VeilAudit: Breaking the Deadlock Between Privacy and Accountability Across Blockchains

Minhao Qiao[1], Hai Dong*[,2], and Iqbal Gondal[1]

[1]RMIT University, Australia
[2]Green Cryptocurrency Joint Research Laboratory (GreenCryptoLab)
Email: minhao.qiao@student.rmit.edu.au hai.dong@rmit.edu.au iqbal.gondal@rmit.edu.au

Abstract

Cross-chain interoperability in blockchain systems exposes a fundamental tension between user privacy and regulatory accountability. Existing solutions enforce an "all-or-nothing" choice between full anonymity and mandatory identity disclosure, limiting adoption in regulated financial settings. We propose VEILAUDIT, a cross-chain auditing framework that introduces "Auditor-Only Linkability" (AOL), allowing auditors to link transaction behaviors originating from the same anonymous entity without learning its identity. VeilAudit achieves this via a user-generated "Linkable Audit Tag", which embeds a zero-knowledge proof to attest to its validity without exposing the user's master wallet address and a special ciphertext that only designated auditors can test for linkage. To balance privacy with compliance, VeilAudit further supports a threshold-gated identity revelation under due process. VEILAUDIT provides a mechanism for building reputation in pseudonymous environments, which enables applications such as cross-chain credit scoring based on verifiable behavioral history. We formalize its security guarantees and develop a prototype spanning multiple EVM chains. Our evaluation demonstrates that our framework is practical for today's multi-chain environment. (Submitted to Usenix security 2026 cycle 1 in August 2025)

1 Introduction

The proliferation of cross-chain technologies is rapidly transforming the digital landscape, breaking down the silos between once-isolated blockchain platforms [3,26]. By enabling the secure transfer of data and assets across heterogeneous networks, these interoperability protocols are unlocking novel applications in fields from finance to supply chain management [23]. However, this explosion in connectivity has set two fundamental ideals of the blockchain ecosystem on a collision course: the user's demand for robust privacy and the systemic need for accountability [6]. As users rightfully adopt powerful privacy-enhancing technologies, the transparency required for auditing and regulatory compliance is critically undermined, creating a foundational challenge for the future of decentralized systems.

This tension is not merely a philosophical debate but a deep-seated technical conflict. Modern privacy-preserving mechanisms, such as zero-knowledge proofs (ZKPs) [7, 18], transaction mixing [15, 22], and ring signatures [17], are expressly designed to obfuscate transaction details and break the chain of provenance. While highly effective at protecting user anonymity, these techniques render traditional auditing paradigms—which rely on ledger transparency—ineffective [21]c. The problem is dangerously amplified in the cross-chain context. A transaction may traverse multiple autonomous domains, each with its own identity and privacy standards, creating a fragmented and obscured trail that is nearly impossible for an external observer to reconstruct. This makes crucial tasks like transaction link analysis, behavioral tracing, and responsibility attribution exceptionally difficult.

The consequences of this unresolved conflict are severe, creating a major barrier to mainstream adoption. In finance, the lack of auditable accountability undermines the credibility of cross-chain assets for institutional investors, impedes the growth of innovative DeFi applications like under-collateralized lending, and inadvertently creates safe havens for illicit activities such as money laundering [2, 20]. Without a framework that can reconcile robust privacy guarantees with the practical needs for audit and accountability, the broader vision of a trustworthy and compliant decentralized economy remains unrealized.

To bridge this critical gap, a paradigm shift is needed in how we approach on-chain accountability. A viable solution must be built from the ground up to respect user privacy by default, while providing the necessary tools for authorized oversight. This paper aims to design such a system by addressing the core research problems at the intersection of cryptography, distributed systems, and regulatory compliance.

Research Problems. The fundamental question is: how can one design an auditing and accountability framework that

*Corresponding author: hai.dong@rmit.edu.au

preserves the strengths of secure, private, and unlinkable cross-chain transactions? This challenge manifests in four core aspects:

(1) Minimal Disclosure Auditing. How can we reconcile privacy protection with auditable transparency? The system must allow auditors to validate transaction legality and compliance without revealing any sensitive content or user identity. In particular, we ask: how can one support both *identity unlinkability* and *auditing linkability*, such that multiple transactions on different chains from the same entity can be linked by the auditor (under certain conditions), without revealing the entity itself?

(2) Auditing Completeness and Efficiency. In practice, auditors operate offline and across different time windows. They must retrospectively analyze user behaviors, often across multiple blockchains and time epochs. Therefore, audit data must be persistently available, queryable, and verifiable, without requiring interactive communication with the transaction originator. Moreover, the auditing process must scale efficiently across large datasets and heterogeneous sources (from different chains).

(3) Controlled De-anonymization. Regulatory authorities often require more than just proof of transaction validity—they demand mechanisms for identifying transactors under legal authorization. This requires the design of a controlled de-anonymization mechanism, where user identities can only be revealed when certain thresholds or legal conditions are met. Such mechanisms must prevent unauthorized identity exposure and ensure full accountability and audit traceability.

(4) Low Overhead and Performance Awareness. Cross-chain systems are inherently resource-constrained: blockchain throughput is limited, and inter-chain communication adds latency. While recursive zero-knowledge proofs provide strong privacy guarantees, they introduce prohibitive costs. Thus, we ask: how can one design privacy-preserving audit mechanisms with minimal computational and bandwidth overhead?

A related line of work, zkCross [14], introduces protocols that primarily targets privacy-preserving cross-chain identity authentication in audit. Its design focuses on enabling users to prove their state on a source chain when interacting with a target chain, while maintaining anonymity. However, zkCross does not address the auditability and accountability of transaction behaviors. In contrast, our framework is centered on transaction-level verifiability and responsibility attribution, enabling the recording, reconstruction, and aggregation of user behaviors across heterogeneous chains. This allows for compliance enforcement and risk control at the behavioral level across domains. While zkCross emphasizes anonymity and state privacy, it lacks the capability to determine whether multiple transactions originate from the same user—an essential feature for behavior clustering or entity recognition in compliance scenarios. Moreover, our framework supports threshold-controlled de-anonymization, whereby the real identity associated with a transaction can be revealed only under authorized conditions defined by the auditing chain. This mechanism satisfies the demands of financial regulation and accountability without compromising user privacy under normal operations.

1.1 VeilAudit Approach

VeilAudit resolves the privacy–accountability conflict by shifting the focus from hiding transactions to enabling the linkage of anonymous actions without revealing the master wallet addresses. Our approach avoids any form of on-chain identity registration. Instead, it introduces a novel, post-transaction auditing mechanism centered around a unique cryptographic artifact: the Linkable Audit Tag.

The core of our approach lies in a two-step process for constructing this Tag after a transaction is finalized:

First, to execute a transaction, the user generates a temporary, one-time virtual address. The user then employs a zero-knowledge proof (zk-SNARK) in a particularly novel way: not to hide the transaction amount, but to generate a cryptographic proof of ownership. This proof attests that the user controls the master wallet address that derived this temporary address, all without computationally linking the two or revealing the master wallet address itself. This step assures the auditor that the Tag is authentically generated by a legitimate, key-holding entity.

Second, embedded within each Tag is a specialized ciphertext called equality-test ciphertext ($\mathsf{CT}^{\text{link}}$). This component is created using Public-key Encryption with Equality Test (PKE-ET). It functions as a unique cryptographic lockbox: an authorized auditor holds the exclusive key to test if any two lockboxes originate from the same anonymous user. Crucially, they can verify this link but can never open the box to see the user's actual master wallet address.

This two-part design allows auditors to collect Tags on a dedicated audit chain and construct a behavioral graph of an anonymous entity's cross-chain activities. They can finally break the "all-or-nothing" privacy deadlock—enabling effective compliance through linkability auditing, while rigorously protecting the anonymity of all users.

1.2 VeilAudit Features

The VeilAudit approach introduces several features designed to reconcile the conflicting requirements of privacy and compliance in the multi-chain world.

First, it enables Auditor-Only Linkability (AOL). This is the core benefit. It creates a new paradigm where auditors can verify that Tag A on Ethereum and Tag B on Solana were created by the same anonymous entity, allowing them to detect sophisticated illicit patterns like cross-chain wash trading or money laundering. For all other parties, including other users and the public, these tags remain completely unlinked and anonymous, thus protecting herd privacy.

Second, it offers a Resource-Efficient, Non-Interactive Auditing model. Audit Tags are generated post-hoc, meaning they are only created after a transaction has been successfully confirmed on-chain. This is highly efficient, as no on-chain resources are wasted on failed or pending transactions. Furthermore, once a Tag is submitted, the process is non-interactive; auditors can perform their analysis without any further input from the user, enhancing both efficiency and privacy.

Third, VeilAudit is designed with High Adaptability for the real world. Its architecture is decoupled, using a relay bridge module to separate the audit layer from the execution layer. This modular design makes it compatible with all mainstream cross-chain technologies and allows for easy integration into existing infrastructure without requiring fundamental changes to the underlying blockchains. (Our proof-of-concept validates this by integrating with EVM chains via the Axelar bridge).

Finally, it provides a Threshold-Controlled Accountability Mechanism as a crucial safeguard. For situations where a pattern of activity is deemed highly suspicious and legally actionable, VeilAudit includes a controlled de-anonymization mechanism. This is not a backdoor. Identity revelation can only be triggered under strict, pre-defined conditions through an on-chain collaborative authorization involving multiple, independent parties (e.g., a court, a regulatory body, and a project DAO). This entire process is transparently logged, ensuring accountability is possible without enabling unauthorized access or misuse of power.

1.3 Application Scenarios: Enabling a Cross-Chain Reputation and Credit

A key application of VeilAudit is enabling portable behavioral credentials across blockchains. For instance, a user's history of repaying loans on Ethereum can be cryptographically linked and verified on Solana without revealing their identity. This addresses the "reputation black hole" in DeFi, reducing the need for extreme over-collateralization or off-chain identity disclosure. VeilAudit's Linkable Audit Tags are a practical implementation of such a credential.

This capability is critical for solving a major challenge in today's cross-chain DeFi landscape: the "Reputation Black Hole." Currently, a user's valuable on-chain history is trapped within a single blockchain. A user who has built up a stellar reputation by consistently repaying loans on Ethereum is treated as a complete unknown—and therefore untrustworthy—when they bridge assets to a lending protocol on Solana.

To mitigate this risk, platforms are forced into two undesirable extremes:
(1) Extreme Over-collateralization: They demand excessively high collateral from all anonymous users, leading to massive capital inefficiency across the ecosystem.
(2) Permissioned "Walled Gardens": They abandon the ethos of decentralization and require users to complete off-chain KYC, linking their real-world identity to their wallets, which completely sacrifices user privacy.

VeilAudit resolves this dilemma. A user can leverage VeilAudit to generate a Portable Behavioral Credential from their Ethereum transaction history. This credential, composed of linked, anonymous audit tags, cryptographically proves a consistent pattern of good behavior (e.g., "this entity has successfully closed 50 loan positions"). They can then present this credential to the Solana lending protocol. The protocol can verify the integrity of this behavioral history and, based on this newly established trust, offer the user more favorable terms, such as lower collateral requirements.

In essence, VeilAudit enables the creation of a trustworthy pseudonymous economy. Reputation becomes a portable asset, fostering greater capital efficiency and more sophisticated risk management in DeFi. This same principle extends to other critical areas, such as allowing regulators to verify compliance paths in cross-border finance without pre-emptive identity disclosure, or enabling judicial bodies to attribute responsibility in digital forensics under strict, authorized conditions.

1.4 Contributions

In summary, our contributions are as follows:
(1) We propose VeilAudit, the first general-purpose framework for cross-chain auditing that resolves the structural tension between user privacy and regulatory accountability. Our system introduces the concept of "Auditor-Only Linkability" (AOL), enabling behavioral analysis and risk management in anonymous environments without revealing master wallet addresses.
(2) We formalize the cryptographic core of VeilAudit by introducing Linkable Audit Tags (LATs). We present a novel construction that combines zero-knowledge proofs of ownership with Public-key Encryption with Equality Test (PKE-ET) to create verifiable, portable, and pseudonymous behavioral credentials. We also provide a security analysis for our construction.
(3) We present a practical, proof-of-concept implementation of VeilAudit, demonstrating its feasibility in a real-world multi-chain environment (linking two EVM chains via a Cosmos-based audit chain). Our decoupled and modular architecture ensures high adaptability and resource efficiency, making VeilAudit a practical solution for the diverse cross-chain ecosystem.
Structure of the Paper. The remainder of this paper is organized as follows. Section A presents the necessary preliminaries underlying our work. In Section 2, we introduce the general auditing architecture on which VeilAudit is built. Section 3 formalizes the complete VeilAudit system, detailing its architecture, workflow, threat model, and the core cryptographic protocols it is built upon. Section 4 provides a comprehensive security analysis of the entire system and its protocols.

Performance evaluation results are reported in Section 5. We discuss related work in Section 6, and finally conclude the paper and outline future research directions in Section 7.

2 Framework

2.1 Architectural Principles and Functional Decomposition

To address the dual goals of *privacy-preserving cross-chain behavior tracing* and *regulated accountability*, we propose a general-purpose framework for auditable systems that can operate over heterogeneous blockchains and interoperability protocols. This architecture is not tied to any specific implementation or protocol instance; rather, it is intended to serve as a **reusable design pattern** that other systems can adopt and extend. The framework organizes the system into three distinct but interoperable layers:

- **Transaction Layer:** This layer encompasses all application-specific logic and user-initiated transactions on existing blockchains. It assumes no native support for identity or audit logic. Instead, users interact using anonymous addresses derived from local keys, ensuring unlinkability by default. This layer is agnostic to how auditability is achieved.
- **Audit Layer:** Sitting orthogonally to the base chains, this layer ingests verifiable metadata from finalized transactions (e.g., control proofs, provenance signals, identity commitments). It verifies and stores audit tags that support *equality testing* and *behavioral linkage*, without disclosing identities. The layer can be instantiated as a standalone blockchain, off-chain oracle network, or privacy-preserving storage system.
- **Accountability Layer:** This top layer governs *controlled identity recovery*, typically via *threshold-decryptable capsules* submitted alongside audit metadata. It also enforces due process policies, such as legal threshold conditions and immutable proof trails. This layer is activated only when accountability is required.

This functional separation promotes **modularity, upgradability, and minimal disclosure**, allowing each layer to evolve independently. Within this model, key actors have clearly defined roles. **Users** operate on the Transaction Layer. **Auditors** are the primary actors on the Audit Layer, empowered by its equality-testing capabilities. An **Accountability Authority** (e.g., an Unmasking Committee) is the designated entity for the Accountability Layer, whose power is constrained by a foundational cryptographic setup, such as a **threshold public key** (tpk).

2.2 Technical Challenges

To achieve privacy-preserving cross-chain auditing with regulated accountability, the VeilAudit framework is designed to solve three fundamental and interrelated technical challenges. This section details each challenge and presents our cryptographic solutions.

Challenge 1: Constructing Verifiable yet Privacy-Preserving Audit Tags

The foundational component of our system is the *audit tag*, a piece of metadata that must be verifiable, unforgeable, and privacy-preserving. The core challenge is to design a tag that satisfies three distinct requirements: proving its legitimacy without revealing the user's identity; ensuring it is bound to a real transaction to prevent forgery; and enabling an auditor to link a user's activities while keeping the identity unlinkable and private even from the auditor.

Our solution is a composite tag that addresses each requirement with a specific cryptographic primitive:

(1) Legitimacy via Single-Use Anonymous Identities. To prove a tag's origin without exposing public keys, we use a zk-SNARK to generate a **single-use anonymous identity** for each transaction. The proof attests that this new identity was derived from a valid, user-controlled signing key, but the identity itself is computationally unlinkable to the user's public address or any other generated identity. This provides a cryptographically authentic yet ephemeral persona for each audit event, confirming the tag's legitimacy while preserving long-term anonymity.

(2) Integrity via Transaction Binding. To prevent forgery and replay attacks, each audit tag is structurally and immutably bound to a finalized on-chain transaction. This is achieved by embedding a cryptographic commitment (e.g., a hash) to the unique transaction record within the tag. The audit chain only accepts tags corresponding to a real, irreversible transaction, thus eliminating the possibility of synthetic tag injection.

(3) Private Linkability via Encryption with Equality Test (PKE-ET). To solve the paradox of enabling auditor-only linkage while maintaining anonymity, we employ Public-key Encryption with Equality Test. A derivative of the user's hidden master identity is encrypted under the auditor's public key (apk) to generate a ciphertext, $\mathsf{CT}^{\mathrm{link}}$. While appearing as random, unlinkable data to the public, the auditor can use their exclusive trapdoor key (tk) to test if any two $\mathsf{CT}^{\mathrm{link}}$ values correspond to the same underlying identity. This allows the auditor to link behavior across chains while the user's identity remains cryptographically unexposed.

Challenge 2: Ensuring Audit Integrity in a Trustless Cross-Chain Environment

In a decentralized and asynchronous environment, we cannot assume users will cooperate with audits. The challenge is to design a system that ensures the audit trail is complete, objective, and resistant to manipulation (like replay attacks) without requiring any user interaction post-transaction.

Our solution is a non-interactive process that ensures data integrity and low overhead:
(1) Non-Interactive Auditing for Completeness. The client-side component generates the audit tag and the accompanying zero-knowledge proof **locally and automatically** during the transaction submission process. This enables auditors to verify compliance offline and asynchronously, ensuring a complete audit trail without relying on user participation.
(2) Uniqueness Verification for Replay Resistance. The binding of each tag to a unique transaction hash and timestamp is crucial for mitigating replay attacks. The audit chain enforces a strict "one tag per transaction" rule by performing deduplication checks, preventing an attacker from polluting the audit data by repeatedly broadcasting information related to a single transaction.
(3) Low-Overhead Primitives for Practicality. To ensure the system is deployable in resource-constrained blockchain environments, the design incorporates lightweight and modular cryptographic primitives. For example, we opt for an efficient **on-chain signature challenge** to prove control of funds, avoiding the high gas costs associated with verifying ECDSA signatures inside a ZK circuit.

Challenge 3: Balancing Anonymity with Controlled De-anonymization

A purely anonymous system is incompatible with regulatory requirements. The challenge is to preserve user anonymity by default, yet enable controlled, auditable identity revelation under legally authorized conditions, ensuring that this powerful capability cannot be abused or create new risks for the user.

Our solution is a cryptographically-enforced, governance-driven de-anonymization mechanism:
(1) Identity Escrow via Threshold Encryption. Each user's master public key (pk_{master}) is encrypted under a **threshold public key** (tpk) held by a decentralized committee of authorities. No single party holds the complete key to decrypt this identity.
(2) Governance-Gated Identity Recovery. De-anonymization can only be triggered by a formal, on-chain governance process, requiring the submission of valid audit evidence and a successful vote by a quorum (t-of-n) of the authority committee. Only when a threshold number of members provide their decryption shares can the original identity be reconstructed.
(3) Post-Revelation Security. The accountability mechanism is designed to prevent user impersonation. The revealed identity (pk_{master}) serves as a unique identifier for legal purposes but is cryptographically distinct from the operational signing keys used for daily transactions. Therefore, even an adversary who obtains the revealed public key through the regulated process cannot use it to forge signatures or control the user's funds. This ensures that the de-anonymization process does not introduce new attack vectors against the user.

3 Model

3.1 Network Model

VeilAudit adopts a two-layer blockchain architecture comprising a set of **transaction chains (Layer-1)** and a dedicated **audit chain (Layer-2)**. The two layers are interconnected via a cross-chain communication protocol, Axelar, which facilitates message relay and state synchronization in a secure and decentralized manner.

Layer-1: The transaction layer consists of multiple independent, Turing-complete public blockchains. All chains in this layer are required to support smart contract execution and zero-knowledge proof primitives, enabling expressive interactions and verifiable computation. In the current implementation, VeilAudit utilizes EVM-compatible blockchains (e.g., private Ethereum instances) to ensure compatibility with existing infrastructure and tooling. Users perform transactions using their native wallet accounts, and all transactional data remains on-chain and accessible for subsequent referencing.

Layer-2: The audit layer is instantiated as an independent blockchain built using the Cosmos SDK. It is responsible for receiving and recording structured audit data transmitted from the transaction layer. The audit chain is designed to be lightweight and modular, with support for permission control and multi-party threshold mechanisms. It remains logically decoupled from the transaction chains, and does not participate in transaction execution or consensus formation within Layer-1.

Cross-Chain Communication. To bridge the two layers, VeilAudit integrates the Axelar cross-chain protocol, which provides light-client-based message validation and cryptographic signature forwarding. Bridge relayers operating in the Axelar network observe specific events on the transaction chains and encapsulate them as standardized cross-chain messages. These messages are then authenticated and submitted to the audit chain. Bridge relayers function purely as transport intermediaries, with no ability to modify, forge, or influence the content or status of the underlying transactions.

3.2 Threat Model

Participants and Trust Assumptions. The VeilAudit system consists of the following entities:

$$\mathcal{E} = \{\mathcal{U}, \mathcal{R}, \mathcal{A}, \mathcal{C}\}$$

• **Users** $\mathcal{U}$: All public participants capable of initiating transactions on Layer-1 (the transaction chains). Since the audit chain may be permissionless, $\mathcal{U}$ includes any on-chain actor without prior registration or approval. After a transaction is completed, users are assumed to abstain from any follow-up interaction or verification and may actively evade audit. They are modeled as *malicious* adversaries.

• **Relayers** $\mathcal{R}$: Bridge components under the Axelar protocol responsible for monitoring Layer-1 events and forwarding verified messages to Layer-2. Relayers are assumed to be *trusted under normal network conditions*, meaning they do not modify, drop, or forge messages, though temporary delays and interruptions are tolerated.
• **Auditors** $\mathcal{A}$: Entities operating on Layer-2 (the audit chain) to retrieve and analyze tag data from Layer-1. Auditors are modeled as *honest-but-curious*: they follow the protocol but may attempt to infer user identities or behavioral linkages by inspecting tag content and querying public Layer-1 data.
• **Authority Committee** $\mathcal{C}$: A set of compliance-approved organizations holding threshold decryption keys. This set is modeled under a *Byzantine fault-tolerant majority assumption*: identity recovery from a tag is only possible if $t \leq |\mathcal{C}|$ parties jointly authorize the operation.

Adversary Model. We define the adversary set as:

$$\mathcal{A}dv = \{\mathcal{A}dv_{\mathcal{U}},\ \mathcal{A}dv_{\mathcal{A}}\}$$

where $\mathcal{A}dv_{\mathcal{U}}$ represents malicious users, and $\mathcal{A}dv_{\mathcal{A}}$ denotes curious auditors.

Attack I: Tag Forgery. A malicious user attempts to fabricate a valid-looking tag τ' that is not derived from any legitimate transaction. The goal is to mislead the audit chain into recording falsified data.

$$\exists\ \tau' \notin \mathcal{T}_{\text{valid}} \text{ such that } \texttt{Verify}(\tau') = \texttt{True}$$

Attack II: Tag Replay. An adversary captures a valid tag τ and submits it multiple times to the audit chain to pollute records or confuse correlation algorithms.

$$\tau \in \mathcal{T}_{\text{valid}}, \quad \texttt{Count}(\tau) \gg 1$$

Attack III: Auditor-driven Tag-to-Identity Inference. An honest-but-curious auditor uses public data from Layer-1 to establish a link between a given tag τ and an on-chain address $\texttt{addr}_{\text{L1}}$.

$$\texttt{Link}(\tau) \rightarrow \texttt{addr}_{\text{L1}} \quad \text{with } \Pr > \varepsilon$$

where $\texttt{Link}(\cdot)$ uses metadata such as timestamp, interaction pattern, or contract behavior to perform inference.

Attack IV: Cross-Tag Correlation and Clustering. An auditor aggregates multiple tags and attempts to group them under a common pseudonymous identity, undermining unlinkability.

$$\exists\ \mathcal{T}' \subset \mathcal{T} \text{ such that } \texttt{Cluster}(\mathcal{T}') \rightarrow \mathcal{U}_i$$

Attack V: Unauthorized De-anonymization. An adversary attempts to bypass the threshold decryption policy and extract the original user identity bound to a tag without proper authorization.

$$\texttt{Reveal}(\texttt{addr}_{\text{origin}} \mid \tau) \quad \text{without } t\text{-out-of-}|\mathcal{C}|$$

Attack VI: Post-Disclosure Identity Impersonation. After a legitimate regulatory process reveals a user's master public key $\texttt{pk}_{\text{master}}$, an adversary attempts to impersonate that user by generating valid audit tags or control proofs without knowing the corresponding master private key $\texttt{sk}_{\text{master}}$:

$$\mathsf{Forge}(\texttt{pk}_{\text{master}}) \Rightarrow \texttt{Atag}^* \text{ or } \pi^*_{\text{ctrl}} \quad \text{s.t.} \quad \texttt{Verify}(\cdot) = \texttt{True}$$

3.3 VeilAudit Overview

VeilAudit aims to reconcile privacy preservation with accountable cross-chain auditing by adopting a layered system architecture and a three-phase operational workflow. An overview of the full architecture and workflow is illustrated in Figure 1. As a concrete instantiation of the abstract auditing framework proposed in Section 2, VeilAudit demonstrates how key functional roles and protocol boundaries can be realized under real-world cryptographic and deployment constraints. Within this model, three core protocols—namely, the Anonymous Identity Protocol ($\mathcal{P}_{\text{AIP}}$), the Audit Protocol ($\mathcal{P}_{\text{AP}}$), and the Identity Revealed Protocol ($\mathcal{P}_{\text{IRP}}$)—are invoked at different stages to accomplish unlinkable identity generation, audit label construction, and threshold-based identity disclosure under regulatory supervision. This section provides a comprehensive overview of the system's full-process execution model, while protocol-level details are deferred to Section 3.4.

Transaction Phase. In the transaction phase of *VeilAudit*, two users, Alice and Bob (denoted as $\mathcal{U}_A$ and $\mathcal{U}_B$, respectively), initiate a privacy-preserving cross-chain transaction on their respective Layer-1 public blockchains. ① To protect user anonymity, the system first invokes the *Anonymous Identity Protocol* $\mathcal{P}_{\text{AIP}}$. ② Under this protocol, each user generates a new unlinkable anonymous address, $\texttt{addr}_A^{\text{anon}}$ and $\texttt{addr}_B^{\text{anon}}$, which is solely controlled by its owner and unlinkable to prior identities.

To ensure the validity of these anonymous addresses, each user generates a zero-knowledge proof of control π^A_{ctrl} and π^B_{ctrl}, demonstrating ownership of the corresponding anonymous address without disclosing their original public keys or identities.

For asset preparation, users deposit the intended transaction value $\texttt{val}_A$ and $\texttt{val}_B$ into neutral escrow contracts $\texttt{Esc}_A$ and $\texttt{Esc}_B$ deployed on their respective chains. These escrow contracts are system-level reusable containers not bound to any specific user. Once $\mathcal{P}_{\text{AIP}}$ verifies that both users have sufficient balances and valid control proofs, it instructs the escrows to transfer the designated amount into the anonymous addresses $\texttt{addr}_A^{\text{anon}}$ and $\texttt{addr}_B^{\text{anon}}$. ③ The subsequent cross-chain transfer is then executed between $\texttt{addr}_A^{\text{anon}}$ and $\texttt{addr}_B^{\text{anon}}$, coordinated via a cross-chain messaging bridge set $\mathcal{R}$ such as Axelar.

Concurrently, the protocol captures the users' original public keys $\texttt{pk}_A$ and $\texttt{pk}_B$, and encrypts them via threshold en-

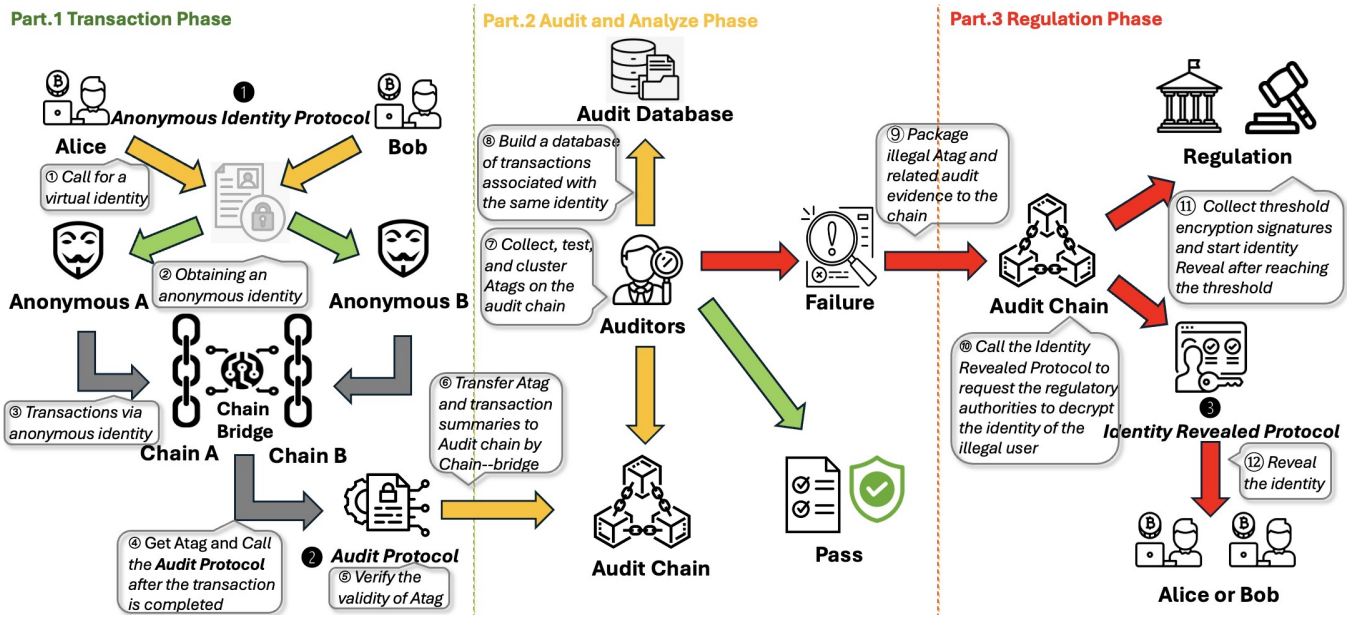


Figure 1: System Overview of VeilAudit.

cryption, resulting in ciphertexts $\texttt{UID}_A = \texttt{Enc}_{\texttt{tpk}}(\texttt{pk}_A)$ and $\texttt{UID}_B = \texttt{Enc}_{\texttt{tpk}}(\texttt{pk}_B)$. These ciphertexts are not broadcast on-chain but are reserved for later inclusion in audit metadata.

The actual cross-chain transaction is then executed by the anonymous addresses $\texttt{addr}_A^{\text{anon}}$ and $\texttt{addr}_B^{\text{anon}}$, resulting in a two-part transaction object $t = (t_{\text{L1}A}, t_{\text{L1}B})$.

Upon confirmation of t on-chain, the system triggers the *Audit Protocol* $\mathcal{P}_{\text{AP}}$. ④ This protocol generates a zero-knowledge execution proof π_{exec}, which attests that the cross-chain transaction t—comprising actions from both anonymous users—has been correctly and completely executed according to protocol semantics. π_{exec} serves as a global post-facto proof that verifies the correctness of the entire execution trace. $\texttt{CT}_A^{\text{link}}, \texttt{CT}_B^{\text{link}}$ are unlinkable identifiers derived for each user $\mathcal{U}_A$ and $\mathcal{U}_B$, respectively, serving as anchors for behavioral correlation.

The system then constructs an audit tag Atag as follows:

$$\texttt{Atag} = \mathcal{P}_{\text{AP}}(t, \pi_{\text{exec}}, \pi_{\text{link}}, \texttt{CT}_A^{\text{link}}, \texttt{CT}_B^{\text{link}}, \texttt{UID}_A, \texttt{UID}_B)$$

The tag Atag binds to a unique transaction hash txid and timestamp ts, serving as the canonical audit record. ⑤ This tag is submitted to the audit chain (Layer-2) for future regulatory analysis under privacy-preserving conditions.

Audit and Analyze Phase. ⑥ The audit tag Atag is submitted to the audit chain (Layer-2) via the bridge $\mathcal{R}$ (Axelar) and received by the auditor set $\mathcal{A}$. ⑦ - ⑧ Each auditor uses their private audit key $\texttt{sk}_{\mathcal{A}}$ to decrypt the encrypted pseudonyms $\texttt{UID}_A, \texttt{UID}_B$, enabling two levels of audit analysis:

- **Single-tag analysis:** Individual tags are assessed for transaction compliance, anomaly detection, or risk profiling.
- **Cross-tag behavioral correlation:** Auditors test whether multiple tags share the same encrypted pseudonym, identifying behavioral clusters without revealing user identity.

⑨ If the auditors detect policy violations or suspect illicit behavior based on these analyses, they formally escalate the case to the Regulation Phase. The escalation includes submitting the implicated Atag entries, associated risk evidence, and compliance justification to the regulation layer through a structured on-chain request. This process complies with pre-established audit rules and acts as a trigger for initiating threshold-based identity recovery under the oversight of regulatory authorities.

Regulation Phase ⑩ Upon the submission of a verifiable identity-reveal request from the auditors, the final phase of VeilAudit—the *Regulation Phase*—is activated. This phase is governed by a set of regulatory authorities modeled as $\mathbb{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n\}$, where each $\mathcal{G}_i$ denotes an independently authorized regulator participating in the threshold decryption protocol.

Upon receiving a claim request including the target audit tag Atag, the corresponding identity-related zero-knowledge proofs, and the behavioral evidence set $\mathcal{E}_{\text{audit}}$, each $\mathcal{G}_i$ verifies the integrity and validity of the submission under its own policy constraints. These include evaluating the consistency of π_{exec}, checking the correlation structure of $\texttt{UID}_A, \texttt{UID}_B$, and ensuring the behavioral traceability warrants identity de-anonymization.

⑪ If the claim passes a threshold policy defined by the system, i.e., $|\mathcal{G}_{\text{approve}}| \geq t$, $\mathcal{G}_{\text{approve}} \subseteq \mathbb{G}$, then a threshold decryption procedure is collaboratively triggered. Under this procedure, the encrypted identity ciphertexts $\texttt{CT}_A, \texttt{CT}_B$ embedded in Atag are jointly decrypted, yielding the original public keys $\texttt{pk}_A, \texttt{pk}_B$ associated with the transaction.

⑫ This de-anonymization procedure is strictly controlled by the regulatory quorum $\mathbb{G}$ to ensure that the anonymity of users is preserved under normal circumstances and can only

be bypassed when formal regulatory thresholds are satisfied. This guarantees that identity revelation occurs solely in accordance with law-enforced accountability and with resistance against unilateral abuse or insider collusion.

3.4 Protocol

Anonymous Identity Protocol ($\mathcal{P}_{\text{AIP}}$)

The Anonymous Identity Protocol is responsible for initializing anonymized user identities in cross-chain transactions while preserving both unlinkability and auditability. Each user $\mathcal{U}_i$ executes the protocol locally before engaging in any transaction, ensuring that the resulting identities are unlinkable across sessions but remain auditable in later stages.

- **Anonymous Address Derivation.** At the beginning of each transaction session, the user $\mathcal{U}_i$ derives a fresh anonymous key pair from its long-term master key. Specifically, the master private key $\mathtt{sk}_{\text{master}}$ is combined with a session-specific salt $\mathtt{salt}_{\text{addr}}$ through a key derivation function to produce $\mathtt{sk}_i^{\text{anon}}$. The corresponding public key yields the anonymous address $\mathtt{addr}_i^{\text{anon}}$. This process guarantees that every transaction session produces a unique address, thereby achieving unlinkability between different transactions.
- **Zero-Knowledge Control Proof π^i_{ctrl}.** To ensure that escrow contracts only transfer funds to legitimate anonymous addresses, the user must provide a zero-knowledge proof π^i_{ctrl}. This proof attests that $\mathcal{U}_i$ indeed knows the secret key associated with $\mathtt{addr}_i^{\text{anon}}$, without revealing the key itself. The escrow contract verifies π^i_{ctrl} before releasing funds, ensuring that only the rightful owner can initiate transactions. (See Algorithm 1 and Circuit 4 for details.)
- **Execution Validity Proof π_{exec}.** Once the escrow verification succeeds, the anonymous addresses execute the cross-chain transaction. To prevent replay attacks and ensure correctness, the protocol generates a proof π_{exec} that attests to the full execution of the cross-chain workflow under the system-defined semantics. This proof guarantees that the transaction trace—including contract calls, bridge relays, and asset transfers—is unique, authenticated, and non-replayable. (See Circuit 5 for the construction.)
- **Threshold Identity Encryption.** For accountability, the original public key of each user, pk_i, is encrypted under the global threshold public key $\mathtt{tpk}$. The resulting ciphertext $\mathtt{UID}_i = \mathsf{Enc}_{\mathtt{tpk}}(pk_i)$ is unlinkable to any transaction unless a regulated decryption is later authorized. This ensures that the system can, under due process, recover identities when legally mandated.
- **Commitments and Equality-Test Ciphertexts.** To support long-term behavioral auditing, each user commits to its master key by publishing $\mathtt{Com}_{\mathtt{pk},i}$, a binding but hiding Pedersen-style commitment. In addition, the user computes an equality-test ciphertext $\mathsf{CT}_i^{\text{link}}$ derived from the same committed identity. This ciphertext supports equality testing under a special audit key, enabling auditors to determine whether two different audit tags originate from the same hidden user. Importantly, different randomizers are used for each transaction, ensuring that the ciphertexts differ across transactions while the underlying linkage remains consistent.
- **Consistency Proof π^i_{link}.** To guarantee that the published values $\mathtt{UID}_i$, $\mathtt{Com}_{\mathtt{pk},i}$, and $\mathsf{CT}_i^{\text{link}}$ are mutually consistent, the user generates a zero-knowledge proof π^i_{link}. This proof demonstrates that all three artifacts are derived from the same hidden pk_i, without revealing the key itself. (See Algorithm 1 and Circuit 6 for details.) This step ensures uniqueness of the binding: although the values differ across transactions due to randomization, they remain cryptographically tied to the same identity.
- **Protocol Output.** The final output of $\mathcal{P}_{\text{AIP}}$ for user $\mathcal{U}_i$ is the tuple $\left(\mathtt{addr}_i^{\text{anon}},\ \pi^i_{\text{ctrl}},\ \pi_{\text{exec}},\ \mathtt{UID}_i,\ \mathtt{Com}_{\mathtt{pk},i},\ \mathsf{CT}_i^{\text{link}},\ \pi^i_{\text{link}}\right)$,which is later incorporated into the audit tag during the audit phase.

Audit Protocol ($\mathcal{P}_{\text{AUD}}$)

- **Inputs.** From execution, the protocol reads (i) the finalized transaction objects on both legs, together with their chain identifiers and block metadata (transaction hashes, block hashes, heights, and confirmation depths), and (ii) a replay-resistant execution proof π_{exec} attesting that the cross-chain message observed on the destination chain is exactly the one emitted on the source chain and that it has satisfied the bridge's confirmation and de-duplication rules. In our deployment with Axelar, the statement proven by π_{exec} binds the source $(\mathtt{chain\_src}, \mathtt{txid\_src})$, the destination $(\mathtt{chain\_dst}, \mathtt{txid\_dst})$, the bridge message identifier $\mathtt{msg\_id}$, and the required confirmation depth; this prevents replays across time, forks, or chains (see Algorithm 1).

  From $\mathcal{P}_{\text{AIP}}$, the protocol receives for each user $i \in \{A, B\}$ an encrypted identity capsule $\mathsf{UID}_i = \mathsf{Enc}_{\mathtt{tpk}}(\mathtt{pk}_i)$, a long-term commitment $\mathsf{Com}_{\mathtt{pk},i}$ to the user's master public key, an equality-test ciphertext $\mathsf{CT}_i^{\text{link}}$ that enables linkability checks under a trapdoor, and a consistency proof π^i_{link} showing that UID_i, $\mathsf{Com}_{\mathtt{pk},i}$, and the plaintext embedded in $\mathsf{CT}_i^{\text{link}}$ all originate from the same hidden identity (see Circuit 6). No secret is revealed to the audit layer by supplying these objects.
- **Record construction.** The auditor constructs one audit tag per cross-chain transfer. Conceptually, the tag contains: the source/destination chain identifiers, both transaction identifiers, a creation timestamp, the full execution proof π_{exec}, and, for each user, the tuple $(\mathsf{UID}_i, \mathsf{Com}_{\mathtt{pk},i}, \mathsf{CT}_i^{\text{link}}, \pi^i_{\text{link}})$. The transaction fields make the record self-contained and temporally anchored; π_{exec} certifies correctness and non-replayability; the per-user tuples provide privacy-preserving anchors for later clustering and (if authorized) identity recovery. No plain identities or anonymous addresses are stored inside the tag.
- **Transmission and storage.** The tag is emitted as a single cross-chain payload through the same interoperability layer

(e.g., Axelar) and committed to the audit chain. On arrival, the audit contract verifies π_{exec} and both $\pi^{A}_{\text{link}}, \pi^{B}_{\text{link}}$. Verification failure discards the tag. Successful verification appends the tag to the append-only ledger of audit records, keyed by the destination transaction identifier and the bridge message identifier. This keying, together with the statement bound by π_{exec}, enforces deduplication even under adversarial re-submission.

• **Downstream use.** Equality testing over $\mathsf{CT}^{\text{link}}_{A}$ and $\mathsf{CT}^{\text{link}}_{B}$—and across tags over time—enables auditors holding the trapdoor to perform behavioral clustering without learning identities. When a lawful unmasking is required, threshold decryption of UID_i reveals $\mathtt{pk}_i$, whose hash matches the public commitment $\mathsf{Com}_{\mathtt{pk},i}$ already stored in the tag. The combination of UID, $\mathsf{Com}_{\mathtt{pk}}$, and $\mathsf{CT}^{\text{link}}$, validated by π_{link}, guarantees that clustering and unmasking refer to the same underlying identity, while π_{exec} guarantees that the clustered events correspond to actually executed and non-replayable cross-chain transfers.

• **Output.** The protocol outputs the on-chain audit record (the tag) on the audit chain. This record is immutable, unlinkable by default to any real-world identity, and sufficient for later accountability procedures through verification of π_{exec} and π_{link} and, if authorized, threshold decryption of UID.

Identity Revealed Protocol ($\mathcal{P}_{\mathbf{IRP}}$)

• **Goal and setting.** When ex-post audit analyses identify suspicious or policy-violating behavior across a cluster of audit tags, the Identity Revealed Protocol enables *regulated* de-anonymization under a t-of-n committee threshold. The protocol discloses only the long-term master public keys of the implicated principals and only after multi-party authorization. Ordinary transactions remain anonymous and unlinkable.

• **Inputs.** A disclosure request contains a case identifier $\mathtt{caseID}$, a set of implicated audit tags $\{\mathtt{Atag}_1, \ldots, \mathtt{Atag}_k\}$, and the auditor's clustering evidence indicating that these tags correspond to the same anonymous principal via equality tests on the linkable ciphertexts. Each $\mathtt{Atag}$ (as produced by $\mathcal{P}_{\text{AUD}}$) already carries: the execution proof π_{exec}, the identity ciphertexts $\mathtt{UID}_i = \mathsf{Enc}_{\mathtt{tpk}}(\mathtt{pk}_i)$, the commitments $\mathsf{Com}_{\mathtt{pk},i}$, the equality-test ciphertexts $\mathsf{CT}^{\text{link}}_i$, and the consistency proofs π^{i}_{link}.

• **Request submission.** The auditor submits $(\mathtt{caseID},\ \{\mathtt{Atag}_\ell\}_{\ell=1}^{k},\ \text{clustering evidence})$. The clustering evidence states that all tags in the set are linked to the same plaintext $m = H\big(\mathsf{Com}_{\mathtt{pk}} \,\|\, \mathtt{ctx}_{\mathsf{GLOBAL}}\big)$ as witnessed by successful equality tests on the corresponding $\mathsf{CT}^{\text{link}}$ values under the authorized trapdoor. By construction, each π^{i}_{link} (cf. Circuit 6) attests that $\mathtt{UID}_i$, $\mathsf{Com}_{\mathtt{pk},i}$, and $\mathsf{CT}^{\text{link}}_i$ are derived consistently from the same hidden master identity; thus, the cluster can be interpreted as a single anonymous principal.

• **Committee authorization.** Let $\mathbb{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$ be the regulator committee with policy Π_{reg} and threshold t. Each member independently reviews the request against Π_{reg} (e.g., statutory triggers, severity, proportionality) and issues an approval or denial. If the number of approvals reaches t, the system records an authorization artifact containing the approving set and a timestamp and proceeds to threshold decryption.

• **Threshold decryption.** For each implicated party i, committee members produce partial decryption shares $\mathsf{Dec}_{\mathcal{G}_j}(\mathtt{UID}_i)$. Once at least t valid shares are collected, the system computes $\mathtt{pk}_i^{\text{master}} \leftarrow \mathsf{Combine}\big(\{\text{shares}_i\}\big)$, revealing the master public key associated with the clustered tags. If multiple tags in the cluster correspond to the same $\mathsf{Com}_{\mathtt{pk}}$, a single decryption suffices; the remaining tags are linked to the result through $\mathsf{Com}_{\mathtt{pk}}$.

• **Result materialization and accountability.** The audit chain records the tuple $(\mathtt{caseID},\ \mathtt{Atag}_\ell$ authorization artifact, $\mathtt{pk}_i^{\text{master}})$, and emits an IdentityRevealed event. Regulators then map $\mathtt{pk}^{\text{master}}$ to off-chain KYC records and pursue actions mandated by law and policy.

• **Relation to prior artifacts.** The proof π_{exec} (cf. Circuit 5)—already included in each $\mathtt{Atag}$—attests that the cross-chain execution satisfied protocol semantics and anti-replay constraints. The proof π^{i}_{link} (cf. Circuit 6) guarantees the internal consistency among $\mathtt{UID}_i$, $\mathsf{Com}_{\mathtt{pk},i}$, and $\mathsf{CT}^{\text{link}}_i$. The IRP itself introduces no new zero-knowledge verification; it only performs threshold-authorized decryption and on-chain materialization of the outcome.

4 Security Analysis

Our security analysis, conducted within the Random Oracle Model (ROM), demonstrates that VeliAudit is secure against any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$. By formalizing security games and providing proof sketches for each property, we address the threat scenarios defined in Section 3.2 (Attacks I–VI), along with an additional identity-impersonation scenario following public key disclosure. The analysis confirms that under these assumptions, VeliAudit successfully provides minimal-disclosure auditing, auditor-only linkability, unforgeable and replay-resistant audit tags, and threshold-gated identity revelation, with any adversary achieving only a negligible advantage.

4.1 Assumptions

We rely on the following cryptographic assumptions:

• **zk-SNARKs:** The zk-SNARK scheme Π_{SN} satisfies completeness, soundness, and zero-knowledge.

• **Hash Function:** H is collision-resistant and modeled as a random oracle.

• **Commitments:** Pedersen commitments are computationally binding and perfectly hiding.

• **Threshold Encryption:** Threshold ElGamal is IND-CPA secure in the t-out-of-n setting.

• **PKE-ET:** Public-key encryption with equality test is semantically secure; randomized encryption prevents public equality testing without the equality-test key.
• **Threshold Signatures:** BLS threshold signatures are existentially unforgeable under chosen-message attack (EUF-CMA).

4.2 Security Properties and Games

Property 1: Minimal Disclosure Auditing (MDA). Given only public chain data and the set of audit tags $\{\mathsf{Atag}\}$, no PPT adversary can compute any non-trivial function of pk_{master} with non-negligible probability (see Attack 3.2 for the corresponding adversarial goal; full proof in Appendix D.2):

$$\mathsf{Adv}^{\mathsf{MDA}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Game $\mathsf{Game}^{\mathsf{MDA}}$**:**

1. *Setup:* Challenger $\mathcal{C}$ generates $(pk_{\mathsf{master}}, sk_{\mathsf{master}})$ and corresponding Atag values for multiple transactions.
2. *Challenge:* $\mathcal{C}$ sends all public data to $\mathcal{A}$.
3. *Guess:* $\mathcal{A}$ outputs $f(pk_{\mathsf{master}})$ for some non-trivial f.

The advantage is $\Pr[f(pk_{\mathsf{master}}) \text{ correct}] - \frac{1}{|\mathcal{R}|}$, where $\mathcal{R}$ is the range of f.
Proof Sketch: Perfect hiding of commitments and IND-CPA security of both threshold encryption and PKE-ET ensure that all values related to pk_{master} are computationally indistinguishable from random.

Property 2: Auditor-Only Linkability (AOL). Without the equality-test key tk, no PPT adversary can determine whether two tags $\mathsf{Atag}_i, \mathsf{Atag}_j$ originate from the same entity with advantage better than random guessing (see Attack 3.2 for the corresponding adversarial goal; full proof in Appendix D.3):

$$\mathsf{Adv}^{\mathsf{AOL}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Game $\mathsf{Game}^{\mathsf{AOL}}$**:**

1. *Setup:* $\mathcal{C}$ generates keys and issues tags for multiple users across transactions.
2. *Challenge:* $\mathcal{C}$ picks either two tags from the same user ($b = 1$) or from different users ($b = 0$) and gives them to $\mathcal{A}$.
3. *Guess:* $\mathcal{A}$ outputs b'.

The advantage is $|\Pr[b' = b] - 1/2|$.
Proof Sketch: Since $\mathsf{CT}^{\mathsf{link}}$ encrypts $m = H(\mathsf{Com}_{\mathsf{pk}} \parallel \mathsf{ctx}_{\mathsf{GLOBAL}})$ under randomized PKE-ET, without tk all ciphertexts are computationally indistinguishable.

Property 3: Audit Tag Unforgeability (ATUF). No PPT adversary can create a valid new Atag without a valid Axelar threshold signature and a satisfying proof π_{link}(see Attack 3.2 and Attack 3.2 for the corresponding adversarial goal; full proof in Appendix D.4):

$$\mathsf{Adv}^{\mathsf{ATUF}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda).$$

Game $\mathsf{Game}^{\mathsf{ATUF}}$**:**

1. *Setup:* $\mathcal{C}$ generates public parameters and gives them to $\mathcal{A}$.
2. *Challenge:* $\mathcal{A}$ outputs a candidate Atag^*.
3. *Verify:* Atag^* is accepted if it passes signature and ZKP verification.

Advantage is the success probability of producing a fresh valid tag.
Proof Sketch: Follows from EUF-CMA security of threshold BLS signatures and zk-SNARK soundness.

Property 4: Replay Resistance (RR). Any replay of $(\mathsf{chain\_id}, \mathsf{txid}, \mathsf{seq})$ is rejected; cross-domain replay is rejected via $\mathsf{chain\_id}$ binding (see Attack 3.2 for the corresponding adversarial goal; full proof in Appendix D.5).
Game $\mathsf{Game}^{\mathsf{RR}}$**:**

1. *Setup:* $\mathcal{C}$ initializes the audit chain state.
2. *Challenge:* $\mathcal{A}$ attempts to submit a duplicate transaction record.
3. *Verify:* The audit chain rejects any duplicate.

Proof Sketch: Sequence numbers and chain binding ensure uniqueness without relying on timestamps.

Property 5: IRP Privacy and Correctness. With fewer than t shares, no coalition can learn UID; with at least t shares, reconstruction yields the correct pk_{master} Attack 3.2 for the corresponding adversarial goal; full proof in Appendix D.6).
Game $\mathsf{Game}^{\mathsf{IRP}}$**:**

1. *Setup:* $\mathcal{C}$ runs (t, n)-threshold encryption to produce UID.
2. *Challenge:* $\mathcal{A}$ obtains $t' < t$ shares.
3. *Guess:* $\mathcal{A}$ outputs a guess for UID.

Advantage is $|\Pr[\text{correct guess}] - 1/|\mathcal{M}||$.
Proof Sketch: Directly from threshold ElGamal IND-CPA security.

5 Evaluation

5.1 Implementation and Parameter Setup

We implemented VEILAUDITas a prototype with three major components: (i) a transaction layer deployed on two independent EVM-compatible chains, (ii) an audit layer built with the

Cosmos SDK, and (iii) a cross-chain bridge realized through Axelar's General Message Passing (GMP).

The cryptographic components are implemented as follows: $\mathcal{P}_{\text{AIP}}$, $\mathcal{P}_{\text{AUD}}$, and $\mathcal{P}_{\text{IRP}}$are written in Solidity (Layer-1 contracts), Go (Ethereum), and Circom 2.2.2/SnarkJS 0.7.5 (ZK circuits). For zero-knowledge proofs we adopt Groth16 due to its mature ecosystem and compact proof size. We used about 20k lines of code to implement protocols and test tools.

Our experiments were conducted in two environments. **Local Workstation:** Ubuntu 22.04.5 LTS, Intel Xeon Gold 5220R @2.20GHz (48 cores), 376GB RAM, 16GB swap. **Cloud Deployment:** AWS, EC 20 nodes, Ubuntu 22.04,Intel Xeon Platinum 8269CY 4vCPU, 8GB RAM. The local workstation is primarily used for proof generation and verification, while the cloud Server supports distributed deployment of the cross-chain bridge and latency.

All Nodes and supporting services are containerized using Docker. Kubernetes is further employed to orchestrate node deployment and simulate distributed network topologies.

Unless otherwise specified, each reported data point is the mean of 10 independent runs, with 95% confidence intervals computed via bootstrap resampling (1,000 resamples). We generate cross-chain transactions with a interaction patterns: escrow settlements (fund release after multi-party agreement). The cross-chain bridge is implemented via Axelar; For Auditor-Only Linkability (AOL), we construct ground-truth transaction pairs spanning multiple chains to measure equality-testing throughput, latency, and correctness. Details are given in Appendix E.

5.2 Off-Chain Performance Evaluation

This part validates the efficiency and practicality of *VeilAudit* through micro-benchmarks on three key aspects. First, we demonstrate that all core off-chain cryptographic components operate in the **millisecond regime**. Second, we show that its zero-knowledge proofs require approximately **one second** for off-chain generation, while on-chain verification costs remain stable at **tens of milliseconds**. Finally, by analyzing the compact scale of our circuits, we substantiate the core design philosophy of being *off-chain heavy, on-chain light*.

Performance of Core Cryptographic Primitives. We first measure the latency of the cryptographic primitives used to derive anonymous addresses and prepare audit material. As shown in **Table 1**, all operations exhibit high efficiency. These **millisecond-scale** latency results indicate that *VeilAudit*'s fundamental operations impose a negligible performance impact on the server and client-side, ensuring they do not become a system bottleneck.

Zero-Knowledge Circuit Performance and Scale. VeilAudit relies on three Groth16 circuits: a control proof (π_{ctrl}), an execution proof (π_{exec}), and a consistency/linkage proof (π_{link}). We first analyze their R1CS scale, which forms the basis for their performance.

Operation	Latency (ms)	Role in *VeilAudit*
$\mathsf{AddrGen}$	1.069	Derive a one-time anonymous address from the master key.
UIDEnc	2.746	Threshold-encrypt a masked identity for accountability.
$\mathsf{CT}^{\text{link}}$ Enc	0.273	Encrypt a linkage plaintext for equality testing.
$\mathsf{Commit}(\mathsf{Com}_{\text{pk}})$	1.307	Commit to a public key for later ZK consistency.
$m \leftarrow H(\cdot)$	0.168	Derive a global-domain message for digital signatures.

Table 1: Latency of core cryptographic primitives.

Circ	Setup (s)	Witness (s)	Prove (s)	Verify (ms)	Constraints
π_{ctrl}	2.064	0.196	1.159	18.927	1,034
π_{exec}	1.690	0.194	1.132	18.834	517
π_{link}	2.192	0.203	1.226	19.151	1,449

Table 2: Groth16 circuit scale and runtime for *VeilAudit*. Setup is one-time; Witness/Prove are per-proof off-chain; Verify is the on-chain.

The experimental results validate our design goals. Owing to the compact circuit sizes shown in **Table 2**, off-chain proving costs are stable at **1.1–1.2 seconds**, while on-chain verification latency is only ≈**20 milliseconds**. Combined with the millisecond-level latency of the core primitives, these results collectively demonstrate that *VeilAudit*, as an *off-chain heavy, on-chain light* system, offers a practical auditing solution that balances privacy and efficiency for high-throughput blockchain applications.

5.3 On-Chain Gas Evaluation

To quantify the on-chain footprint of *VeilAudit*, we decompose a full transaction into three comparable scenarios: **Baseline:** A standard public transaction consisting of an **ERC20 lock and unlock**. This scenario represents a typical on-chain interaction. **VeilAudit (emit):** Builds upon the Baseline by adding the on-chain verification of three Groth16 proofs. The resulting audit tag is recorded as a low-cost log entry (an event), which is not persisted in the chain's state but is easily accessible to off-chain services. **VeilAudit (store):** Also builds upon the Baseline with the same three proof verifications. However, the audit tag is persisted directly into the contract's state storage. This approach incurs a higher gas cost.

Cost Analysis As summarized in **Table 3**, the on-chain overhead of *VeilAudit* is dominated by ZK proof verification, which accounts for over **67%** of the total gas overhead. This on-chain cost structure, combined with the computationally expensive proving step which incurs zero gas off-chain, validates our *off-chain heavy, on-chain light* design. Furthermore, the audit tagging mechanism presents a significant cost trade-off, with event-based logging being substantially cheaper than storage persistence, offering developers architectural flexibil-

ity. *VeilAudit* adds approximately **695k–872k gas** (~$3.0–$3.8 at 1 Gwei) on top of a baseline transaction. This cost is comparable to common DeFi interactions, indicating that *VeilAudit* is economically viable for practical deployment.

5.4 Auditor-Only Linkability (AOL): Equality and Clustering

We evaluate the auditor-only linkability pipeline that operates on equality-test ciphertexts $\mathsf{CT}^{\text{link}}$. The goal is to show that (i) the equality stage sustains high throughput, and (ii) the downstream clustering faithfully reconstructs user groups.

Parameters setup. Each run processes a batch of B audit tags (each with a $\mathsf{CT}^{\text{link}}$). The auditor's *visibility* is modeled by a sampling rate $p \in (0,1]$, i.e., the fraction of tags the auditor observes. We fix the number of latent users to S and the mean tags-per-user to $\bar{k}$, and use two regimes to mimic low/high load: *low* (B=10,000, S=2,500, $\bar{k}$=4) and *high* (B=30,000, S=7,500, $\bar{k}$=4). Unless otherwise noted, we use `repeats` =5 and `warmup` =1 per point. Units are milliseconds (ms), seconds (s), and pairs per second (pairs/s).

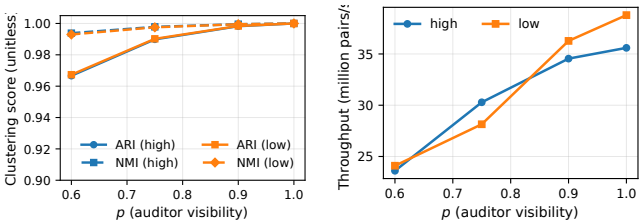
Metrics. The equality stage performs trapdoor-enabled equality tests on candidate pairs derived from the visible set. The total pairs tested are $N_{\text{pairs}} \approx p\binom{B}{2} = p\,\frac{B(B-1)}{2}$. Let T_{batch} be the wall-clock time for one batch (in seconds). We report equality throughput $Q = N_{\text{pairs}}/T_{\text{batch}}$ (pairs/s). From the equality graph we recover clusters and compare to ground truth using two standard, unitless scores in $[0,1]$: Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI); higher is better, and 1.0 indicates perfect recovery.

Results. Across realistic auditor visibility ($p \in [0.60, 1.00]$) and batch scales (*low* : B=10k, *high* : B=30k), the equality stage sustains tens of millions of comparisons per second without becoming a bottleneck. As shown in Fig. 2b, throughput remains in the $\sim$ 24–39M pairs/s range: for B=10k we observe N_{pairs}=30,004,980 in 1.245 s at p=0.60 ($\approx$ 24M pairs/s) and 49,995,000 in 1.289 s at p=1.00 ($\approx$39M pairs/s); for B=30k the trend is similar ($\sim$ 24–36M pairs/s). Clustering fidelity is already high at partial visibility and quickly approaches perfect: Fig. 2a shows ARI $\approx$ 0.97 and NMI $\approx$ 0.99 at p=0.60, rising to $\approx$ 1.0 by $p \geq$ 0.90. In short, *VeilAudit* delivers engineering-grade scalability and auditing-grade accuracy (high ARI/NMI) under realistic settings.

5.5 On-Chain Scalability

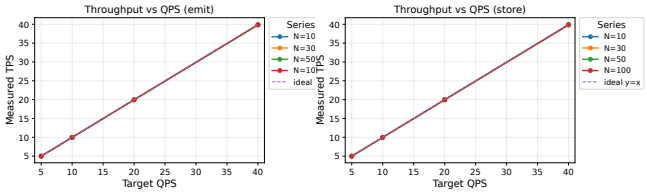
We stress-test the on-chain step of *VeilAudit* to answer two questions: (i) does throughput scale linearly with the offered load, and (ii) is end-to-end confirmation latency dominated by consensus cadence rather than contract logic?

Parameters setup. We benchmark two contract paths: **emitAtag** and **storeAtag**. Blocks are produced every T=500 ms. For each run we fix the number of concurrent senders $N \in \{10, 30, 50, 100\}$ and sweep the *offered load* (QPS)



(a) Accuracy vs. p (b) Throughput vs. p

Figure 2: AOL accuracy and throughput.



(a) TPS vs. QPS (emit) (b) TPS vs. QPS (store)

Figure 3: On-chain throughput under 500 ms blocks.

$Q \in \{5, 10, 20, 40\}$. Here, **QPS** means the aggregate *submission rate* of transactions generated by clients (before inclusion); we enforce it with a client-side rate limiter that splits the target evenly across senders, so each sender emits Q/N tx/s (e.g., Q=20, N=10 $\Rightarrow$ 2 tx/s per sender). Each operating point is sustained for 30 s, so the expected number of attempts is $\approx 30Q$, which matches our "sent" counts. We report **TPS** as the *realized throughput* (mined tx/s) computed from receipts over the same window; when the mempool isn't congested, TPS $\approx$ QPS, which is exactly what our results show. Results are representative across N because the dominant bound is the block cadence (not sender concurrency).

Metrics. P50/P95 are the 50/95th percentile of end-to-end confirmation time (submission $\rightarrow$ inclusion), in milliseconds. Average gas per transaction is reported for completeness. Under fixed block interval T, a simple arrival model predicts median confirmation $\approx T/2$ and tail $\approx T$: P50 $\approx T/2$, P95 $\approx T$ when contract execution is not bottleneck.

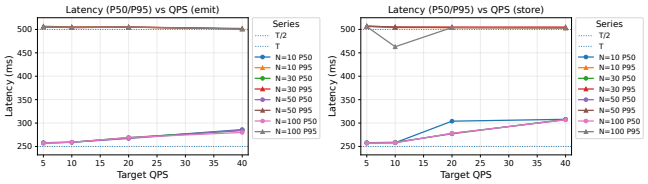
Results. Across offered loads $Q \in \{5, 10, 20, 40\}$, realized throughput tracks the ideal line (TPS $\approx Q$) in both recording modes—see Fig. 3a (`emit`) and Fig. 3b (`store`). Confirmation latency is governed by the 500 ms block cadence rather than contract execution: median P50 $\approx T/2$ and tail P95 $\approx T$ with T=500 ms in both modes (Fig. 4a, b), and is stable across $N \in \{10, 30, 50, 100\}$. In short, *VeilAudit* preserves line-rate scalability and adds negligible latency under realistic loads.

6 Related Work

We review four lines of research related to VEILAUDIT: (i) trust-minimized cross-chain communication and proof-based bridges, (ii) zero-knowledge proofs that enable auditable privacy on public ledgers, (iii) deanonymization and graph-based

Scenario	Lock	Unlock	π_{ctrl}	π_{exec}	π_{link}	Total ZK	Atag	Total	ETH	USD
Baseline (lock+unlock)	61,967	54,412	0	0	0	0	0	116,379	0.000,116	0.52
VeilAudit (emit)	61,967	54,412	221,797	214,585	228,917	665,299	29,925	811,603	0.000,812	3.61
VeilAudit (store)	61,967	54,412	221,797	214,585	228,917	665,299	206,533	988,211	0.000,988	4.39

Table 3: End-to-end on-chain gas for baseline and *VeilAudit* variants. Gasprice = 1 Gwei, 1 Ether = 109 Gwei, and 1 ETH=$4,450.



(a) P50/P95 latency vs. QPS (emit)

(b) P50/P95 latency vs. QPS (store)

Figure 4: End-to-end latency under 500 ms blocks.

analytics from the auditor's perspective, and (iv) threshold cryptography for regulated disclosure.

Trustless Cross-Chain Communication and Proof-Based Bridges. A large body of work investigates *trust-minimized* cross-chain messaging using succinct clients and proof-verified state transitions, thereby avoiding trusted relays and reducing verification costs on destination chains [9, 14, 21, 27, 29]. These systems primarily optimize *executability correctness* of asset movement or message passing across heterogeneous chains. In contrast, VEILAUDIT targets *auditor-facing verifiability*: our commit–prove–record workflow yields compact *audit tags* consumable by an audit chain, providing publicly verifiable evidence of correct execution and linkage while keeping identities encrypted for policy-gated access.

ZK for Auditable Privacy on Public Ledgers. ZK systems reconcile transparency with confidentiality via succinct on-chain verification [4, 5, 13, 24, 28, 29]. These designs inform our circuit choices (binding/hiding commitments, link proofs, and amortized verification). However, most do not expose a *behavior-linkage without identity* interface for auditors, nor a first-class, policy-bound threshold decryption path. VEILAUDIT explicitly separates *behavioral linkage* (via commitments and a link proof) from *identity revelation* (via threshold decryption of an encrypted identity capsule), so routine audits never require deanonymization.

Deanonymization, Clustering, and Auditor-Side Attacks. Prior studies show that timing, denomination patterns, and multi-hop flows can enable re-identification via clustering on transaction graphs [8, 16, 19]. We treat them as the adversarial capability our design aims to neutralize. VEILAUDIT deliberately withholds the features those attacks rely on: plaintext addresses and linkable identifiers are never exposed; amounts/denominations are hidden; cross-tag linkage is revealed only through trapdoor-gated equality proofs; and identities remain threshold-encrypted. Our goal is that, absent authorized decryption shares, an honest-but-curious auditor gains no advantage beyond protocol-visible events. Accordingly, our experiments evaluate linkage recoverability *under the VeilAudit interface* (AOL equality + clustering on audit tags), rather than reproducing public-graph deanonymization heuristics.

Threshold Cryptography and Regulated Disclosure. Threshold encryption and verifiable secret sharing align accountability with confidentiality [1, 10–12, 25]. VEILAUDIT integrates a t-out-of-n decryption path directly into the protocol semantics: audit tags carry threshold-encrypted identity capsules; policy compliance is evidenced on-chain; and only when a regulator quorum attests to policy predicates are decryption shares combined. Our analysis covers confidentiality against $< t$ colluding authorities and correctness of share verification/combination, yielding compliant identity revelation with cryptographic auditability.

7 Conclusion

This paper addressed the fundamental tension between privacy and accountability in cross-chain systems. We introduced VEILAUDIT, a framework that breaks this deadlock through a novel primitive: Auditor-Only Linkability (AOL). Our core mechanism, the Linkable Audit Tag, combines zero-knowledge proofs with specialized encryption to allow auditors to trace anonymous behaviors without deanonymizing users, while a threshold-gated protocol ensures accountability is possible only under due process.

Our prototype and evaluation demonstrate that VEILAUDIT is practical: it adds a modest on-chain footprint comparable to common DeFi interactions, while the off-chain auditing pipeline achieves engineering-grade scalability and accuracy. By enabling a trustworthy pseudonymous economy, VEILAUDIT provides a foundational layer for balancing capital efficiency with compliance in the decentralized future.

Appendix

A Preliminaries

This section introduces the core cryptographic and blockchain foundations used throughout the design of VeilAudit, including public-key wallet models, zk-SNARKs, cross-chain bridges, and threshold encryption. These primitives collectively enable privacy-preserving yet auditable interactions across multiple heterogeneous blockchain environments.

A.1 Public-Key Wallets

Public-key wallets are the foundational identity mechanism in most blockchain systems. Users independently generate a private key $sk \in \mathbb{Z}_q$, where q is a large prime defining the order of an elliptic curve group $\mathbb{G}$. The corresponding public key is computed as $pk = sk \cdot G$, where G is the standard generator of $\mathbb{G}$.

In Ethereum and many EVM-compatible blockchains, the account address is derived from the public key via a hash-based procedure: $\mathsf{addr} = \mathsf{LSB}_{160}(\mathsf{Keccak256}(pk))$, where LSB_{160} denotes the least-significant 160 bits extraction. This ensures the address is compact, efficiently verifiable, and collision-resistant under standard cryptographic assumptions. For any fixed sk, the resulting (pk, addr) pair is uniquely determined. Given that sk is chosen uniformly at random from $\mathbb{Z}_q$, the entropy of the key space is approximately 2^{256}, making key collision or impersonation attacks computationally infeasible.

In VEILAUDIT, we explicitly retain this standard wallet structure without requiring any key registration, identity binding, or modification of the existing account generation logic. All cryptographic derivations are constructed on top of the native wallet model, ensuring compatibility with existing infrastructures while preserving decentralization and self-sovereign identity.

A.2 zk-SNARKs

Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) are cryptographic proof systems that enable a prover to convince a verifier that a certain computation was correctly performed, without revealing any sensitive input data.

Formally, a zk-SNARK proof system is defined as a tuple $(\mathsf{Gen}, \mathsf{Prove}, \mathsf{Verify})$:

- $\mathsf{Gen}(C) \to (\mathsf{pk}, \mathsf{vk})$: Given a computation circuit C, a trusted setup phase generates a proving key pk and a verification key vk.
- $\mathsf{Prove}(\mathsf{pk}, x, w) \to \pi$: A prover holding a witness w for input x produces a succinct proof π attesting that $(x, w) \in R_C$.
- $\mathsf{Verify}(\mathsf{vk}, x, \pi) \to \{0, 1\}$: Anyone with the verification key vk can check the validity of π on input x.

In VEILAUDIT, zk-SNARKs are used to achieve two core guarantees. First, they prove that an anonymous audit tag is correctly derived from a legitimate secret key and contextual parameters, without disclosing the key or linking it to any identity. Second, zk-SNARKs ensure that the contents embedded in the tag—such as transaction metadata relevant for audit—are correctly constructed according to a well-formed circuit, enabling consistent validation across chains. Both usages rely on non-interactive proof generation and public verification, ensuring scalability and trustless auditability.

A.3 Cross-Chain Bridges

A *cross-chain bridge* is a protocol that enables the secure transfer of data or assets between heterogeneous blockchains. Bridges typically operate under one of several architectural paradigms: *notary schemes*, where a quorum of trusted parties attest to cross-chain events; *hashed time-lock contracts* (HTLCs), where conditional payments are enforced via hash preimages and timeouts; or *light-client–based relays*, where block headers and Merkle proofs are verified across chains to ensure event authenticity.

In VEILAUDIT, we instantiate the bridge layer using the **Axelar network**, a decentralized interoperability platform that supports message passing across multiple EVM-compatible chains and Cosmos SDK–based chains. Axelar's consensus employs a *threshold signature scheme* (TSS) to collectively sign outbound messages, ensuring that no single relayer can forge cross-chain data. Each cross-chain message is subject to a *confirmation depth* policy, requiring that the source-chain transaction is finalized with a sufficient number of confirmations before relay, thereby mitigating reorganization attacks. Furthermore, Axelar implements *deduplication* on the receiving chain to prevent replay of previously delivered messages, which is critical for the correctness of audit-tag provenance.

A.4 Threshold Encryption

Threshold encryption is a cryptographic primitive that enables secret recovery only when a quorum of authorized parties collaborate. Specifically, a secret m is encrypted under a public key pk such that it can only be decrypted when at least t out of n designated decryption parties participate. This technique enhances robustness, accountability, and resistance to abuse in sensitive operations such as identity de-anonymization.

In the context of VEILAUDIT, threshold encryption serves a critical role in supporting *authorized identity revelation*. Here, a user's identity is defined as their **original wallet public key** pk_{master}—prior to any salt-based masking or pseudonymous derivation. This value remains hidden under a threshold-encrypted payload during normal operation.

To formalize, let:

- $\mathsf{Enc}_{\mathsf{tpk}}(pk_{\mathsf{master}})$ be the ciphertext encrypted under a threshold public key tpk.
- $\mathsf{Dec}_{\mathsf{sk}_i}$ denote the partial decryption share from authority i.

The de-anonymization process is triggered only upon satisfying the on-chain authorization policy $\mathsf{Policy}_{\mathsf{reveal}}$ recorded on the audit chain. Once the number of valid signatures reaches the predefined threshold t, the original identity is reconstructed:

$$\{\mathsf{Dec}_{\mathsf{sk}_1}, \ldots, \mathsf{Dec}_{\mathsf{sk}_t}\} \xrightarrow{\text{Reconstruction}} pk_{\mathsf{master}}$$

$$\mathsf{Reveal}(\mathsf{uid}_i) \rightarrow pk_{\mathsf{master}} \text{ only if } |\Sigma_{\mathsf{reveal}}| \geq t$$

Here, $|\Sigma_{\mathsf{reveal}}|$ denotes the number of *valid authorization signatures* collected from distinct approved parties.

This mechanism guarantees that no single party can reveal a user's true identity unilaterally. It also ensures **auditability** of the de-anonymization process itself: each authorization event is publicly recorded, and every signature is verifiable against a registered public key.

By treating identity as an encrypted commitment and requiring multiparty consent for its disclosure, VEILAUDIT balances privacy protection with regulatory accountability in cross-chain transaction auditing.

A.5 Public-Key Encryption with Equality Test (PKE-ET)

Public-Key Encryption with Equality Test (PKE-ET) [**?**] is a cryptographic primitive that extends conventional public-key encryption to support equality comparison on ciphertexts without revealing the underlying plaintext. In addition to the usual public/secret key pair (pk, sk), a PKE-ET system generates a separate *equality-test key* etk that enables a designated tester to determine whether two ciphertexts encrypt the same message, while learning nothing about the message itself.

A PKE-ET scheme is defined by the following algorithms:

- $\mathsf{KeyGen}(1^\lambda) \rightarrow (pk, sk, \mathsf{etk})$: On input the security parameter λ, output a public key pk, a secret key sk, and an equality-test key etk.
- $\mathsf{Enc}(pk, m) \rightarrow c$: Encrypt a message m under pk to obtain ciphertext c.
- $\mathsf{Dec}(sk, c) \rightarrow m$: Decrypt ciphertext c under sk to recover the plaintext m.
- $\mathsf{Test}(\mathsf{etk}, c_1, c_2) \rightarrow \{0, 1\}$: Using etk, output 1 if c_1 and c_2 encrypt the same message, and 0 otherwise.

In VEILAUDIT, the plaintext m encrypted under PKE-ET is a *global behavior-equivalence string*: $m = \mathsf{Hash}(\mathsf{Com}_{\mathsf{pk}} \parallel \mathsf{domain})$, where $\mathsf{Com}_{\mathsf{pk}}$ is a commitment to the user's master public key and domain is a contextual domain separator (e.g., chain identifier). The corresponding ciphertext $\mathsf{CT}^{\mathrm{link}}$ is included in each $\mathtt{Atag}$, allowing an auditor—who possesses etk—to determine whether two tags originate from the same committed identity without learning the identity itself. The public, lacking etk, cannot perform any such linkage, thus preserving unlinkability.

B Algorithm

Algorithm 1 Anonymous Identity Protocol ($\mathcal{P}_{\mathrm{AIP}}$)

Require: Long-term master keypair $(\mathtt{sk}_{\mathrm{master}}, \mathtt{pk}_{\mathrm{master}})$; threshold public key $\mathtt{tpk}$; equality-test public key $\mathtt{apk}$; escrow contracts $\mathtt{Esc}_i$ (per chain); domain tags $\mathtt{ctx}_{\mathsf{CTRL}}, \mathtt{ctx}_{\mathsf{EXEC}}, \mathtt{ctx}_{\mathsf{GLOBAL}}$.

Ensure: Tuple $(\mathtt{addr}_i^{\mathrm{anon}}, \pi_{\mathrm{ctrl}}^i, \pi_{\mathrm{exec}}, \mathtt{UID}_i, \mathtt{Com}_{\mathtt{pk},i}, \mathsf{CT}_i^{\mathrm{link}}, \pi_{\mathrm{link}}^i)$.

(Ephemeral address) Sample fresh salt $\mathtt{salt}_{\mathrm{addr}}$.
Derive $\mathtt{sk}_i^{\mathrm{anon}} \leftarrow \mathsf{KDF}(\mathtt{sk}_{\mathrm{master}}, \mathtt{salt}_{\mathrm{addr}})$.
Compute $\mathtt{pk}_i^{\mathrm{anon}} \leftarrow \mathtt{sk}_i^{\mathrm{anon}} \cdot G, \mathtt{addr}_i^{\mathrm{anon}} \leftarrow \mathsf{Addr}(\mathtt{pk}_i^{\mathrm{anon}})$.
(Control proof) Let $c_{\mathsf{CTRL}} \leftarrow H(\mathtt{ctx}_{\mathsf{CTRL}} \parallel \mathtt{addr}_i^{\mathrm{anon}} \parallel \mathtt{nonce}_{\mathsf{sess}})$.
Produce π_{ctrl}^i using Circuit 4 with challenge c_{CTRL}.
Submit π_{ctrl}^i to $\mathtt{Esc}_i$; upon on-chain verification, $\mathtt{Esc}_i$ funds $\mathtt{addr}_i^{\mathrm{anon}}$.
(Execute x-chain) Perform cross-chain transfer between anonymous addresses (coordinated by bridge).
Obtain bridge attestation σ_{Axelar} and message m_{exec} (includes $\mathtt{src}, \mathtt{dst}, \mathtt{txid}, \mathtt{nonce}, \mathtt{depth}, \ldots$).
(Exec proof) Compute domain-separated statement $c_{\mathsf{EXEC}} \leftarrow H(\mathtt{ctx}_{\mathsf{EXEC}} \parallel m_{\mathsf{exec}})$.
Produce π_{exec} using Circuit 5 (verifies Axelar TSS signature and binds anti-replay fields).
Post π_{exec} to the audit pipeline.
(Identity encryption) Set $\mathtt{UID}_i \leftarrow \mathsf{Enc}_{\mathtt{tpk}}(\mathtt{pk}_i)$.
(Commitment & ET) Let $x \leftarrow H_{\mathbb{F}}(\mathtt{pk}_i)$, sample fixed per-user $r_\star$ (long-term), and per-transaction r (fresh).
Compute $\mathtt{Com}_{\mathtt{pk},i} \leftarrow g^x h^{r_\star}$.
Define $m \leftarrow H(\mathtt{Com}_{\mathtt{pk},i} \parallel \mathtt{ctx}_{\mathsf{GLOBAL}})$.
Compute $\mathsf{CT}_i^{\mathrm{link}} \leftarrow \mathsf{ET.Enc}(\mathtt{apk}, m; r)$.
(Consistency proof) Produce π_{link}^i with Circuit 6, proving consistency among $\mathtt{UID}_i, \mathtt{Com}_{\mathtt{pk},i}, \mathsf{CT}_i^{\mathrm{link}}$.
return $(\mathtt{addr}_i^{\mathrm{anon}}, \pi_{\mathrm{ctrl}}^i, \pi_{\mathrm{exec}}, \mathtt{UID}_i, \mathtt{Com}_{\mathtt{pk},i}, \mathsf{CT}_i^{\mathrm{link}}, \pi_{\mathrm{link}}^i)$.

Algorithm 2 Audit Protocol ($\mathcal{P}_{\text{AUD}}$)

Require: Source/destination chain ids $\mathsf{cid}_{\text{src}}, \mathsf{cid}_{\text{dst}}$; txids $\mathsf{txid}_{\text{src}}, \mathsf{txid}_{\text{dst}}$; bridge message id msgid; timestamp ts; finality depth Δ; execution proof π_{exec}.
Require: For $i \in \{A, B\}$: $(\mathtt{UID}_i,\ \mathtt{Com}_{\mathsf{pk},i},\ \mathsf{CT}_i^{\text{link}},\ \pi_{\text{link}}^i)$.
Require: Domain tags $\mathtt{ctx}_{\mathsf{EXEC}},\ \mathtt{ctx}_{\mathsf{GLOBAL}}$.
Ensure: Immutable audit record on the audit chain (or rejection on failure).

1: **(Dedup key)** $k \leftarrow H(\mathsf{cid}_{\text{dst}} \parallel \mathsf{txid}_{\text{dst}} \parallel \mathsf{msgid})$.
2: **require** $\neg$ EXISTS(k). ▷ replay/dedup guard
3: **(Verify execution) require** VERIFYEXEC$(\pi_{\text{exec}}) = 1$.
binds: $\mathsf{cid}_{\text{src}}, \mathsf{txid}_{\text{src}}, \mathsf{cid}_{\text{dst}}$.
and $\mathsf{txid}_{\text{dst}}, \mathsf{msgid}, \Delta, \mathtt{ctx}_{\mathsf{EXEC}}$.
4: **(Verify link for parties)** For $i \in \{A, B\}$ **require** VERIFYLINK$(\pi_{\text{link}}^i) = 1$.
binds: $\mathtt{UID}_i, \mathtt{Com}_{\mathsf{pk},i}, \mathsf{CT}_i^{\text{link}}, \mathtt{ctx}_{\mathsf{GLOBAL}}$.
5: **(Assemble core)**
$\mathtt{Atag.core} \leftarrow (\mathsf{cid}_{\text{src}}, \mathsf{txid}_{\text{src}}, \mathsf{cid}_{\text{dst}}, \mathsf{txid}_{\text{dst}}, \mathsf{msgid}, \mathsf{ts})$.
6: **(Bundle A)** $\mathtt{Atag.A} \leftarrow (\mathtt{UID}_A, \mathtt{Com}_{\mathsf{pk},A}, \mathsf{CT}_A^{\text{link}}, \pi_{\text{link}}^A)$.
7: **(Bundle B)** $\mathtt{Atag.B} \leftarrow (\mathtt{UID}_B, \mathtt{Com}_{\mathsf{pk},B}, \mathsf{CT}_B^{\text{link}}, \pi_{\text{link}}^B)$.
8: **(Aggregate)**
$\mathtt{Atag} \leftarrow (\mathtt{Atag.core};\ \pi_{\text{exec}};\ \mathtt{Atag.A};\ \mathtt{Atag.B})$.
9: **(Commit)** STORE$(k,\ \mathtt{Atag})$;
EMIT TagCommitted$(k,\ H(\mathtt{Atag}))$.
10: **return** k.

Algorithm 3 Identity Revealed Protocol ($\mathcal{P}_{\text{IRP}}$)

Require: Case identifier caseID;
Implicated tag set $\mathcal{S} = \{\mathtt{Atag}_1, \ldots, \mathtt{Atag}_k\}$;
Regulator policy Π_{reg};
Committee $\mathbb{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$ with threshold t.
Ensure: Recovered master public keys $\{\mathtt{pk}_u^{\text{master}}\}$ (possibly empty).

1: **Request.** Auditorsubmits $\langle \mathtt{caseID},\ \mathcal{S},\ \text{clusterEvidence} \rangle$.
2: **Vote.** Each $\mathcal{G}_j \in \mathbb{G}$ issues approve/deny according to Π_{reg}.
3: **if** $\#\{\text{approve}\} < t$ **then**
4: **return** $\varnothing$ ▷ insufficient approvals
5: **end if**
6: Record authorization artifact:
approving set, timestamp.
7: **UID aggregation.** Extract $\mathcal{U} \leftarrow \{\, \mathtt{UID} \mid \mathtt{UID} \in \mathtt{Atag}_\ell,\ \mathtt{Atag}_\ell \in \mathcal{S} \,\}$.
8: **for all** $\mathtt{UID} \in \mathcal{U}$ **do**
9: **Shares.** Collect partial decryptions:
$\mathcal{D} \leftarrow \{\mathsf{Dec}_{\mathcal{G}_j}(\mathtt{UID})\}_{j \in \mathcal{A}}$, with $|\mathcal{A}| \geq t$.
10: **Combine.** Compute $\mathtt{pk}^{\text{master}} \leftarrow \mathsf{Combine}(\mathcal{D})$.
11: Append $\mathtt{pk}^{\text{master}}$ to the output set.
12: **end for**
13: **Materialize.** Emit on the audit chain:
IdentityRevealed(caseID, $\mathcal{S}$, authorization artifact, $\{\mathtt{pk}^{\text{master}}\}$).
14: **return** $\{\mathtt{pk}^{\text{master}}\}$.

C ZK Circuit

Algorithm 4 ZK Circuit: CTRL (π_{ctrl}^i) — Control of Anonymous Address

Require: Public Inputs: anonymous address $\mathtt{addr}_i^{\text{anon}}$; challenge c_{CTRL}.
Require: Private Witness: $\mathtt{sk}_i^{\text{anon}}$, signature σ_{anon} on c_{CTRL}.
Ensure: Public Outputs: none.

1: **Key relation:** reconstruct $\mathtt{pk}_i^{\text{anon}} \leftarrow \mathtt{sk}_i^{\text{anon}} \cdot G$.
2: **Address check:** enforce $\mathsf{Addr}(\mathtt{pk}_i^{\text{anon}}) = \mathtt{addr}_i^{\text{anon}}$.
3: **Signature check:** enforce $\mathsf{VerifySig}(c_{\mathsf{CTRL}}, \sigma_{\mathsf{anon}}, \mathtt{pk}_i^{\text{anon}}) = 1$.
4: **Domain separation:** c_{CTRL} is hashed with $\mathtt{ctx}_{\mathsf{CTRL}}$, binding session nonce and address.
5: **Soundness:** the circuit proves knowledge of $\mathtt{sk}_i^{\text{anon}}$ that controls $\mathtt{addr}_i^{\text{anon}}$, without revealing it.

Algorithm 5 ZK Circuit: EXEC (π_{exec}) — Cross-Chain Execution Validity

Require: Public Inputs: message summary m_{exec} (includes $\texttt{src}, \texttt{dst}, \texttt{txid}, \texttt{nonce}, \texttt{depth}$); Axelar threshold public key $\texttt{tpk}_{\text{Axelar}}$.

Require: Private Witness: bridge signature σ_{Axelar}; optional Merkle proof(s) of inclusion for attestation batch or gateway state.

Ensure: Public Outputs: anti-replay nullifier null.

1: **Domain binding:** compute $c_{\text{EXEC}} \leftarrow H(\texttt{ctx}_{\text{EXEC}} \parallel m_{\text{exec}})$.
2: **Signature check:** enforce $\text{VerifyTSS}_{\text{Axelar}}(c_{\text{EXEC}}, \sigma_{\text{Axelar}}, \texttt{tpk}_{\text{Axelar}}) = 1$.
3: **Inclusion (optional):** if batching is used, verify Merkle inclusion of $(c_{\text{EXEC}}, \sigma_{\text{Axelar}})$ in the published batch root.
4: **Anti-replay:** compute $\text{null} \leftarrow H(\texttt{txid} \parallel \texttt{nonce} \parallel \texttt{src} \parallel \texttt{dst})$.
5: **Output:** expose null as public output so that the audit contract can mark it used exactly once.

Algorithm 6 ZK Circuit: LINK (π^i_{link}) — Consistency of $\texttt{UID}_i$, $\texttt{Com}_{\texttt{pk},i}$, $\text{CT}^{\text{link}}_i$

Require: Public Inputs: $\texttt{UID}_i$; $\texttt{Com}_{\texttt{pk},i}$; $\text{CT}^{\text{link}}_i$; audit domain $\texttt{ctx}_{\text{GLOBAL}}$.

Require: Private Witness: $\texttt{pk}_i$, $x \leftarrow H_{\mathbb{F}}(\texttt{pk}_i)$; long-term randomness $r_\star$; ET randomness r.

Ensure: Public Outputs: optional $\texttt{Com}_m$ (binding).

1: **Commitment validity:** enforce $\texttt{Com}_{\texttt{pk},i} = g^x h^{r_\star}$ with $x = H_{\mathbb{F}}(\texttt{pk}_i)$.
2: **UID correctness:** enforce $\texttt{UID}_i = \text{Enc}_{\texttt{tpk}}(\texttt{pk}_i)$ via an *encryption correctness* relation (scheme-specific constraint).
3: **Behavioral plaintext:** compute $m \leftarrow H(\texttt{Com}_{\texttt{pk},i} \parallel \texttt{ctx}_{\text{GLOBAL}})$.
4: **ET-ciphertext validity:** enforce $\text{CT}^{\text{link}}_i = \text{ET.Enc}(\texttt{apk}, m; r)$ (scheme-specific constraint).
5: **Bind output (optional):** set $\texttt{Com}_m \leftarrow \text{Com}(m)$ and expose it as a public output if the implementation uses external binding of m.
6: **Privacy note:** the circuit reveals none of $\texttt{pk}_i, r_\star, r$, yet proves all three artifacts are consistent w.r.t. the same hidden identity.

D Formal Proofs for Security Properties

This appendix provides full game-based reductions for the properties stated in Section 4, under the adversary capabilities specified in the Threat Model (Section 3.2). We consider PPT adversaries and work in the Random Oracle Model (ROM) for the hash H.

D.1 Preliminaries and Notation

Assumptions. (i) zk-SNARK Π_{SN} is complete, sound, and zero-knowledge; (ii) Pedersen commitments are perfectly hiding and computationally binding; (iii) Threshold ElGamal is IND-CPA secure in the t-out-of-n setting; (iv) Public-key encryption with equality test (PKE-ET) is semantically secure and supports equality testing *only* with the equality-test key; (v) BLS threshold signatures are EUF-CMA secure.

Notation (as in the main text). $\text{pk}_{\text{master}}$: user's master public key; Com_{pk}: Pedersen commitment to $\text{pk}_{\text{master}}$; UID: threshold-ElGamal ciphertext of $\text{pk}_{\text{master}}$; CT^{link}: PKE-ET ciphertext of $m := H(\text{Com}_{\text{pk}} \parallel \text{ctx}_{\text{GLOBAL}})$; π_{link}: zk proof that $(\text{Com}_{\text{pk}}, \text{UID}, \text{CT}^{\text{link}})$ are consistently derived per the circuit; $\texttt{Atag}$: audit tag containing the above and provenance fields $(\text{chain\_id}, \text{txid}, \text{seq})$ plus the Axelar-side threshold signature.

D.2 Property ??: Minimal Disclosure Auditing

Game. The challenger samples setup and a hidden $\text{pk}_{\text{master}}$, produces honest tags $\{\texttt{Atag}\}$ (for this and other users), and gives the adversary all public data and $\{\texttt{Atag}\}$. The adversary outputs a non-trivial function value $f(\text{pk}_{\text{master}})$. Define

$$\text{Adv}^{\text{MDA}}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ outputs } f(\text{pk}_{\text{master}})] - \max_y \Pr[y].$$

Theorem A.1. $\text{Adv}^{\text{MDA}}_{\mathcal{A}}(\lambda) \le \text{Adv}^{\text{IND-CPA}}_{\text{ThEnc}} + \text{Adv}^{\text{SEM}}_{\text{PKE-ET}} + \text{negl}(\lambda)$.

Proof. Hybrid sequence: H_0 real; H_1 simulate π_{link} (zk zero-knowledge) $\Rightarrow$ negl gap; H_2 replace $\text{UID} = \text{Enc}_{\text{tpk}}(\text{pk}_{\text{master}})$ by $\text{Enc}_{\text{tpk}}(0)$ (threshold IND-CPA); H_3 replace $\text{CT}^{\text{link}} = \text{ET.Enc}(m)$ by $\text{ET.Enc}(0)$ (PKE-ET semantic security); H_4 replace Com_{pk} by a commitment to 0 (perfect hiding; zero gap). In H_4 the distribution is independent of $\text{pk}_{\text{master}}$, so the adversary cannot compute any non-trivial $f(\text{pk}_{\text{master}})$ beyond guessing. Summing gaps yields the bound. ■

D.3 Property 4.2: Auditor-Only Linkability (AOL)

Game. In challenge bit b: if b=1, two tags come from the *same* user; if b=0, from *different* users. The adversary receives the two tags *without* the equality-test key and outputs b'. Advantage $\text{Adv}^{\text{AOL}}_{\mathcal{A}} = |\Pr[b'{=}b] - \frac{1}{2}|$.

Theorem A.2. $\text{Adv}^{\text{AOL}}_{\mathcal{A}}(\lambda) \le \text{Adv}^{\text{SEM}}_{\text{PKE-ET}}(\lambda) + \text{negl}(\lambda)$.

Proof. Simulate π_{link} (zero-knowledge). Replace both challenge CT^{link} by encryptions of 0; the only b-dependent difference was whether the plaintexts were equal. Without the equality-test key, PKE-ET semantic security renders the two distributions indistinguishable up to the stated bound. ■

Auditor correctness. An authorized auditor with the equality-test key runs $\mathsf{ET.Test}(\mathsf{CT}_i^{\mathsf{link}}, \mathsf{CT}_j^{\mathsf{link}})$ which returns 1 iff the underlying $m_i{=}m_j$. Because $m = H(\mathsf{Com}_{\mathsf{pk}} \| \mathsf{ctx}_{\mathsf{GLOBAL}})$ is stable per entity across domains and time, equality holds exactly for same-user pairs.

D.4 Property 4.2: Audit Tag Unforgeability

Game. The adversary outputs a fresh $\mathtt{Atag}^\star$ that passes the audit-chain verifier: valid Axelar threshold signature on provenance fields, valid $\pi^\star_{\mathsf{link}}$, valid execution proof(s) if required, and passes deduplication on $(\mathsf{chain\_id}, \mathsf{txid}, \mathsf{seq})$.

Theorem A.3.

$$\mathsf{Adv}^{\mathsf{ATUF}}_{\mathcal{A}} \;\le\; \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{BLS\text{-}Thresh}} + \mathsf{Adv}^{\mathsf{Sound}}_{\Pi_{\mathsf{SN}}} + \mathsf{negl}(\lambda).$$

Proof. If $\mathtt{Atag}^\star$ verifies, either (a) the Axelar threshold signature is forged (EUF-CMA break), or (b) some zk proof attests a false statement (soundness break), or (c) it corresponds to a finalized transaction, contradicting freshness given deduplication. Standard reductions give the bound. ■

D.5 Property 4.2: Replay Resistance

Game. The verifier maintains a set I of seen tuples $(\mathsf{chain\_id}, \mathsf{txid}, \mathsf{seq})$. Success means re-accepting an element already in I.

Lemma A.4. $\Pr[\text{replay accepted}] = 0$.

Proof. Insertion into I happens *after* acceptance; subsequent submissions of the same tuple are deterministically rejected. Cross-domain replays fail by the $\mathsf{chain\_id}$ binding. This is a functional (non-cryptographic) invariant. ■

D.6 Property 4.2: IRP Privacy and Correctness

Privacy game. The challenger samples (X_0, X_1) and returns $\mathsf{UID}^\star = \mathsf{Enc}_{\mathsf{tpk}}(X_b)$ for a hidden $b \in \{0,1\}$. The adversary obtains $t' < t$ valid partial decryptions and outputs b'. Advantage $|\Pr[b'{=}b] - \frac{1}{2}|$.

Theorem A.5 (IRP Privacy). $\mathsf{Adv}^{\mathsf{IRP\text{-}Priv}}_{\mathcal{A}} \le \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ThEnc}}$.

Proof. A distinguisher against the IRP privacy game yields an IND-CPA adversary for threshold ElGamal: treat the $t'{<}t$ partial decryptions as efficiently simulatable views that leak no information on the plaintext under IND-CPA; simulate missing shares via the public commitments (e.g., Feldman). ■

Correctness game. Given $\mathsf{UID} = \mathsf{Enc}_{\mathsf{tpk}}(X)$, the adversary collects t valid shares but recombination outputs $X' \neq X$ (or fails).

Theorem A.6 (IRP Correctness). $\Pr[\text{IRP recombination fails or outputs } X] \le \mathsf{negl}(\lambda)$.

Proof. With t valid shares, Shamir reconstruction is unique; malformed shares are rejected by verification against public commitments. Hence recombination returns X except with negligible probability. ■

D.7 Attack 3.2 and Post-Disclosure Unforgeability

Setting. After authorized revealing, $\mathsf{pk}_{\mathsf{master}}$ (but not $\mathsf{sk}_{\mathsf{master}}$) becomes public. We show a new valid tag bound to that key cannot be produced.

Game. The adversary is given $\mathsf{pk}_{\mathsf{master}}$ and public parameters and outputs a fresh verifying $\mathtt{Atag}^\star$ bound (per the circuit) to $\mathsf{pk}_{\mathsf{master}}$.

Theorem A.7.

$$\mathsf{Adv}^{\mathsf{PDUF}}_{\mathcal{A}} \le \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{BLS\text{-}Thresh}} + \mathsf{Adv}^{\mathsf{Sound}}_{\Pi_{\mathsf{SN}}} + \mathsf{negl}(\lambda).$$

Proof. A fresh verifying tag requires (i) a valid Axelar threshold signature (EUF-CMA hard), and (ii) a valid π_{link} on a witness consistent with $\mathsf{pk}_{\mathsf{master}}$ (soundness hard). Public knowledge of $\mathsf{pk}_{\mathsf{master}}$ does not enable producing the witness required by the circuit. Hence any successful forger breaks one of the assumptions. ■

D.8 Collected Bounds

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{MDA}}_{\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ThEnc}} + \mathsf{Adv}^{\mathsf{SEM}}_{\mathsf{PKE\text{-}ET}} + \mathsf{negl},\\
\mathsf{Adv}^{\mathsf{AOL}}_{\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{SEM}}_{\mathsf{PKE\text{-}ET}} + \mathsf{negl},\\
\mathsf{Adv}^{\mathsf{ATUF}}_{\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{BLS\text{-}Thresh}} + \mathsf{Adv}^{\mathsf{Sound}}_{\Pi_{\mathsf{SN}}} + \mathsf{negl},\\
\Pr[\text{Replay accepted}] &= 0,\\
\mathsf{Adv}^{\mathsf{IRP\text{-}Priv}}_{\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ThEnc}}, \quad \Pr[\text{IRP incorrect}] \le \mathsf{negl},\\
\mathsf{Adv}^{\mathsf{PDUF}}_{\mathcal{A}} &\le \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{BLS\text{-}Thresh}} + \mathsf{Adv}^{\mathsf{Sound}}_{\Pi_{\mathsf{SN}}} + \mathsf{negl}.
\end{aligned}$$

E Measurement Details

E.1 Bootstrap-Based 95% Confidence Intervals

We report each metric as the mean of $n = 10$ independent runs and compute 95% confidence intervals (CIs) via non-

parametric bootstrap with $B = 1{,}000$ resamples. Given observations

$$\mathcal{S} = \{x_1, x_2, \ldots, x_n\},$$

we repeat for each $b \in \{1, \ldots, B\}$: (1) draw a bootstrap sample $\mathcal{S}^{(b)}$ of size n by sampling from $\mathcal{S}$ *with replacement*; (2) compute its mean $\bar{x}^{(b)}$. Let $\mathcal{M} = \{\bar{x}^{(1)}, \ldots, \bar{x}^{(B)}\}$ and sort $\mathcal{M}$ ascending. The 95% CI is the empirical quantile interval

$$\mathrm{CI}_{95\%} = \left[\, Q_{0.025}(\mathcal{M}),\ Q_{0.975}(\mathcal{M}) \,\right].$$

Plots show the sample mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ with error bars equal to $\mathrm{CI}_{95\%}$.

E.2 Workload Generation

We use two interaction patterns that our prototype supports:

1. **Simple transfer:** a direct cross-chain asset movement (source L1 $\rightarrow$ destination L1) that triggers audit-tag emission and bridging to the audit chain.

2. **Escrow settlement:** funds deposited into a neutral escrow on the source chain and released upon satisfaction of a contract predicate (e.g., multi-party acknowledgement), followed by cross-chain completion and audit-tag emission.

Unless otherwise noted, workloads are generated in closed loop: the next transaction is issued after the previous one reaches the configured confirmation depth and the corresponding Atag is persisted on the audit chain.

E.3 Axelar Confirmation Depth and Provenance Metrics

We vary the Axelar confirmation depth $d \in \{1, 2, 4, 8\}$, i.e., the number of source-chain block confirmations required before a cross-chain message is accepted for relay. We measure:

- **End-to-end latency:** time from the source-chain transaction commit to the Atag being committed on the audit chain.

- **Replay-safety:** we verify that deduplication on $(\texttt{chain\_id}, \texttt{txid})$ (and a monotonic $\texttt{seq}$ if present) rejects replays across reorgs or redundant relays.

This isolates the latency vs. replay-safety trade-off induced by d without relying on timestamps for correctness.

E.4 Auditor-Only Linkability (AOL) Evaluation Protocol

We construct ground-truth pairs across chains for equality testing under auditor-only linkability:

1. **Same-entity pairs:** multiple cross-chain transactions produced by the *same* user (same master key) across heterogeneous chains, yielding audit tags with equality on the auditor-side link ciphertexts.

2. **Different-entity pairs:** transactions from *distinct* users (different master keys), expected to test unequal.

For each pair (i, j) we run the auditor's equality test

$$\mathsf{ET.Test}(\mathsf{tk}, \mathsf{CT}_i^{\mathrm{link}}, \mathsf{CT}_j^{\mathrm{link}}) \in \{\mathsf{equal}, \mathsf{unequal}\}.$$

We report:

- **AOL correctness:** true-positive rate (TPR) on same-entity pairs and true-negative rate (TNR) on different-entity pairs.

- **Throughput/latency:** number of $\mathsf{ET.Test}$ operations per second and per-call latency on the auditor machine, using the same bootstrap method (Section E.1) to produce 95% CIs.

Privacy of AOL (i.e., public-side unlinkability without tk) is covered by the formal proofs and is not re-evaluated empirically.

E.5 Measurement Conventions

All clocks are taken from the local measurement host unless otherwise specified. For each configuration, we perform $n = 10$ independent runs; we randomize transaction nonces and chain-side gas parameters to avoid scheduler artifacts. Reported metrics are the sample mean with 95% CIs computed via the bootstrap in Section E.1.

References

[1] Shahla Atapoor, Karim Baghery, Daniele Cozzo, and Robi Pedersen. Vss from distributed zk proofs and applications. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 405–440. Springer, 2023.

[2] Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch Lafuente. Sok: lending pools in decentralized finance. In *International Conference on Financial Cryptography and Data Security*, pages 553–578. Springer, 2021.

[3] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. A survey on blockchain interoperability: Past, present, and future trends. *Acm Computing Surveys (CSUR)*, 54(8):1–41, 2021.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE symposium on security and privacy (SP)*, pages 315–334. IEEE, 2018.

[5] Benedikt Bünz, Lucianna Kiffer, Loi Luu, and Mahdi Zamani. Flyclient: Super-light clients for cryptocurrencies. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 928–946. IEEE, 2020.

[6] Vitalik Buterin, Jacob Illum, Matthias Nadler, Fabian Schär, and Ameen Soleimani. Blockchain privacy and regulatory compliance: Towards a practical equilibrium. *Blockchain: Research and Applications*, 5(1):100176, 2024.

[7] Stefanos Chaliasos, Jens Ernstberger, David Theodore, David Wong, Mohammad Jahanara, and Benjamin Livshits. {SoK}: What don't we know? understanding security vulnerabilities in {SNARKs}. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 3855–3872, 2024.

[8] Yourong Chen, Hao Chen, Yang Zhang, Meng Han, Madhuri Siddula, and Zhipeng Cai. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confidence Computing*, 2(2):100048, 2022.

[9] Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX symposium on networked systems design and implementation (NSDI 17)*, pages 259–282, 2017.

[10] Yvo G Desmedt and Yair Frankel. Homomorphic zero-knowledge threshold schemes over any finite abelian group. *SIAM journal on Discrete Mathematics*, 7(4):667–679, 1994.

[11] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 427–438. IEEE, 1987.

[12] Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ecdsa with fast trustless setup. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 1179–1194, 2018.

[13] Jens Groth. On the size of pairing-based non-interactive arguments. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 305–326. Springer, 2016.

[14] Yihao Guo, Minghui Xu, Xiuzhen Cheng, Dongxiao Yu, Wangjie Qiu, Gang Qu, Weibing Wang, and Mingming Song. {zkCross}: A novel architecture for {Cross-Chain}{Privacy-Preserving} auditing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6219–6235, 2024.

[15] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. In *Network and distributed system security symposium*, 2017.

[16] Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. {BlockSci}: Design and applications of a blockchain analysis platform. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2721–2738, 2020.

[17] Russell WF Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling private payments without trusted setup. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–48, 2019.

[18] Junkai Liang, Daqi Hu, Pengfei Wu, Yunbo Yang, Qingni Shen, and Zhonghai Wu. Sok: Understanding zk-snarks: The gap between research and practice. *arXiv preprint arXiv:2502.02387*, 2025.

[19] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[20] Fieke Miedema, Kelvin Lubbertsen, Verena Schrama, and Rolf Van Wegberg. Mixed signals: Analyzing {Ground-Truth} data on the users and economics of a

bitcoin mixing service. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 751–768, 2023.

[21] Neha Narula, Willy Vasquez, and Madars Virza. {zkLedger}:{Privacy-Preserving} auditing for distributed ledgers. In *15th USENIX symposium on networked systems design and implementation (NSDI 18)*, pages 65–80, 2018.

[22] Wangze Ni, Yiwei Zhao, Pengze Chen, Lei Chen, Peng Cheng, and Chen Jason Zhang. Cmixing: An efficient coin mixing platform to enhance anonymity in cryptocurrency transactions. *Proceedings of the VLDB Endowment*, 17(12):4405–4408, 2024.

[23] Minhao Qiao, Xuanchang Chen, Yangping Zhou, and PY Mok. Blockchain-driven innovation in fashion supply chain contractual party evaluations as an emerging collaboration model. *Blockchain: Research and Applications*, 6(2):100266, 2025.

[24] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.

[25] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[26] Gang Wang, Qin Wang, and Shiping Chen. Exploring blockchains interoperability: A systematic survey. *ACM Computing Surveys*, 55(13s):1–38, 2023.

[27] Tiancheng Xie, Jiaheng Zhang, Zerui Cheng, Fan Zhang, Yupeng Zhang, Yongzheng Jia, Dan Boneh, and Dawn Song. zkbridge: Trustless cross-chain bridges made practical. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3003–3017, 2022.

[28] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J Knottenbelt. Sok: Communication across distributed ledgers. In *International Conference on Financial Cryptography and Data Security*, pages 3–36. Springer, 2021.

[29] Alexei Zamyatin, Dominik Harz, Joshua Lind, Panayiotis Panayiotou, Arthur Gervais, and William Knottenbelt. Xclaim: Trustless, interoperable, cryptocurrency-backed assets. In *2019 IEEE symposium on security and privacy (SP)*, pages 193–210. IEEE, 2019.