Privacy-preserving Web3 recovery

Abstract. Guardian-based account recovery provides an important safeguard for cryptocurrency and Web3 users by enabling trusted third parties, known as *guardians*, to assist in recovering lost or compromised accounts. While effective in restoring access, current guardian-based mechanisms expose sensitive recovery actions publicly on-chain, revealing both the identities of guardians and the fact that recovery is in progress. Such visibility invites targeted attacks, opportunistic exploitation, reputational harm, and regulatory scrutiny.

In this paper, we study the problem of achieving *indistinguishability* between regular transactions and guardian-initiated recoveries in account-based blockchains. We formalize the inherent Δ-delay challenge, where guardian transactions require a waiting period to allow owner challenges, and show how this timing difference leaks transaction type to external observers. We propose practical solutions that combine threshold cryptography, threshold ring signatures, linkable ring signatures, and off-chain obfuscation layers to conceal recovery origin while preserving security. Our modular design enables robust governance through misbehavior detection and mitigates timing-based inference attacks. We discuss open challenges for extending the protocol to work with confidential and privacy-preserving asset frameworks.

We conclude by presenting a protocol framework that achieves both privacy and safety in guardian-based recovery, along with future directions for implementing trustless off-chain obfuscation channels and integrating zero-knowledge proofs for guardian set membership.

1 Introduction

In cryptocurrency and Web3 ecosystems, users manage significant digital assets through private keys. Loss or compromise of these keys typically results in permanent asset loss, underscoring the critical need for robust account recovery mechanisms. One widely adopted approach is *guardian-based recovery*, in which a set of trusted individuals or institutions, called "guardians", can authorize restoration of account access.

Guardian-based recovery has proven effective and practical. However, existing implementations suffer from serious privacy drawbacks. The identities of guardians are publicly visible on-chain and recovery actions themselves are transparent. Even if one attempted to obfuscate these, the timing differences between normal and recovery transactions would allow external observers to trivially distinguish them. This transparency introduces multiple risks, including targeted attacks against guardians, opportunistic exploitation of vulnerable accounts, reputational damage for organizations, and potential regulatory scrutiny based on

misinterpreted recovery events. High-profile users may also become targets for harassment or exploitation if their recovery operations are publicly visible.

Our objective is to design a recovery protocol that preserves *privacy* while maintaining *security* and *usability*. Specifically, the protocol should make recovery transactions indistinguishable from regular transactions, hide guardian identities on-chain and retain the ability for owners to challenge unauthorized recoveries within a specified delay period Δ with minimal overhead on the overall system or impact on the user experience.

Motivation. A privacy-preserving guardian recovery protocol addresses several critical concerns. First, it protects guardian identities by concealing which guardians participate in a recovery, reducing the risk of coercion, bribery, targeted hacking, or legal pressure. Second, it prevents opportunistic attacks by eliminating on-chain signals that an account is undergoing recovery. This is particularly important in smart contract-based recovery protocols, which are highly sensitive to the emission of explicit recovery events such as KELP [3]; obfuscating such events strengthens the security of the underlying recovery logic. Third, it improves reputation management for organizations, as it prevents visible recoveries from triggering market instability, loss of user confidence, or negative publicity, and it enhances personal security by shielding high-net-worth individuals and public figures from unwanted attention tied to recovery operations. Finally, it mitigates regulatory and compliance risks by avoiding unnecessary audits or investigations that may be triggered by publicly visible recovery activity, especially in tightly regulated environments.

Our contributions. This work is, to our knowledge, the first to address privacy-preserving guardian-based account recovery in detail. While a recent SoK paper [6] systematizes Web3 recovery mechanisms and identifies privacy-preserving recovery as a critical open problem, it does not propose concrete solutions or analyze the fundamental privacy challenges that arise from timing asymmetries in guardian-based protocols. We make the following contributions to fill this gap.

First, we formalize the indistinguishability problem in guardian-based recovery, with particular emphasis on the inherent Δ-delay challenge: guardian transactions require a waiting period to allow owner challenges, but this very delay leaks information about transaction type. We show how this timing asymmetry can be exploited by observers to infer recovery events. Second, we propose a modular protocol design that combines threshold cryptography, threshold ring signatures, linkable ring signatures, and off-chain obfuscation layers to conceal recovery origin while preserving security. Our design ensures that guardian authorizations remain anonymous, recovery operations cannot be detected by timing analysis until finalization, and owners retain the ability to challenge malicious recoveries during the delay period. Third, we analyze alternative approaches, including commit–reveal schemes, uniform delay enforcement, reversible execution models, and hypothetical conditional execution proofs, assessing their privacy guarantees, usability trade-offs, and compatibility with existing blockchain architectures. Finally, we explore governance mechanisms for detecting and pun-

ishing guardian misbehavior, and we discuss open challenges for extending our protocol to work with confidential and privacy-preserving asset frameworks.

Related Work. Guardian-based recovery has been explored in various forms, most notably in social recovery wallets such as those discussed in Ethereum's EIP-4337 account abstraction proposal [5] and implemented in solutions like Argent [1] and Safe [21]. These systems typically rely on multisignature schemes or explicit guardian transaction calls, both of which are fully transparent on-chain. The standard approach uses t-of-n multisig, where t guardians must each post a signature on-chain; this is efficient but reveals which guardians participated in the recovery. More recently, threshold cryptography and distributed key generation (DKG) have been applied to reduce guardian visibility and provide robustness against guardian compromise [10]. However, none of these approaches hide the *occurrence* of a recovery or make recovery indistinguishable from normal transactions.

Privacy-preserving techniques developed for payment anonymity provide useful primitives but are not directly applicable to recovery contexts. Ring signatures, introduced by Rivest, Shamir, and Tauman [19], provide signer anonymity by mixing a signer with a set of decoys, making it impossible to determine which ring member signed a message. Linkable ring signatures [13] extend this to prevent double-spending by allowing detection of double-signatures by the same signer without revealing identity. These techniques have been adopted in privacy-focused cryptocurrencies (Monero, Zcoin) and recently studied for blockchain voting systems [18]. Threshold ring signatures [4], which combine threshold cryptography with ring signatures, enable threshold-based signing with signer anonymity. Beyond ring signatures, zero-knowledge proofs have enabled stronger privacy guarantees: Zerocash [2] and Confidential Transactions [15] hide transaction amounts using range proofs and zero-knowledge arguments [11]. However, applying these techniques to recovery requires addressing the unique constraint that owners must be able to detect and challenge unauthorized recoveries—a requirement absent in payment privacy systems.

Ethereum's account abstraction framework (EIP-4337) [9] and modular smart contract wallets like Safe [22] enable flexible recovery logic implemented as contract code rather than protocol-level features. This provides an ideal deployment target for our protocol, as all recovery logic can be implemented in Solidity without requiring consensus-level changes.

2 Cryptographic Preliminaries

In this section, we review the cryptographic primitives used in our protocol design. These include threshold signatures [23,10], linkable ring signatures [13], and threshold ring signatures [4]. We present their formal definitions, algorithms, and security properties, as well as their relevance to privacy-preserving guardian-based recovery.

2.1 Threshold Signatures

A (t, n)-*threshold signature scheme* enables n parties to share the ability to sign messages such that any coalition of size at least t can produce a valid signature, but coalitions smaller than t cannot [23].

Formally, a threshold signature scheme consists of four algorithms:

- $\mathsf{Setup}(1^\lambda) \to pp$: Generates public parameters pp given a security parameter λ.
- $\mathsf{KeyGen}(pp) \to (sk_i, pk_i)$: For each signer $i \in \{1, \ldots, n\}$, generates a secret key sk_i and public key pk_i.
- $\mathsf{Sign}(pp, m, \{sk_i\}_{|S| \geq t}) \to \sigma$: Given public parameters pp, a message m, and at least t secret keys from the set S, outputs a signature σ.
- $\mathsf{Verify}(pp, m, \sigma) \to \{0, 1\}$: Verifies that σ is a valid signature on m under the public parameters.

Informally, a threshold signature scheme should satisfy the following security properties [10]: **Correctness**: Any set of t or more honest signers can produce a signature that verifies successfully. **t-Unforgeability**: No adversary controlling fewer than t secret keys can forge a valid signature. **Threshold Anonymity**: The signature does not reveal which subset of signers produced it, preventing identification of individual contributors.

Threshold signatures are used in our recovery protocol to allow guardian sets to authorize recovery without revealing which specific guardians participated.

2.2 Linkable Ring Signatures

A *linkable ring signature* allows a signer to produce an anonymous signature within a set (ring) of possible public keys, while enabling anyone to determine if two signatures were produced by the same (anonymous) signer [13].

A linkable ring signature scheme comprises:

- $(sk, pk) \leftarrow \mathsf{KeyGen}()$: Generates a key pair for the signer.
- $\sigma \leftarrow \mathsf{Sign}(sk, \vec{pk}, m)$: Signs message m using secret key sk and public key list $\vec{pk}$ representing the ring.
- $b \leftarrow \mathsf{Verify}(\vec{pk}, m, \sigma)$: Verifies that σ is a valid signature for m under the ring $\vec{pk}$.
- $b \leftarrow \mathsf{Link}((\vec{pk}_1, m_1, \sigma_1), (\vec{pk}_2, m_2, \sigma_2))$: Determines whether σ_1 and σ_2 were created by the same signer within their respective rings.

Informally, a linkable ring signature scheme should satisfy the following security properties [13]: **Anonymity**: The signature hides the actual signer among the ring members. **Unforgeability**: Only a ring member with a valid secret key can produce a signature. **Linkability**: Signatures from the same signer can be linked, enabling detection of repeated actions without revealing identity.

In our design, linkable ring signatures enable owners to challenge guardian recoveries and detect repeated misuse by the same guardian without exposing guardian identities.

2.3 Threshold Ring Signatures

A (T, n)-*threshold ring signature* combines threshold and ring signature properties: at least T of n ring members must sign a message, but the signature hides which specific members participated [4,12].

Formally, such a scheme includes:

- $\mathsf{Setup}(\lambda) \to \mathsf{pp}$: Outputs public parameters.
- $\mathsf{KeyGen}(\mathsf{pp}) \to (sk, pk)$: Generates key pairs for ring members.
- $\mathsf{ThSign}(\mathsf{pp}, S, T, \{(sk_i, pk_i)\}_{i \in I}, m) \to \sigma$: Produces a signature σ for message m with at least T signers from the public key set S.
- $\mathsf{Verify}(\mathsf{pp}, S, T, m, \sigma) \to \{0, 1\}$: Verifies that σ is valid and was produced by at least T members of S.

Informally, a threshold ring signature scheme should satisfy the following security properties [4,12]: **Correctness**: Any T valid signers from S can produce a signature that verifies. **T-Unforgeability**: No coalition of fewer than T signers can forge a valid signature. **T-Anonymity**: The signature hides the identities of the actual signers among all n ring members.

Threshold ring signatures are particularly well-suited for guardian-based recovery because they allow the recovery authorization to be both threshold-secure and anonymous, thereby concealing which guardians participated while still enforcing a threshold policy.

3 Problem Description

We consider a blockchain account recovery setting with the following conditions:

- An account A has an owner public key pk_O and one or more guardian public keys $\{pk_{G_1}, pk_{G_2}, \dots\}$.
- The blockchain state includes a delay parameter Δ, representing the mandatory time delay[1] for guardian-initiated recoveries.
- Transactions signed by the owner take effect immediately, while guardian-initiated transactions become effective only after Δ time units (or blocks). During this period, if an owner who monitors the blockchain detects an unauthorized recovery attempt (e.g., a malicious guardian attempting to use the recovery protocol to transfer the funds to an account controlled by an adversary), he/she may challenge it by invoking the appropriate smart contract function and revert the funds back under the owner's control.

Goals. The protocol must satisfy two primary requirements:

1. **Indistinguishability:** External observers must be unable to distinguish an owner-initiated transaction from a guardian-initiated recovery transaction, even in the presence of timing or metadata differences.

[1] This parameter is typically set by the wallet software or chosen by the user. See [7] for an analysis of the parameter choice in wallet security.

2. **Security:** The owner must be able to challenge and cancel unauthorized guardian-initiated recoveries within the delay period Δ.

More formally, we require that blockchain observers cannot distinguish immediate execution transaction from pk_O or a delayed execution transaction from guardians $\{pk_{G_i}\}$.

Fundamental Challenge. The inherent timing difference between immediate and delayed execution leaks transaction origin:

$$\text{Immediate execution (Owner)} \neq \text{Delayed execution (Guardian)}$$

This timing-based leakage must be mitigated while preserving the owner's challenge capability.

3.1 Potential Solutions

We now explore a series of increasingly sophisticated approaches, analyzing their strengths and limitations to motivate our solution.

Baseline: Uniform Delay. The simplest approach is to apply a uniform delay Δ to *all* transactions, regardless of origin. This would achieve perfect indistinguishability by design: owner and guardian transactions would become observationally identical. However, usability suffers dramatically. Every normal transaction incurs a Δ-delay penalty, severely degrading user experience. A variant using randomized delay offers marginally better usability but introduces unpredictability that may not be acceptable to users who need consistent performance. The fundamental trade-off (perfect indistinguishability at the cost of severe usability loss) makes this approach impractical.

Commit-Reveal with Off-Chain Hints. To preserve usability, consider a commit-reveal scheme: both owner and guardian transactions post an identical cryptographic commitment on-chain,

$$\text{Commit} = \mathsf{Hash}(\text{receiver}, \text{amount}, \text{nonce}),$$

hiding the transaction parameters. Owner transactions would immediately reveal and execute; guardian transactions would reveal after Δ. The limitation is that the reveal timing still leaks origin: an observer sees immediate reveals (owner) vs. delayed reveals (guardian), defeating indistinguishability. The key insight is: what if the sender privately reveals the commitment opening to the receiver *off-chain* before any on-chain reveal? The receiver could then verify and consider funds available immediately, without waiting for public on-chain confirmation. This preserves usability while deferring the distinguishing event.

Uniform Finalization with Off-Chain Usability. Extend the commit-reveal idea: both owner and guardian transactions post an identical-form commitment, but neither reveals immediately on-chain. Instead: (i) the sender privately delivers the commitment opening to the receiver via a secure off-chain channel; (ii) the receiver verifies the opening matches the on-chain commitment and considers the transaction complete for practical purposes (e.g., for use in off-chain or

Layer 2 contexts); (iii) both transaction types finalize on-chain *uniformly* after delay Δ, at which point the opening is publicly revealed. To external observers, all on-chain transactions now look identical (uniform commitments and uniform reveal timing) until finalization. Owner and guardian transactions are indistinguishable. Usability is also preserved: receivers can use funds immediately in off-chain or Layer 2 contexts without waiting for on-chain confirmation. This approach still does not fully address the attacker's ability to distinguish if they observe off-chain communication, nor does it hide guardian identities in the recovery authorization itself. These gaps motivate the addition of cryptographic anonymity primitives.

Our Solution: Off-Chain Obfuscation + Cryptographic Anonymity. Our approach combines the uniform finalization strategy above with anonymous authorization primitives. At a high level, the protocol flow is as follows:

1. Both owner and guardian transactions post identical commitments on-chain.
2. Guardian-initiated recovery authorizations use *threshold ring signatures* to hide which guardians (or even that guardians) participated.
3. The owner privately delivers the commitment opening off-chain for immediate usability.
4. After delay Δ, the opening is revealed uniformly on-chain.
5. To challenge a malicious recovery, the owner uses a *linkable ring signature* to anonymously signal that a recovery is unauthorized, enabling later detection of repeated misbehavior by the same guardian.

By combining (i) uniform on-chain transaction structure and timing, (ii) threshold ring signatures for anonymous guardian authorization, and (iii) off-chain private reveals for immediate usability, we achieve indistinguishability to external observers while maintaining the owner's challenge capability and enabling guardian misbehavior detection. This synthesis of techniques directly addresses the limitations of each earlier approach.

While we focus on recovery-specific design choices in this section, we note that many general privacy primitives from the broader cryptography and blockchain literature, such as stealth addresses, zero-knowledge proofs, ring signatures, and confidential transactions, have been considered for privacy-preserving systems. We analyze why these general-purpose techniques, while useful in isolation, are insufficient for addressing the specific indistinguishability challenge in guardian-based recovery. A detailed comparison is provided in Appendix A.

3.2 Cryptographic Structures

Here we discuss how the primitives introduced in Section 2 integrate into our solution and address specific aspects of the indistinguishability and security goals.

Commitment Schemes for Uniform On-Chain Appearance. The foundation of our approach rests on identical commitment structures posted by both owner and guardian transactions:

$$\text{Commit} = \mathsf{Hash}(\text{receiver}, \text{amount}, \text{nonce}).$$

To an external observer, this commitment reveals no information about the transaction origin (owner vs. guardian). Both transaction types produce commitments of the same form and are posted to the blockchain in the same manner. This uniform structure is essential for achieving indistinguishability (IND) in the initial on-chain phase, before any reveal occurs. The commitment also binds the sender to a specific set of transaction parameters, preventing substitution or equivocation.

Threshold Ring Signatures for Guardian Anonymity. While the commitment hides transaction parameters, the authorization signature could still leak guardian identities if guardians sign directly and visibly. To prevent this, guardian-initiated recoveries are authorized using a (T, n)-threshold ring signature, where T is the recovery threshold and n is the total number of guardians. This primitive serves two critical roles:

- **Threshold Enforcement (RR):** The T-Unforgeability property of threshold ring signatures ensures that fewer than T guardians cannot produce a valid authorization, satisfying Recovery Robustness. No coalition of size $T-1$ can forge a recovery.
- **Guardian Privacy (GP):** The T-Anonymity property hides which specific subset of T guardians signed the recovery. External observers see a threshold ring signature on the commitment but cannot identify the participating guardians. This protects guardians from targeted attacks, bribery, or coercion based on their participation in a recovery.

By combining guardians into a ring and proving only that a threshold was met (without revealing who), threshold ring signatures decouple the recovery authorization from individual guardian identities.

Linkable Ring Signatures for Challenge and Misbehavior Detection. During the delay period Δ, if the owner detects an unauthorized recovery, they must challenge it. The owner could simply sign a revocation message directly; however, this would reveal the owner's involvement and potentially leak information. Instead, the owner uses a linkable ring signature to challenge:

$$\sigma_{\text{challenge}} \leftarrow \mathsf{Sign}_{\text{LRS}}(sk_O, \text{PKring}, \text{challenge\_msg}),$$

where PKring includes the owner's public key and possibly other public keys to maintain anonymity.

This primitive enables two key properties:

- **Owner Challenge Capability (OCC):** The owner can challenge any recovery attempt and revert it within Δ. The LRS authenticates the challenge (only the legitimate owner can produce a valid signature from sk_O) while the ring structure hides the owner's involvement from external observers. This allows the owner to immediately cancel any unauthorized recovery without revealing their presence on-chain.

– **Misbehavior Linkability (ML):** On a single unauthorized recovery attempt, the owner's challenge proves the recovery is invalid and reverts it immediately. However, the challenge does not identify which guardian initiated the recovery: the TRS remains anonymous (we do not know which T guardians out of n signed), and the challenge LRS proves only that the owner signed a message, not which guardian misbehaved. If, however, the same guardian initiates *multiple* unauthorized recovery attempts against the same account, the owner challenges each one. The linkability property of LRS allows the smart contract to detect when multiple challenges correspond to the same signer (the owner), and by correlating these challenges with their recovery commits, to infer repeated misbehavior by the same guardian. The contract can then flag this guardian for slashing or exclusion from future guardian sets, enabling accountability without deanonymization. This design balances privacy (single misbehavior does not expose the guardian's identity) with accountability (persistent attackers are detected and punished).

Linkable ring signatures thus enable accountability (repeated misbehavior can be detected and punished via correlating linked challenges) while preserving anonymity (the responsible guardian's identity is never revealed).

Interplay of the Three Primitives. Together, these primitives form a cohesive system:

1. Commitments ensure on-chain indistinguishability of owner vs. guardian transactions until reveal.
2. Threshold ring signatures authorize guardian recoveries anonymously while enforcing the threshold policy.
3. Linkable ring signatures allow the owner to challenge anonymously and enable detection of repeated misbehavior.
4. Off-chain private reveals (delivered via a secure channel) provide immediate usability without compromising on-chain indistinguishability.

This modular design ensures that no single primitive bears the full burden of achieving security and privacy. Instead, each primitive addresses a specific aspect of the problem, and their combination yields a robust protocol.

3.3 Smart Contract Model

The recovery protocol is implemented as a smart contract that manages account state, enforces authorization rules, and coordinates owner and guardian actions, as shown in Figure 1.

State Transitions. The contract enforces the following rules:

– **Owner Spend:** If initiate receives an owner signature with type "owner", the commitment is posted. After Δ blocks, any party may call reveal to finalize and execute the transfer.

Smart Contract State and Interface

State Variables:

- C.owner: Current owner public key pk_O (authorizes immediate spending and challenges).
- C.guardians: Guardian public keys $\{pk_{G_1}, \ldots, pk_{G_n}\}$ or Merkle commitment (reduces storage, hides list on-chain).
- C.threshold: Recovery threshold T (minimum guardians required).
- C.delay: Delay parameter Δ (blocks/time units during which owner can challenge).
- C.pendingRecovery: In-flight recovery (commit, σ_{TRS}, blockNum) or $\perp$.
- C.challengeLog: List of linkable ring signatures for misbehavior detection.

Public Functions:

initiate(**commit**, σ)

- The contract receives only the commitment and signature, with no explicit type field.
- First, attempt to verify σ as an owner signature under pk_O. If valid, post commitment (owner transaction).
- Otherwise, attempt to verify σ as a valid threshold ring signature under C.guardians and threshold C.threshold. If valid, set C.pendingRecovery $\leftarrow$ (commit, σ, now) (recovery transaction).
- If both verifications fail, reject the transaction.

reveal(**receiver**, **amount**, **nonce**)

- Verify Hash(receiver, amount, nonce) = commit.
- If C.pendingRecovery $\neq \perp$: require now $\geq C$.pendingRecovery.blockNum + Δ.
- Transfer funds or rotate owner key.

challenge($\sigma_{\textbf{challenge}}$)

- Require C.pendingRecovery $\neq \perp$ and now $< C$.pendingRecovery.blockNum + Δ.
- Verify $\sigma_{\text{challenge}}$ is a valid linkable ring signature from owner's key.
- Clear C.pendingRecovery; append $\sigma_{\text{challenge}}$ to C.challengeLog.
- If Link($\sigma_{\text{challenge}}, \sigma'$) = true for $\sigma' \in C$.challengeLog, flag guardian for repeated misbehavior.

Fig. 1: Smart contract state variables, public functions, and their semantics for privacy-preserving guardian recovery.

- **Guardian Recovery:** If initiate receives a valid threshold ring signature σ_{TRS} (authenticating at least T guardians) with type "recovery", the recovery is marked pending in C.pendingRecovery. After Δ blocks, reveal can be called to finalize the key rotation or fund transfer. If challenge is called before Δ blocks have elapsed, the pending recovery is cancelled and the owner regains control.
- **Challenge:** Upon receiving a valid linkable ring signature challenge from the owner, C.pendingRecovery is cleared, and the signature is appended to C.challengeLog. If multiple challenges link to the same signer (using the Link operation), the contract can flag or penalize that guardian.

Security Guarantees. This contract design ensures:

- **Recovery Robustness (RR):** Only a coalition of at least T guardians can authorize a recovery (T-Unforgeability of threshold ring signatures).
- **Guardian Privacy (GP):** The guardians' identities are hidden in the threshold ring signature and in the challenge log, as the challenge is a linkable ring signature over a broader set.
- **Owner Challenge Capability (OCC):** The owner has an exclusive, unforgeable right to challenge via their private key; challenges must be submitted within Δ blocks.
- **Misbehavior Linkability (ML):** Repeated challenges linked to the same guardian (via Link) enable punishment mechanisms while preserving the guardian's anonymity.

3.4 Discussion and Design Trade-offs

Delay parameter Δ: The delay must be long enough for the owner to detect and challenge malicious recoveries, yet short enough to permit timely legitimate recovery [7]. We mitigate this tension via immediate off-chain usability: the receiver obtains the commitment opening privately and considers funds available in off-chain or Layer 2 contexts, decoupling user experience from on-chain finality and allowing Δ to be tuned for security rather than usability.

Guardian anonymity vs. registry: Threshold ring signatures hide participating guardians but not the guardian set itself, which must be known for verification. Hiding the guardian list entirely would require zero-knowledge proofs that verify T valid guardians participated without revealing the set.

Lost vs. stolen keys: [14] Our protocol handles key loss and key theft asymmetrically. In the *lost-key* scenario, if a guardian has knowledge that the owner no longer possesses their key, the guardian can safely initiate a recovery without fear of challenge (the owner simply cannot challenge), so the recovery proceeds after Δ. In the *stolen-key* scenario, the owner and the attacker are in a race: if the owner detects the unauthorized recovery and challenges within Δ, the owner wins and the recovery is cancelled; if the attacker can proceed undetected for Δ blocks, the attacker wins and the recovery succeeds. The protocol does not cryptographically distinguish between these scenarios; instead, it reflects the reality that only the owner and potential attackers know whether the key is truly lost or compromised. This asymmetry is inherent to any guardian-based recovery mechanism [7]: if a guardian knows with certainty that the owner is unavailable, recovery cannot be prevented without additional out-of-band authentication (e.g., legal attestation, biometric verification, or time-locked collateral).

Side-channels: If recovery transactions always transfer the entire balance while normal transactions are arbitrary-valued, transaction amounts become distinguishing. This can be mitigated by allowing partial recoveries, UTXO-like abstractions, or range proofs, which are orthogonal improvements, compatible with our protocol. Similarly, the private off-chain delivery of commitment openings assumes a secure authenticated channel (direct message, Signal, TOR, or a relay).

4 Solution Design

In this section, we formally define the threat model, specify the security goals our protocol achieves, and present the detailed protocol flow. We show how threshold ring signatures, linkable ring signatures, and commitment schemes work together to achieve indistinguishability while preserving security and usability.

4.1 Threat Model

We assume a powerful network and blockchain observer but an uncompromised owner key and standard cryptographic hardness. The adversary can observe all on-chain activity, including transaction timing, structure, and sender/recipient metadata, and can corrupt up to $t-1$ guardians who may collude but still cannot satisfy the t-of-n recovery threshold. The adversary can also delay, drop, or reorder off-chain messages between participants. The main goals are to (i) distinguish recovery transactions from normal transactions, (ii) identify which guardians participated in a recovery, or (iii) execute unauthorized recovery with fewer than t corrupted guardians or by bypassing the owner's challenge. We assume that the owner's key is not stolen (i.e., not in the possession of an attacker), though it may be lost and thus unavailable; in the lost-key scenario, the owner cannot challenge and the protocol allows recovery to proceed. We further assume the underlying cryptographic primitives (threshold ring signatures, linkable ring signatures, hash functions) are secure against polynomial-time adversaries, while attacks relying on key theft by the adversary are out of scope.

4.2 Security Goals

Our protocol aims to satisfy the following goals:

Indistinguishability (IND): Owner-initiated and guardian-initiated transactions must be indistinguishable to external observers until the finalization reveal, preventing inference of transaction type from timing or structure.

Guardian Privacy (GP): The identities of participating guardians remain hidden from external observers, preventing targeted attacks.

Recovery Robustness (RR): No coalition of fewer than t guardians can authorize a recovery.

Owner Challenge Capability (OCC): The owner can detect and challenge any unauthorized recovery within the delay period Δ.

Misbehavior Linkability (ML): Repeated malicious recovery attempts by the same guardian can be linked and detected, enabling punishment without deanonymization.

4.3 Protocol Flow

The protocol consists of three main transaction types. We define the following notation: pk_O is the owner public key, $\text{PKG} = \{pk_{G_1}, \ldots, pk_{G_n}\}$ are the guardian public keys, t is the recovery threshold, and Δ is the challenge delay period.

Algorithm 1: Standard Owner Transaction.

1. Owner computes a commitment: com $\leftarrow \mathsf{Hash}(\text{receiver}, \text{amount}, \text{nonce})$.
2. Owner signs the commitment: $\sigma_O \leftarrow \mathsf{Sign}(sk_O, \text{com})$.
3. Owner posts (com, σ_O) to the blockchain.
4. Owner privately sends (receiver, amount, nonce) to the receiver off-chain.
5. After Δ blocks, any party calls reveal(receiver, amount, nonce) on-chain, finalizing the transaction.

Algorithm 2: Guardian Recovery Transaction.

1. A set $S \subseteq \text{PKG}$ of at least t guardians agrees to recover the account.
2. Guardians jointly compute: com $\leftarrow \mathsf{Hash}(\text{receiver}, \text{amount}, \text{nonce})$.
3. Guardians produce a threshold ring signature: $\sigma_{\text{TRS}} \leftarrow \mathsf{ThSign}(pp, \text{PKG}, t, \{sk_i\}_{i \in S}, \text{com})$.
4. Guardians post $(\text{com}, \sigma_{\text{TRS}})$ to the blockchain.
5. Guardians privately send (receiver, amount, nonce) to the receiver off-chain.
6. After Δ blocks (if unchallenged), any party calls reveal(receiver, amount, nonce), finalizing the recovery.

Algorithm 3: Owner Challenge.

1. The owner continuously monitors the blockchain (or a monitoring service acting on the owner's behalf).
2. If an unauthorized commitment appears (one the owner did not initiate), the owner detects it immediately, since all outgoing commitments from their account are visible to them.
3. Within the delay period Δ, the owner generates a linkable ring signature: $\sigma_{\text{LRS}} \leftarrow \mathsf{Sign}_{\text{LRS}}(sk_O, \text{PKring}, \text{challenge\_msg})$, where PKring includes pk_O and possibly other public keys to maintain some anonymity.
4. The owner posts $(\text{challenge\_msg}, \sigma_{\text{LRS}})$ to the blockchain.
5. The smart contract verifies σ_{LRS} and cancels the pending recovery.
6. For misbehavior detection, the contract logs $(\text{commit}, \sigma_{\text{LRS}})$ pairs in a challenge log. If the same owner posts multiple challenges that link via the Link operation (i.e., $\mathsf{Link}(\sigma_{\text{LRS}}, \sigma'_{\text{LRS}}) = \text{true}$) against distinct recovery commits from the same guardian set, the contract infers repeated misbehavior by a guardian in that set. This repeated pattern enables slashing or exclusion of the guardian from future guardian sets.

4.4 Security Analysis

We now summarize how the protocol flow achieves each goal:

Indistinguishability (IND): Algorithms 1 and 2 post identical-form commitments and delay the reveal uniformly to Δ blocks. To external observers, owner and guardian transactions are structurally indistinguishable until reveal. The timing is also uniform (both finalize after Δ), preventing timing-based side channels.

Guardian Privacy (GP): The threshold ring signature in Algorithm 2 hides the identity of the participating guardians via its T-Anonymity property. Combined with the challenge mechanism in Algorithm 3, which uses linkable ring signatures, individual guardian actions remain unlinkable to their identities.

Recovery Robustness (RR): The T-Unforgeability property of threshold ring signatures ensures that fewer than T guardians cannot produce a valid recovery authorization, even if they collude with the adversary.

Owner Challenge Capability (OCC): The account-based blockchain model ensures the owner observes all outgoing transactions from their account, allowing immediate detection of unauthorized recoveries. The linkable ring signature mechanism allows the owner to challenge anonymously and unforgeably, reverting any unauthorized recovery within Δ.

Misbehavior Linkability (ML): On each unauthorized recovery, the owner challenges via Algorithm 3 (posting an LRS-signed challenge message). On a single unauthorized recovery, this challenge immediately reverts the recovery but does not identify the responsible guardian. However, if the same guardian initiates repeated unauthorized recoveries against the same account, the contract logs each challenge and detects linkability via the Link operation. By correlating linked challenges with their corresponding recovery commits, the contract infers repeated misbehavior by a guardian in that set. This repeated pattern triggers slashing or exclusion from future guardian sets, enabling accountability without deanonymization. The design reflects the judgment that single attacks are detected and quickly cancelled (via owner monitoring within Δ), while persistent attackers are flagged and punished.

4.5 Discussion

The account-based blockchain model is central to our design. All transactions on the blockchain are publicly visible, but the owner has a critical asymmetry: they *know which account is theirs* and can immediately detect any recovery attempt posted to that account by monitoring their account address. External observers see the same on-chain transactions, but without knowing the account's ownership, they cannot distinguish whether a recovery is authorized or not—this is precisely the indistinguishability property we achieve. The owner's ability to detect and challenge unauthorized recoveries depends on active monitoring of their account (via a full node, indexing service, or monitoring service) and timely response within the delay period Δ.

Indistinguishability is preserved for external observers through a combination of factors: (i) commitments are uniform in structure regardless of transaction type, (ii) reveal timing is uniform across both transaction types (both finalize after Δ), and (iii) guardian authorization is anonymous via threshold ring signatures. These factors conceal recovery events from external observers while retaining the owner's ability to detect and challenge unauthorized attempts based on their knowledge of account ownership.

Off-chain delivery of commitment openings plays a crucial role in usability: receivers can immediately use funds in off-chain or Layer 2 contexts without waiting for on-chain finalization. This decouples user experience from the finalization delay Δ, allowing Δ to be tuned purely for security rather than usability.

The linkable ring signature mechanism for challenges achieves a subtle balance: it allows owners to challenge anonymously (preserving privacy even when responding to compromise), while enabling repeated misbehavior by a guardian to be detected and punished via the Link property and correlation with recovery commits. This means accountability for malicious guardians is maintained despite the anonymity of individual signatures—a single attack is detected and cancelled, while repeated attacks are flagged and punished without ever deanonymizing the guardian.

5 Conclusion

In this work, we addressed the privacy shortcomings of existing guardian-based account recovery mechanisms in blockchain and Web3 environments. Current implementations, while effective for restoring access to lost accounts, expose both guardian identities and ongoing recovery activities, creating opportunities for targeted attacks, opportunistic exploitation, reputational damage, and regulatory interference.

We formalized the problem of *indistinguishability* between normal and recovery transactions, identifying the inherent Δ-delay challenge that arises from the need to give owners time to challenge guardian-initiated recoveries. We demonstrated that this delay, if applied only to guardian transactions, allows external observers to infer transaction type from timing patterns. Our proposed modular protocol conceals transaction origin and maintain privacy without sacrificing security. The design enables robust governance through misbehavior detection and mitigates timing-based inference attacks while remaining compatible with standard account-based blockchain architectures. Off-chain commitment openings preserve usability while maintaining uniform on-chain transaction appearances until delayed finalization.

The protocol is designed as a drop-in smart contract overlay for existing account-based blockchains (Ethereum, Layer 2 systems, etc.) and requires no consensus-level changes. For on-chain costs, standard owner transactions cost approximately 34,000 gas (1.6× a standard ETH transfer), based on documented Ethereum gas costs for signature verification and hashing [24]. Guardian recovery transactions require threshold ring signature verification, which dominates costs at approximately 200,000 gas for typical (3,7) configurations; owner challenges cost 125,000–185,000 gas. Costs scale linearly with guardian set size; typical configurations ($n = 7$ to 11 guardians) remain practical [9].

Future directions include zero-knowledge guardian set verification to strengthen privacy against statistical analysis, adaptive delay parameters to defend against timing-based inference, and integration with privacy-preserving asset systems.

References

1. Argent: Argent: Smart contract wallet with social recovery. https://www.argent.xyz (2018), accessed: 2026-01-09
2. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from Bitcoin. In: 2014 IEEE Symposium on Security and Privacy. pp. 459–474. IEEE Computer Society Press (May 2014). https://doi.org/10.1109/SP.2014.36
3. Blackshear, S., Chalkias, K., Chatzigiannis, P., Faizullabhoy, R., Khaburzaniya, I., Kokoris-Kogias, E., Lind, J., Wong, D., Zakian, T.: Reactive key-loss protection in blockchains. In: Bernhard, M., Bracciali, A., Gudgeon, L., Haines, T., Klages-Mundt, A., Matsuo, S., Perez, D., Sala, M., Werner, S. (eds.) FC 2021 Workshops. LNCS, vol. 12676, pp. 431–450. Springer, Berlin, Heidelberg (Mar 2021). https://doi.org/10.1007/978-3-662-63958-0_34
4. Bresson, E., Stern, J., Szydlo, M.: Threshold ring signatures and applications to ad-hoc groups. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 465–480. Springer, Berlin, Heidelberg (Aug 2002). https://doi.org/10.1007/3-540-45708-9_30
5. Buterin, V., Weiss, Y., et al.: Eip-4337: Account abstraction using alt mempool. https://eips.ethereum.org/EIPS/eip-4337 (2021), ethereum Improvement Proposal 4337, accessed: 2026-01-09
6. Chatzigiannis, P., Chalkias, K., Kate, A., Mangipudi, E.V., Minaei, M., Mondal, M.: SoK: Web3 recovery mechanisms. Cryptology ePrint Archive, Report 2023/1575 (2023), https://eprint.iacr.org/2023/1575
7. Chatzigiannis, P., Wang, K.C., Arora, S.S., Minaei, M.: A composability analysis framework for Web3 wallet recovery mechanisms. In: Blanton, M., Enck, W., Nita-Rotaru, C. (eds.) 2025 IEEE Symposium on Security and Privacy. pp. 1531–1546. IEEE Computer Society Press (May 2025). https://doi.org/10.1109/SP61157.2025.00158
8. Chaum, D.: The dining cryptographers problem: Unconditional sender and recipient untraceability. Journal of Cryptology **1**(1), 65–75 (1988)
9. Ethereum Improvement Proposals: Eip-4337: Account abstraction using alt mempool. https://eips.ethereum.org/EIPS/eip-4337 (2021), accessed: 2026-01-09
10. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 295–310. Springer, Berlin, Heidelberg (May 1999). https://doi.org/10.1007/3-540-48910-X_21
11. Groth, J.: On the size of pairing-based non-interactive arguments. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 305–326. Springer, Berlin, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49896-5_11
12. Haque, A., Scafuro, A.: Threshold ring signatures: New definitions and post-quantum security. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part II. LNCS, vol. 12111, pp. 423–452. Springer, Cham (May 2020). https://doi.org/10.1007/978-3-030-45388-6_15
13. Liu, J.K., Wei, V.K.: Linkable ring signatures: Security models and new schemes. In: Proceedings of the 2004 International Conference on Information Security and Cryptology. LNCS, vol. 2971, pp. 614–630. Springer (2004)
14. Maram, D., Kelkar, M., Eyal, I.: Interactive multi-credential authentication. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. p. 408–422. CCS '24, Association for Computing Machinery,

New York, NY, USA (2024). https://doi.org/10.1145/3658644.3670378, https://doi.org/10.1145/3658644.3670378
15. Maxwell, G.: Confidential transactions. Ledger **1**, 1–18 (2016), available at https://elementsproject.org/features/confidential-transactions
16. Maxwell, G.: Confidential transactions. Ledger **1**, 1–18 (2016), available at https://elementsproject.org/features/confidential-transactions
17. Noether, S., Mackenzie, A., Lab, M.R.: Ring confidential transactions. Ledger **1**, 1–18 (2016), monero privacy protocol implementation
18. Petrakis, F., Magkos, E., Siavvas, M.A.: Blockchain based anonymous voting system using ring-signature. In: Proceedings of the 2nd International Conference on Advances in Computing, Communication and Control. pp. 1–8 (2020)
19. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Berlin, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_32
20. van Saberhagen, N.: Cryptonote v2.0. https://cryptonote.org/whitepaper.pdf (2013), accessed: 2026-01-09
21. Safe: Safe (formerly gnosis safe) smart contract wallets. https://safe.global (2018), accessed: 2026-01-09
22. Safe: Safe smart contract wallet architecture and design. https://docs.safe.global (2018), accessed: 2026-01-09
23. Shoup, V.: Practical threshold signatures. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 207–220. Springer, Berlin, Heidelberg (May 2000). https://doi.org/10.1007/3-540-45539-6_15
24. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger. https://ethereum.org/en/whitepaper/ (2014), accessed: 2026-01-09

A Alternative Approaches

While developing our solution, we considered various privacy-preserving techniques from broader cryptography and blockchain literature. In this section, we analyze whether and why existing privacy primitives—originally designed for payment anonymity, confidential transactions, and recipient privacy—fall short when applied to guardian-based recovery. This analysis motivates our integrated approach, which combines multiple techniques in a recovery-specific manner.

A.1 Stealth Addresses and One-Time Address Schemes

Stealth addresses, extensively used in privacy-oriented cryptocurrencies [20,17], generate a unique, unlinkable address for each transaction, ensuring that an outside observer cannot link multiple payments to the same recipient.

Applicability: One might hope that stealth addresses could hide the recovery recipient's identity or obscure the recovery transaction among normal payments by making all outputs look identical.

Limitation: Stealth addresses solve recipient anonymity but not sender privacy or transaction type inference. More critically, they do not address the fundamental timing asymmetry: guardian recovery transactions incur an enforced delay Δ before finalization, while normal owner transactions finalize after a uniform delay. An observer monitoring the blockchain sees this timing difference

and infers transaction type. Additionally, stealth addresses require the recipient's public spend and view keys to be on-chain or publicly known, which does not apply to the account owner (who is already public by virtue of the account-based model). Stealth addresses alone do not solve indistinguishability.

A.2 Encrypted Metadata and Content Hiding

Another approach embeds encrypted metadata within transactions, such as a PIN, secret string, or encrypted parameters known only to the owner. Recovery transactions could encrypt metadata under the owner's public key to theoretically hide recovery intent.

Applicability: If only recovery transactions carry this encrypted metadata, the owner could use its presence or absence to detect whether an outgoing transaction originated from a guardian.

Limitation: This introduces key management complexity without solving the timing problem. More fundamentally, if recovery transactions are the *only* ones carrying metadata, the metadata itself becomes a distinguishing feature. If all transactions must carry metadata (to provide cover), then the metadata provides no privacy benefit. The timing asymmetry remains completely unresolved: the fact that a transaction finalizes after Δ blocks still reveals it as a recovery. Encrypted metadata is a necessary condition for some applications, but alone it is insufficient for indistinguishability in the context of guardian recovery.

A.3 General Zero-Knowledge Proofs for Transaction Validity

Zero-knowledge proofs [2,11] can prove that a transaction satisfies one of several conditions without revealing which condition was met. For instance, a ZKP could prove that a transaction is cryptographically valid under *either* owner authorization *or* guardian threshold authorization, without leaking the logical origin.

Applicability: ZKPs could contribute to hiding the logical source (owner vs. guardian) by proving authorization without revealing the authorization path.

Limitation: While ZKPs can hide the *logical* origin, they cannot mask the *execution timing* asymmetry. The fundamental constraint is architectural: the security model requires that guardian transactions have a delay Δ to allow the owner time to challenge, whereas owner transactions finalize after the same delay uniformly. This timing difference is inherent to the security requirements, not a cryptographic choice. An observer tracking when transactions are posted to the mempool vs. when they are included and finalized can distinguish owner from guardian transactions regardless of ZKP proofs. In fact, adding ZKPs increases computational overhead without addressing the core timing leakage.

A.4 Ring Signatures for Sender Anonymity

Ring signatures [19] allow a signer to produce an anonymous signature within a set (ring) of possible signers. The signature proves that one of the ring members signed but does not reveal which.

Applicability: Ring signatures could theoretically hide the sender of a recovery authorization by proving that one of several accounts' guardians authorized it, without revealing which account.

Limitation: Ring signatures are designed for anonymity within a fixed, public ring. In the context of guardian recovery, the owner already has a known account address (accounts are not hidden by default in account-based blockchains). A ring signature that includes multiple owner accounts would itself be suspicious: why would an account post a ring signature if not to hide its identity? Furthermore, ring signatures do not address the timing leakage problem; they only hide the sender within a specified set. The timing difference between owner and guardian transactions remains observable. Ring signatures can be *part* of a solution (as in our use of threshold ring signatures), but they alone do not achieve indistinguishability for recovery.

A.5 Dummy Transactions and Mixing

The idea of creating multiple dummy transactions alongside the genuine one [8] has been explored in privacy contexts to obfuscate the real action among noise. Combined with ZKPs, one could generate k dummy recovery requests and prove via ZKP which one is genuine without revealing it.

Applicability: By generating multiple dummy recovery requests, an observer cannot infer which is real without solving the ZKP, potentially providing privacy.

Limitation: Dummy transactions impose severe practical overhead. Generating k dummy transactions, each with its own ZKP proof and on-chain footprint, multiplies costs by a factor of k. For a practical security level (e.g., $k = 100$), this is prohibitively expensive, increases blockchain congestion, and requires multiple guardian coordinations. Furthermore, the scheme still inherits the timing leakage problem: all dummy transactions must also incur the delay Δ to avoid distinguishability, further degrading usability. The computational and operational burden makes this approach impractical for most users.

A.6 Confidential Transactions and Amount Hiding

Confidential transactions [16], as implemented in systems like Confidential Assets, use range proofs and commitments to hide transaction amounts while maintaining auditability.

Applicability: If recovery transactions transfer the entire account balance but normal transactions are arbitrary-valued, the amount could be a distinguishing feature. Confidential transactions could hide amounts from observers.

Limitation: Confidential transactions hide amounts but do not address timing leakage. An observer still sees a transaction that finalizes after Δ blocks and infers it is a recovery. Moreover, confidential transactions increase on-chain proof size and computational verification cost. While amount-hiding can be a useful supplementary measure to prevent side-channel inference, it does not solve the

indistinguishability problem on its own. In our protocol, we address amount-hiding as a separate orthogonal concern through protocol design (allowing partial recoveries, using UTXO-like abstractions) rather than heavy cryptographic machinery.

A.7 Comparison with Our Approach

The limitations of these approaches cluster around two axes: (i) they address one aspect of privacy (recipient anonymity, sender anonymity, amount hiding, etc.) but not the transaction type inference problem inherent to the Δ-delay challenge, and (ii) many impose high computational or operational overhead without solving the fundamental problem.

Our integrated approach sidesteps these limitations by:

- **Addressing timing directly:** We use uniform on-chain commitment and reveal timing, combined with off-chain private delivers, to make all transactions structurally and temporally identical to external observers.
- **Guardian anonymity via threshold ring signatures:** We hide not just the sender (as in ring signatures) but the *set of signers*, preventing identification of participating guardians even if the account is public.
- **Usability via off-chain coordination:** We separate on-chain finality timing from user experience by allowing off-chain assurance, avoiding the usability penalties of uniform delays or dummy transaction approaches.
- **Minimal overhead:** We combine existing cryptographic primitives (commitments, TRS, LRS) without requiring new consensus changes, excessive computational proofs, or dummy transaction multiplication.

A.8 Summary

Existing privacy techniques from the cryptography and blockchain literature each address specific aspects of privacy (sender anonymity, recipient anonymity, amount hiding, etc.) but are insufficient in isolation for guardian-based recovery. Stealth addresses and metadata hiding do not address timing leakage. Ring signatures and ZKPs hide logical origin but not execution timing. Dummy transactions and mixing are computationally expensive and do not solve the timing problem. Confidential transactions hide amounts but not transaction type.

Our solution integrates timing-aware design (commit-reveal with uniform finalization), cryptographic anonymity (threshold ring signatures for guardian authorization), off-chain usability (private commitment reveals), and misbehavior accountability (linkable ring signatures for challenges). This modular combination achieves indistinguishability, guardian privacy, and usability without the pitfalls of single-technique approaches or the overhead of heavy cryptographic machinery.