud. KYC procedures involve collecting sufficient personal and financial data to ascertain that a customer is who they claim to be and that their funds derive from legitimate sources. AML regulations further mandate financial institutions to monitor transactions, flag suspicious patterns, and maintain records that enable oversight and investigation by regulatory authorities [1, 2]. Across jurisdictions worldwide, these processes have become de facto industry standards, underscoring the gravity of financial crime prevention and the need for global regulatory harmonization [3].

Despite their critical value, traditional KYC and AML mechanisms often require significant data sharing. Financial institutions maintain extensive personal records names, addresses, identity documents, transaction histories — accessible to internal compliance departments and potentially subject to disclosure during audits or investigations. Although these measures serve regulatory obligations, they can infringe upon customer privacy and raise concerns regarding the security of personal data, especially when systems risk being compromised through hacking or insider threats [4]. Furthermore, existing processes frequently operate in siloed databases, preventing seamless customer verification across different institutions and introducing redundant compliance checks that escalate operating costs.

The primary objective of KYC regulations is to ensure that financial institutions perform adequate due diligence on their clients. This diligence prevents anonymous or pseudonymous parties from exploiting the system to launder proceeds of crime or finance illegal activities [2]. Beyond meeting these legal requirements, robust KYC programs also protect the institution itself — reducing reputational risk, limiting exposure to financial penalties, and establishing a culture of compliance that fosters trust among customers and regulators alike. AML frameworks expand on these goals by stipulating that institutions must also continuously monitor customer transactions, detect patterns indicative of money laundering, and file Suspicious Activity Reports (SARs) where appropriate. Failure to comply can result in sizable fines, delicensing, and severe reputational harm, as demonstrated by various high-profile enforcement actions over the past decade [5].

Traditional KYC/AML checks typically take place within a centralized environment, where the customer's personal data is stored and processed by one or more financial intermediaries. Although many jurisdictions impose strict controls on data handling (e.g., the EU's General Data Protection Regulation, or GDPR [6]), the risk of personal information being misused or compromised remains a persistent concern. Additionally, the inherent tension between transparency for regulatory compliance and confidentiality for privacy protection becomes more pronounced when these processes are migrated to a blockchain or other distributed ledger technologies [7] (DLT). Placing sensitive customer details on a public or partially public ledger could expose them to untrusted nodes, a potentially unacceptable risk in high-stakes financial settings.

3. Zero-knowledge proofs as a potential solution

Zero-knowledge proofs (ZKPs) allow a prover to demonstrate knowledge of a secret or the validity of a transaction without revealing the secret itself. Since their original formulation in the mid-1980s [8], ZKPs have evolved from interactive protocols into efficient, non-interactive variants suitable for real-world applications. These properties make them particularly attractive for financial compliance systems where proving attributes (e.g., "this transaction meets regulatory criteria") without disclosing the underlying data is a primary goal.

The concept of zero-knowledge was first formally introduced by Goldwasser, Micali, and Rackoff [9]. In an interactive proof system, a prover convinces a verifier that a statement is true without revealing any additional information beyond the validity of the statement. Formally, a protocol is "zero-knowledge" if a simulator can generate a transcript indistinguishable from that produced in an actual execution of the protocol.

To address practical constraints — such as the need for minimal communication overhead — researchers have developed non-interactive zero-knowledge proofs (NIZKs) using methods like the Fiat–Shamir heuristic. Modern instantiations include:

- **zk-SNARKs** (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge). These provide short proofs and fast verification times. Systems such as Pinocchio [10] and Zerocash [14] have demonstrated the potential for financial anonymity.
- **zk-STARKs** (Zero-Knowledge Scalable Transparent Arguments of Knowledge). Emphasizing scalability and transparent setups, zk-STARKs eliminate the need for a trusted setup while providing strong security guarantees.
- **Bulletproofs.** Proposed by Bünz [12], Bulletproofs enable short, efficient range proofs (crucial for confidential transactions) without a trusted setup.

Zero-knowledge proofs (ZKPs) constitute a family of cryptographic protocols that enable one party (the "prover") to demonstrate knowledge of a secret or satisfy a certain property without revealing the underlying confidential information to another party (the "verifier") [2]. In a blockchain-based KYC/AML context, ZKPs allow the system to verify that a user has passed regulatory checks — such as not being on a sanctions list — without publicly exposing the user's identity documents or transactional history. This paradigm mitigates privacy concerns by storing personal data off-chain under the custody of a trusted KYC provider, while the on-chain protocol only processes succinct cryptographic proofs [11].

In many financial applications, institutions must verify that a transaction adheres to regulatory constraints (e.g., proving that funds are derived from legitimate sources or that a transaction falls within approved parameters) without revealing the underlying amounts or parties. ZKPs allow a prover (e.g., a bank or a customer) to furnish evidence of compliance (such as "the sum of inputs equals the sum of outputs in a transaction") without exposing sensitive details.

For example, in blockchain-based systems like Zerocash [11], users can transact anonymously while still providing cryptographic guarantees that the ledger remains balanced. Such systems illustrate how ZKPs can bridge the gap between privacy and auditability.

Beyond transaction verification, financial compliance often involves verifying identity attributes under KYC regulations. Systems based on anonymous credential schemes [13] allow users to prove that they possess verified credentials (such as age, residency, or financial status) without disclosing the underlying data. ZKPs play a crucial role here by enabling:

- Selective Disclosure: Proving possession of a valid credential without revealing the entire credential or personal details.
- Revocability: Allowing regulated entities to revoke credentials if necessary while maintaining user privacy until that point.

Integrating ZKPs into financial compliance systems offers several advantages:

- Enhanced Privacy: Customers retain control over sensitive data, reducing exposure to breaches.
- Improved Efficiency: Automated cryptographic proofs can streamline compliance checks and audits.
- Cross-Jurisdictional Utility: A standard cryptographic proof can serve as a verifiable claim accepted by regulators in multiple regions, easing cross-border financial transactions.

These benefits are balanced against challenges such as the computational overhead of proof generation and the complexities introduced by trusted setups (in some ZKP systems).

Ben-Sasson [11] introduced Zerocash, demonstrating how ZKPs could be used to create anonymous digital cash. Similarly, Miers [14] developed the Zerocoin protocol for Bitcoin, laying the groundwork for privacy-preserving financial transactions.

Bünz [12] Bulletproofs provide a practical method for proving that a secret value lies within a certain range — a critical requirement for confidential financial transactions where amounts must be validated without disclosure.

Camenisch and Lysyanskaya [13] proposed systems that enable anonymous credentials with controlled disclosure and revocation — a paradigm that directly informs privacy-preserving KYC mechanisms.

Contemporary research continues to refine ZKP techniques for real-world financial compliance. Some recent trends include Transparent Setup Protocols and Integration with Distributed Ledger Technologies (DLTs). zk-STARKs and similar protocols address one of the major concerns of zk-SNARKs—the need for a trusted setup—thereby enhancing security in distributed financial systems.

Research in blockchain and distributed systems (e.g., in works such as those surveying the applications of privacy-enhancing technologies in blockchain) explores how ZKPs can underpin both transactional privacy and regulatory oversight. These studies [12] examine trade-offs between scalability and security and suggest hybrid models where only key attributes are proven via ZKPs while transaction metadata remains auditable.

Pilot projects and industry consortia are beginning to test frameworks where ZKPs are used to "attest" compliance with AML/KYC rules. For example, some proposals enable a customer to prove that they have passed a KYC check without disclosing any additional personal information, thus preserving privacy while satisfying regulatory inquiries.

While modern ZKP systems (e.g., Bulletproofs, zk-STARKs) have significantly reduced proof sizes and verification times, integrating these systems into high-frequency financial environments remains challenging. Ongoing research is needed to optimize proof generation for large-scale systems without incurring prohibitive computational costs.

Many efficient ZKP systems rely on a trusted setup phase, which can be a point of vulnerability. Future work is focusing on "transparent" protocols (e.g., zk-STARKs) to eliminate such dependencies while maintaining efficiency and succinctness.

Beyond the cryptographic primitives, integrating ZKP-based systems into existing financial infrastructures requires robust interfaces, user-friendly key management, and comprehensive auditability. Bridging the gap between cutting-edge cryptography and practical, deployable systems is an active area of research and development.

By reconciling the contradictory demands of regulatory transparency and customer privacy, zero-knowledge proof systems offer a promising route for secure, trust-minimized compliance in decentralized [19] or partially decentralized settings [18]. However, deploying these solutions entails complexities, ranging from circuit design and trusted setup procedures to key management and post-quantum considerations. The work undertaken in this article contributes to the growing literature on applying zero-knowledge proofs to financial compliance, seeking to demonstrate that KYC and AML checks can be both effective and privacy-preserving when the appropriate cryptographic tools are integrated into modern blockchain architecture.

4. Zero-knowledge proofs: algorithmic framework and integration

Zero-knowledge proofs are cryptographic protocols that enable one party (the prover) to demonstrate the truth of a statement to another party (the verifier) without conveying any information about the witness beyond the validity of the claim [8,9]. Our framework adapts a circuit-based ZKP algorithm — such as Groth16 [15] — to the regulatory compliance context. The algorithm is organized into three phases.

Let C be an arithmetic circuit encoding a predicate $P(x, w)$. Here, x is the public input (or statement) and w is the secret witness (private data). In a KYC scenario, the secret witness might encapsulate the user's unique credential issued by a KYC provider, while the public input could contain a commitment or hash references needed for verification.

Phase 1. Setup

$$(pk, vk) \leftarrow \mathrm{Setup}(\lambda, C).$$

The setup procedure takes the security parameter λ and the circuit C as inputs. It generates two key objects:

- pk: The **proving key**, used by the prover to generate succinct proof.
- vk: The **verification key**, used by the verifier (e.g., a smart contract) to check correctness without learning the witness w.

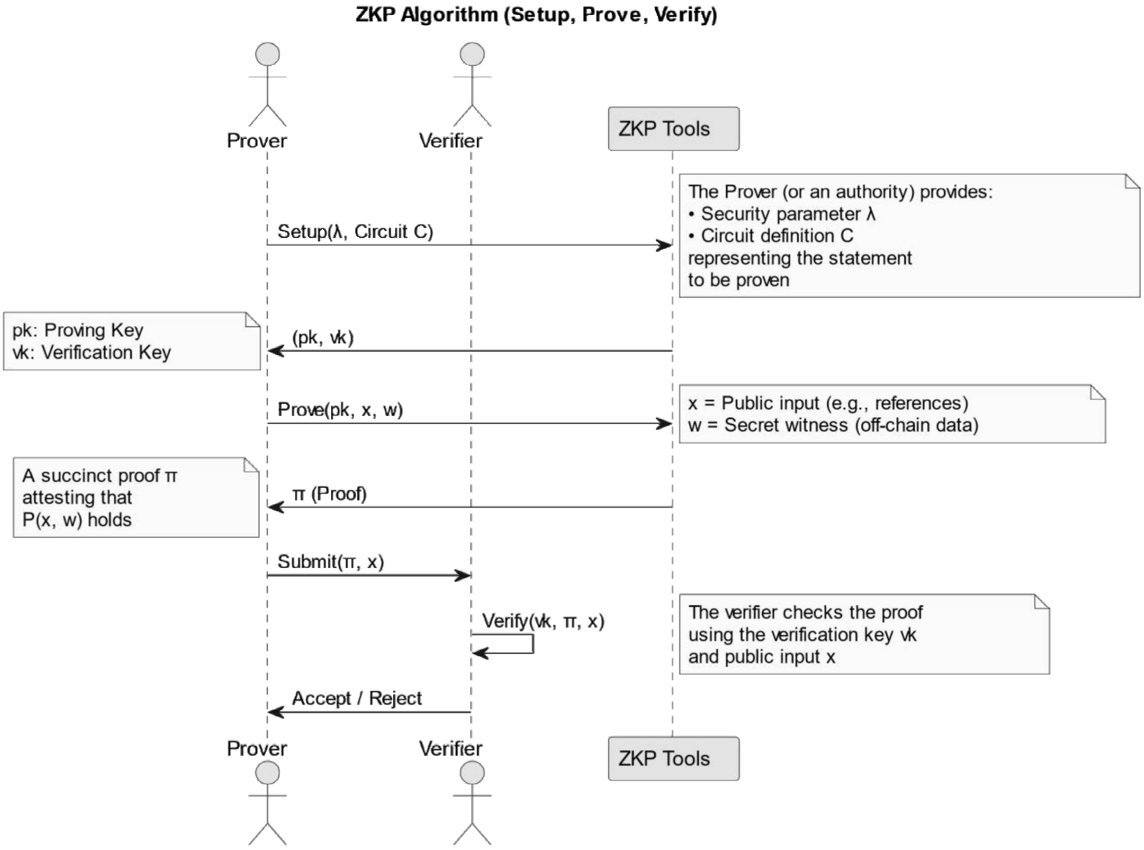

Fig. 1. Zero knowledge proof flow algorithm.

Depending on the proof system, this step may or may not involve a trusted setup phase (for example, Groth16 requires a one-time secret ceremony, while STARK-based systems do not).

Phase 2. Prove

$$\pi \leftarrow \text{Prove}(pk, x, w).$$

Given the proving key pk, the public input x and the secret witness w, the prover (i.e., the user) computes a proof π. This proof asserts that $P(x, w)$ holds true — concretely, that the user's credential passes the KYC constraint embedded in C. The proof π is typically a small, cryptographically sealed artifact that can be posted on-chain without revealing w.

Phase 3. Verify

$$b \leftarrow \text{Verify}(vk, x, \pi).$$

The verifier (a smart contract deployed on a blockchain) takes the verification key vk, the public input x, and the proof π. It outputs a Boolean value $b \in \{true, false\}$, indicating whether the proof is valid. A result of "true" confirms that there exists a secret witness w satisfying the circuit constraints, without exposing the witness itself.

The predicate $P(x, w)$ in a KYC context typically checks conditions such as $userID \in sanctionList$ or $user's\ credential == valid$. This predicate is transformed into an arithmetic circuit C that the proof system can interpret. For example:

1. The user's credential might be a hash-commitment *comm* referencing off-chain data attesting to compliance.
2. The circuit verifies the credential is signed by a recognized KYC provider and does not appear in an internal list representing sanctioned entities.
3. The final output wire (a Boolean) signals "compliant" if all constraints hold.

For On-Chain Integration the verification key vk is typically embedded within or referenced by the deployed smart contract. User broadcasts π (and any necessary public inputs x) to the blockchain as part of a transaction. The contract invokes $\text{Verify}(vk, x, \pi)$. If "true," the contract can record that "this user has demonstrated compliance," thus permitting further operations without revealing the underlying user identity or additional personal details. An honest prover with a valid witness w will always produce a proof π that causes Verify to accept. No malicious prover can produce a valid proof π if the witness w does not actually satisfy the circuit constraints. The verifier learns nothing about w beyond the fact that $P(x, w)$ is true. Even public chain participants cannot extract private data, because the proof reveals no sensitive information.

This formal algorithmic perspective underscores how zero-knowledge proofs bridge the gap between regulated compliance checks and privacy. By encoding KYC constraints in an arithmetic circuit and verifying them on-chain, it becomes possible to honor AML mandates while keeping personal data off-chain. The succinct proof π ensures minimal overhead when integrated with a blockchain environment, allowing transactions to retain confidentiality without undermining trust or verifiability.

5. Hybrid framework system architecture

The proposed system is architected as a hybrid framework that separates the processing of sensitive data from the publicly verifiable computation, thereby ensuring regulatory compliance while maintaining strict user privacy. This architecture is composed of two primary environments: an off-chain environment that handles sensitive data processing and credential issuance, and an on-chain environment dedicated to the verification of succinct zero-knowledge proofs (ZKPs).

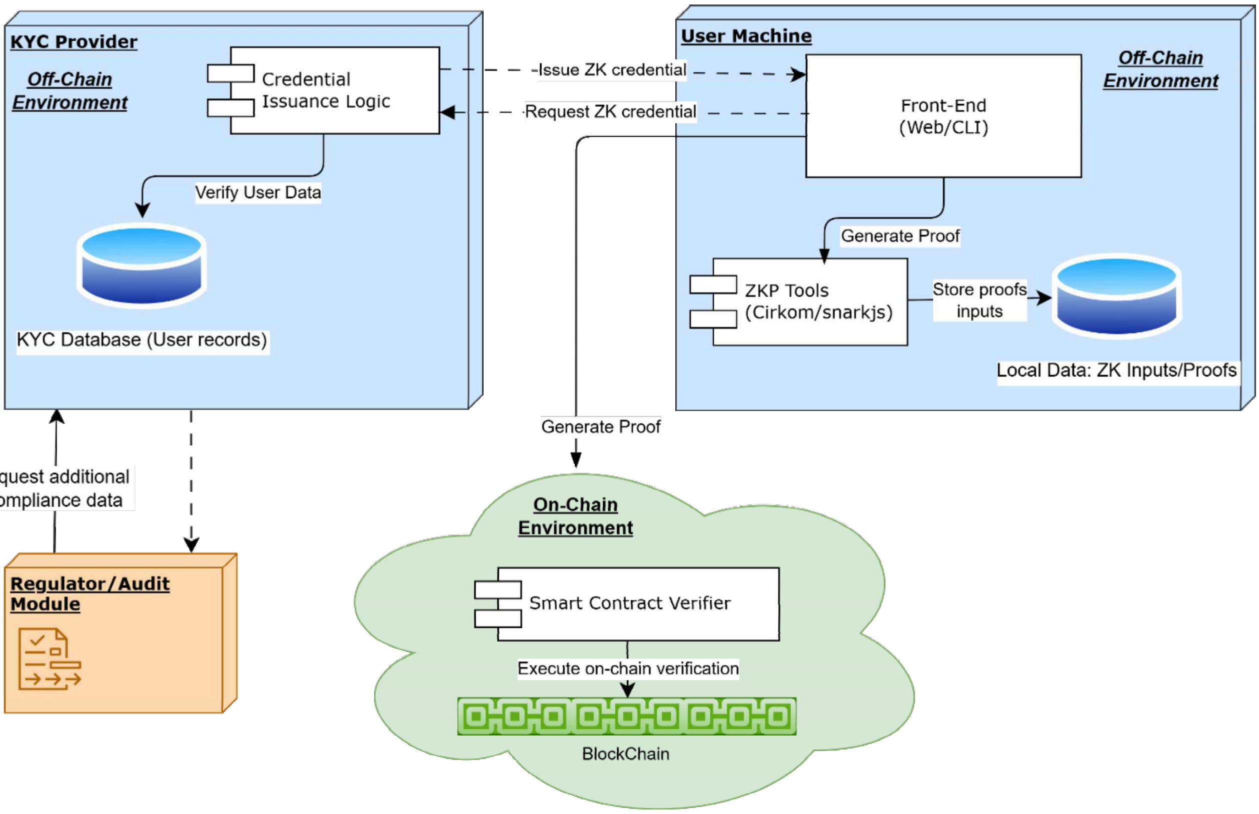

Fig. 2. Hybrid framework system architecture.

The off-chain environment is responsible for tasks that require access to personal or sensitive information, but which must remain confidential to protect user privacy. It comprises the following key components.

KYC Provider. This trusted entity is responsible for performing the initial identity verification in accordance with know-your-customer (KYC) and anti-money laundering (AML) regulations. The provider conducts background checks — such as confirming that a user is not present on any sanctions list — and, upon successful validation, issues a cryptographic credential. This credential, often implemented as a digital signature or commitment, encapsulates the user's compliance status without revealing any underlying personal data.

User Interface and Local Processing. The user interacts with the system via a web-based or command-line interface, which facilitates data submission and communicates with the KYC provider. In addition, this interface incorporates a local processing module that integrates zero-knowledge proof tools (e.g., Circom and snarkjs) to generate proofs based on the issued credential and user-specific private inputs. The resulting proof generation, which involves compiling an arithmetic circuit that models the compliance condition and computing a witness from the provided data, is entirely executed off-chain. This design ensures that raw personal data and sensitive attributes never leave the secure confines of the off-chain environment.

Data Storage. Sensitive data, including the issued credentials and the intermediate inputs required for proof generation, are stored locally in a secure manner. Only non-sensitive references (such as hashes or minimal public inputs) are generated for later use in on-chain verification.

The on-chain environment is dedicated to the transparent and trustless verification of compliance without exposing sensitive data. Its main components are:

Blockchain Network. The system is deployed on a public blockchain test network (e.g., Ethereum's Sepolia testnet [20]), which provides an immutable and decentralized ledger. This ledger serves as the execution platform for the verification logic while ensuring that all transactions are auditable.

Verifier Smart Contract. A key component of the on-chain environment is the verifier smart contract. This contract is pre-loaded with the verification key derived during the setup phase of the zero-knowledge proof system. When a user submits a transaction containing the ZKP and the associated public input, the contract executes the verification algorithm. This algorithm, based on the properties of the underlying proof system (for instance, Groth16), confirms that there exists a valid secret witness satisfying the compliance constraints without revealing any personal information. Only if the proof is verified successfully does the contract record the compliance status, thereby enabling subsequent actions (such as transaction authorization or service access) without compromising privacy.

The overall workflow is orchestrated as follows:

Credential Issuance: The user submits personal data to the KYC provider, which verifies the data and issues a credential off-chain. The credential is stored securely and is later used as part of the input to the ZKP generation process.

Proof Generation: Using the local processing module, the user assembles the required public input (e.g., hashed references) and the secret witness (derived from the credential). The zero-knowledge proof tools then generate a succinct proof, encapsulating the compliance assertion that "the user is not on a sanctions list," without revealing the underlying data.

On-Chain Verification: The user submits the generated proof and the minimal public input in a blockchain transaction that calls the verifier smart contract. The smart contract verifies the proof against the stored verification key and, upon successful validation, records the user's compliance status.

Regulatory Audit (Optional). In cases where further investigation is warranted, a regulatory auditor may request additional details from the KYC provider. Such disclosures are performed off-chain, ensuring that the on-chain process remains privacy-preserving.

This hybrid architecture leverages the strengths of both off-chain and on-chain processing. The off-chain environment allows for computationally intensive tasks and sensitive data handling without incurring the cost and exposure risks associated with on-chain operations. In contrast, the on-chain environment benefits from the inherent trustlessness, transparency, and immutability of the blockchain, enabling secure and verifiable proof verification. The overall design thus provides a scalable and privacy-preserving solution for regulatory compliance in financial systems.

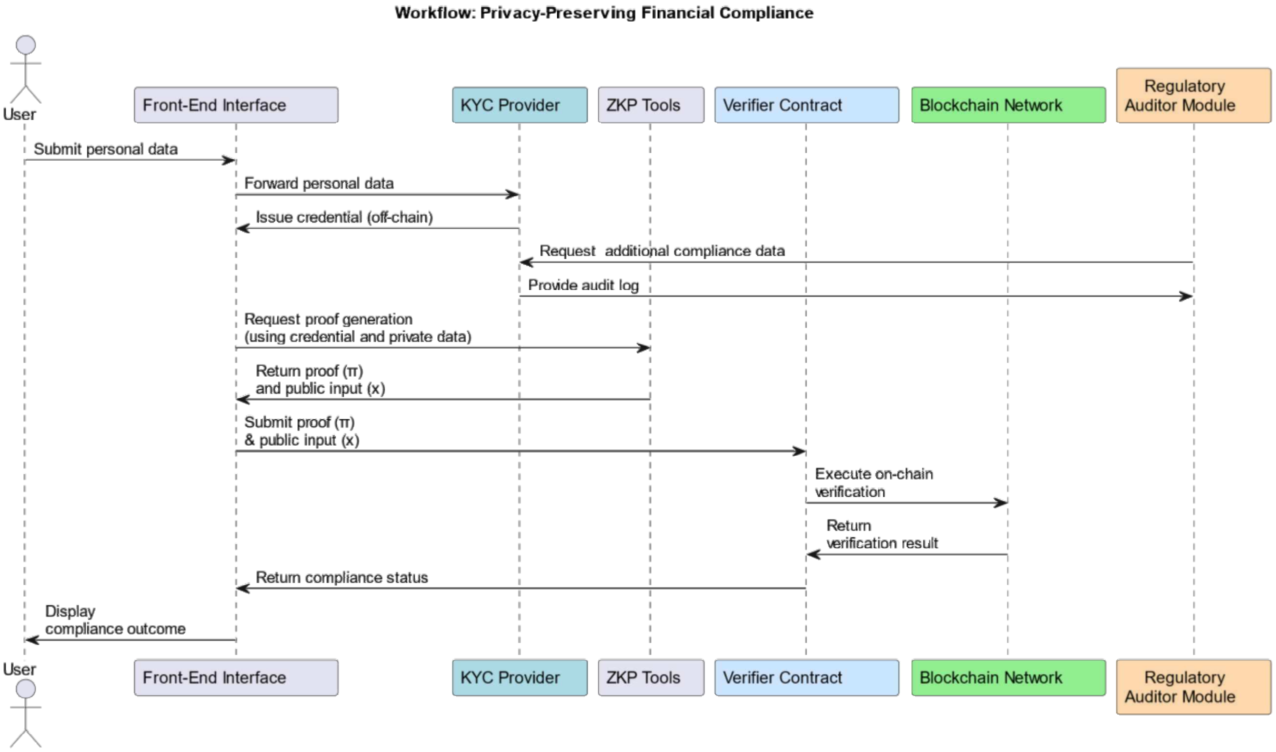
Workflow: Privacy-Preserving Financial Compliance
User
Front-End Interface
KYC Provider
ZKP Tools
Verifier Contract
Blockchain Network
Regulatory Auditor Module
Submit personal data
Forward personal data
Issue credential (off-chain)
Request additional compliance data
Provide audit log
Request proof generation (using credential and private data)
Return proof (π) and public input (x)
Submit proof (π) & public input (x)
Execute on-chain verification
Return verification result
Return compliance status
Display compliance outcome

Fig. 3. Data workflow.

6. Framework implementation details and analysis

To demonstrate the feasibility of the proposed hybrid framework, a minimal viable product (MVP) was developed, focusing on a core KYC compliance check: verifying that a user is not present on a sanctions list. This section provides details on the key components and technologies used in the MVP implementation.

Off-chain environment

- A simulated KYC provider was implemented using a simple web service that maintains a mock sanctions list and issues credentials to users who pass the check. The credential is a signed JWT containing a unique user identifier and a timestamp.
- A command-line interface (CLI) was developed to facilitate user interaction with the system. The CLI allows users to submit their data to the KYC provider, generate ZKP proofs, and interact with the on-chain verifier contract.
- The circom programming language [21] was used to define the arithmetic circuit representing the compliance condition (i.e., checking if a user is on the sanctions list). The snarkjs library was used to generate the proof based on the user's credential and the circuit definition.
- User data and credentials were stored locally in a JSON file. The MVP did not include a persistent database for simplicity.

On-chain environment

- The MVP was deployed on the Sepolia testnet [20], an Ethereum proof-of-authority network suitable for testing and development purposes.
- A Solidity smart contract [22] was developed to verify the ZKP proofs submitted by users. The contract was compiled using the solc compiler and deployed to the Sepolia network using hardhat [23]. The verification key generated during the ZKP setup phase was embedded in the contract.

According to Figure 3 Data Workflow, Users interact with the CLI to submit their data to the KYC provider. The KYC provider verifies the user's data against the mock sanctions list. If the check passes, the provider issues a signed JWT credential to the user. The user utilizes the CLI to generate a

ZKP proof using circom and snarkjs. The proof demonstrates that the user possesses a valid credential without revealing the underlying data.

The user submits the proof and associated public inputs (e.g., a nullifier to prevent double-spending) to the verifier contract on the Sepolia network. The verifier contract verifies the proof using the embedded verification key. If the proof is valid, the contract records the user's compliance status on-chain.

The system was implemented on an AMD® Ryzen™ 7 PRO 7840U running Ubuntu, and experiments were conducted using one hundred distinct customer profiles. Each simulated customer profile included critical data elements that mimic a real KYC provider's output: the customer's age, a country check indicator, a document verification hash, and a signer's public cache. These attributes are essential for establishing customer compliance while ensuring that sensitive data remains off-chain.

In our experimental setup, the process begins with the simulation of customer data, which is used to replicate the outcome of a trusted KYC provider. The simulated data is input into a compiled arithmetic circuit that enforces the compliance logic. The circuit verifies that the customer's age meets the minimum threshold, that the country check is satisfied, that the document verification hash corresponds with the expected value, and that the signature is validated against the signer's public cache. If all these constraints are met, the system generates a witness that encapsulates the secret customer data in a secure manner. This witness is then used to generate a succinct zero-knowledge proof via the Groth16 protocol [15].

The proof generation process was carefully timed across one hundred runs, yielding an average proof generation time of approximately three seconds. Although the actual times varied slightly due to transient system loads and I/O latency, the variance was minimal, indicating a robust and repeatable off-chain computation process. These results highlight the efficiency of our approach, even when using a circuit that integrates multiple compliance checks.

Following proof generation, the verifier contract exported as a Solidity contract from the final trusted setup was deployed on the Sepolia testnet. The on-chain verification phase involves submitting the generated proof, along with the minimal public inputs, to the deployed smart contract. The contract, which contains the embedded verification key, then verifies the proof using cryptographic techniques inherent in the Groth16 system. Our measurements indicated an average gas consumption of approximately 385 000 gas units per verification transaction, while the average transaction latency was observed to be around nineteen seconds. These on-chain metrics are critical for evaluating the system's economic feasibility and scalability in a decentralized environment.

Table 1. Performance metrics over 100 runs.

Metric	Average	Minimum	Maximum	Standard Deviation
Proof Generation Time (s)	0.4 s	0.25 s	0.59 s	0.12 s
Gas Consumption (gas units)	385226	385005	385263	73
Transaction Latency (s)	14 s	7 s	19.9 s	5.89 s

Our analysis further demonstrates that the performance of the system is closely tied to the complexity of the underlying circuit. As the circuit is expanded to incorporate additional compliance checks — such as more granular age validation, comprehensive country verification, and enhanced document authentication — the computational overhead is expected to increase. This would likely lead to longer proof generation times and higher gas costs during on-chain verification. Such a relationship underscores the importance of circuit optimization in the design of privacy-preserving compliance systems.

Transaction latency also shows a clear split: some variants shows consistently exhibit latencies in the 7–8 second range, while a few variants have much higher latencies (15–32 seconds), likely reflecting network congestion or variability in the deployment proces (Figure 4).

Even though the Groth16 verifier contract executes a fixed sequence of operations (resulting in a nearly identical gas cost for each verification), this static gas consumption provides a predictable

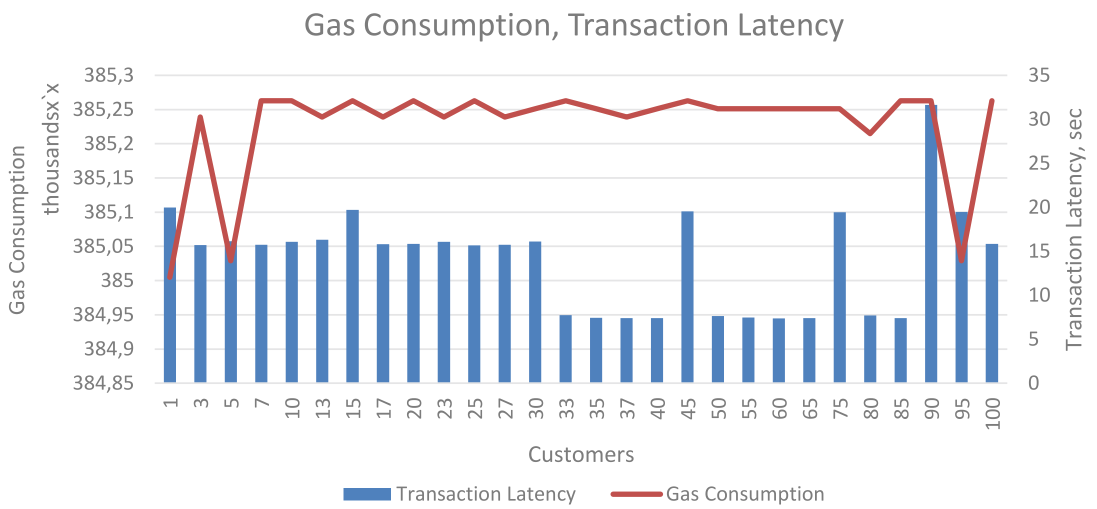

Fig. 4. Run results.

cost for compliance verification. In our evaluation, we focus on the overall system performance by measuring the time required for off-chain proof generation, which is sensitive to circuit complexity and input processing. The efficiency of the off-chain computation, combined with the on-chain verification latency, determines the end-to-end throughput of the system.

Moreover, the fixed gas cost becomes a baseline against which we can compare alternative circuit designs. By experimenting with multiple versions of the circuit (varying the number of dummy constraints or other computational elements), we can observe how changes in circuit complexity influence the overall proof generation time and the corresponding gas cost. Although the on-chain gas cost for a single transaction may remain static for a given circuit, comparing different circuits provides insight into the trade-offs between security (in terms of compliance checks) and efficiency (both computationally off-chain and economically on-chain).

7. Discussion

While zero-knowledge proofs (ZKPs) offer a promising avenue for achieving privacy-preserving financial compliance, it is essential to acknowledge the inherent limitations and trade-offs associated with their application in this context.

ZKPs, particularly those based on complex cryptographic operations, can incur significant computational overhead during proof generation and, in some cases, verification. This can be a limiting factor when dealing with high-frequency transactions or large datasets, potentially impacting the scalability and efficiency of the system. Some ZKP constructions, such as Groth16, rely on a trusted setup phase, which introduces a potential vulnerability if the setup process is compromised. Although newer protocols like zk-STARKs aim to eliminate this reliance, they may come with other trade-offs, such as increased proof size or verification time. The security of ZKP-based systems relies heavily on the secure management of cryptographic keys. Implementing robust key management protocols and ensuring user awareness of security best practices are crucial for preventing unauthorized access and potential fraud.

Constructing arithmetic circuits that accurately capture the nuances of KYC/AML regulations can be a complex and challenging task. The efficiency and security of the ZKP system depend heavily on the design of these circuits, requiring careful consideration of the trade-offs between expressiveness, performance, and security. Integrating ZKP-based systems into existing financial infrastructure can be challenging, requiring careful consideration of compatibility, interoperability, and legacy systems. This may involve significant development effort and coordination among various stakeholders.

The usability of ZKP-based systems can be a concern, particularly for users unfamiliar with cryptographic concepts. Designing user-friendly interfaces and providing clear instructions are essential for ensuring user adoption and trust.

Despite these potential limitations and trade-offs, the benefits of ZKPs in achieving privacy-preserving financial compliance are substantial. Continued research, development, and collaboration among stakeholders can help to mitigate these challenges and pave the way for the wider adoption of ZKPs in the financial industry.

The integration of zero-knowledge proofs into a regulatory compliance framework demonstrates that it is possible to verify complex KYC/AML conditions while maintaining strict privacy guarantees. The off-chain processing of sensitive data, coupled with on-chain verification of succinct proofs, effectively mitigates the risk of data exposure. Although the current implementation exhibits acceptable performance on a test network, further work is needed to optimize circuit complexity, reduce gas costs, and enhance scalability. Moreover, the framework could be extended to accommodate multi-attribute verification and dynamic regulatory requirements.

These findings underscore the importance of carefully optimizing circuit design for practical deployment. Future work could focus on further refining the circuit, exploring alternative proof systems that allow dynamic behavior, and investigating the trade-offs between circuit complexity, security, and computational overhead. Overall, the study confirms that while the on-chain costs are predictable, the off-chain computational burden is more sensitive to circuit design, providing a rich area for further optimization and performance tuning in privacy-preserving financial compliance systems.

8. Conclusions

This paper presents a zero-knowledge proof framework for privacy-preserving financial compliance, addressing the inherent conflict between regulatory transparency and data confidentiality. By decoupling sensitive data from on-chain verification, the framework enables secure and efficient KYC/AML checks without exposing personal information on a public ledger. Future work will focus on refining the circuit designs, exploring alternative ZKP constructions (such as zk-STARKs or Bulletproofs), and integrating layer-2 scaling solutions to further reduce on-chain costs. Additionally, extended user studies and regulatory feedback will be incorporated to ensure the system meets practical financial compliance needs.

Finally, the fundamental value of our MVP lies in its ability to ensure that sensitive customer data remains off-chain while still enabling robust, decentralized verification of compliance. This guarantees that privacy is maintained, and the trustless nature of the system is upheld a critical requirement for modern financial compliance systems. In essence, while gas consumption is an important metric, it must be evaluated in conjunction with other performance indicators such as proof generation time, transaction latency, and overall system security and scalability to fully assess the value of the MVP.

In conclusion, our experimental results confirm that the proposed system effectively maintains user privacy by keeping sensitive customer data off-chain and using only succinct proofs and minimal public inputs for on-chain verification. The system exhibits promising performance metrics under the simulated conditions, and the data collected provides valuable insights into the trade-offs between circuit complexity and operational efficiency. These findings not only validate our approach but also lay the groundwork for future optimizations, such as enhanced circuit design, batching of transactions, and improved scalability measures, all of which are essential for deploying such systems in real-world financial applications.

[1] Wolfsberg Group. Roles and responsibilities for publication. `https://db.wolfsberg-group.org/assets/b60cae63-3a63-46de-983a-cb22a06d14ab/PT_Roles__Responsibilities_forpublication.pdf` (2024).

[2] Financial Action Task Force. What we do. `https://www.fatf-gafi.org/en/the-fatf/what-we-do.html` (2024).

[3] Bank for International Settlements. Implementation. `https://www.bis.org/bcbs/implementation.htm?m=89`.

[4] Financial Crimes Enforcement Network. Anti-Money Laundering Act of 2020. `https://www.fincen.gov/anti-money-laundering-act-2020` (2020).

[5] European Central Bank. (2022). Annual report 2022. `https://www.ecb.europa.eu/pub/pdf/annrep/ecb.ar2022~8ae51d163b.en.pdf`.

[6] GDPR Text. `https://gdpr-text.com/`.

[7] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf` (2008).

[8] Blum M. How to Prove a Theorem So No One Else Can Claim It. Proceedings of the International Congress of Mathematicians, Berkeley, CA. 1444–1451 (1986).

[9] Goldwasser S., Micali S., Rackoff C. The knowledge complexity of interactive proof systems. SIAM Journal on Computing. **18** (1), 186–208 (1989).

[10] Parno B., Howell J., Gentry C., Raykova M. Pinocchio: Nearly practical verifiable computation. 2013 IEEE Symposium on Security and Privacy. 238–252 (2013).

[11] Ben-Sasson E., Chiesa A., Garman C., Green M., Miers I., Tromer E., Virza M. Zerocash: Decentralized anonymous payments from Bitcoin. 2014 IEEE Symposium on Security and Privacy. 459–474 (2014).

[12] Bünz B., Bootle J., Boneh D., Poelstra A., Wuille P., Maxwell G. Bulletproofs: Short proofs for confidential transactions and more. 2018 IEEE Symposium on Security and Privacy. 315–334 (2018).

[13] Camenisch J., Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. Advances in Cryptology – CRYPTO 2001. 93–118 (2001).

[14] Miers I., Garman C., Green M., Rubin A. D. Zerocoin: Anonymous distributed e-cash from Bitcoin. 2013 IEEE Symposium on Security and Privacy. 397–411 (2013).

[15] Groth J. On the size of pairing-based non-interactive arguments. Advances in Cryptology – EUROCRYPT 2016. 305–326 (2016).

[16] Wang W., Hoang D. T., Hu P., Xiong Z., Niyato D., Wang P., Wen Y., Kim D. I. A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access. **7**, 22328–22370 (2019).

[17] Cong L. W., He Z. Blockchain Disruption and Smart Contracts. The Review of Financial Studies. **32** (5), 1754–1797 (2019).

[18] Schär F. Decentralized finance: On blockchain- and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review. **103** (2), 153–174 (2021).

[19] Croman K., Decker C., Eyal I., Gencer A. E., Juels A., Kosba A., Miller A., Saxena P., Shi E., Gün Sirer E., Song D., Wattenhofer R. Financial Cryptography and Data Security. 106–125 (2016).

[20] Ethereum Foundation. Sepolia Testnet. In Ethereum Developer Documentation – Networks. `https://ethereum.org/en/developers/docs/networks/#sepolia`.

[21] Circom. Circom documentation. `https://docs.circom.io/`.

[22] Solidity. Introduction to smart contracts. `https://docs.soliditylang.org/en/latest/introduction-to-smart-contracts.html` (2025).

[23] Hardhat. Getting started. `https://hardhat.org/hardhat-runner/docs/getting-started`.

Використання доказів з нульовим розголошенням для забезпечення конфіденційності у фінансовій сфері

Соломка І. Р., Любінський Б. Б.

Національний університет "Львівська політехніка",
вул. С. Бандери, 12, 79013, Львів, Україна

Стаття реалізує концепцію доказів з нульовим розголошенням для проведення необхідних перевірок в термінах "знай свого клієнта" (know your customer) у мережі блокчейн без розкриття конфіденційної інформації клієнта. Система використовує довіреного поза мережевого постачальника послуг KYC для перевірки облікових даних користувача, а потім створює стислі криптографічні докази, скомпільовані та перевірені за допомогою бібліотек Groth16, Circom та snarkjs, для гарантування відповідності в мережі. Єдиний смарт-контракт, розгорнутий у тестовій мережі (Sepolia), перевіряє ці докази, одночасно захищаючи особисті дані від публічного розголошення. У статті описується практичний потік даних поза блокчейном та в блокчейні (on/off chain), обговорюються основні показники продуктивності, такі як час генерації доказів і витрати на транзакції, а також описується обмежене тестування користувачів для отримання якісного зворотного зв'язку. Інтегруючи регульовані перевірки AML з протоколами ZKP, орієнтованими на конфіденційність, ця робота демонструє, що децентралізовані програми можуть відповідати суворим стандартам відповідності, зберігаючи конфіденційність ідентифікаційних даних користувачів.

Ключові слова: *блокчейн, доказ з нульовим розголошенням, знай свого користувача, поза мережева обробка, перевірка в блокчейні.*