An Analysis of the Architecture and Security Evolution of Cross-Chain Bridges

Anonymous authors
Paper under double-blind review

Abstract

Cross-chain bridges have become essential infrastructure for blockchain interoperability, yet their rapid architectural evolution has been shadowed by systemic vulnerabilities that have led to billions in losses. From early custodial relays to validator-based councils, modular proof systems, and cryptographic light-client designs, each generation has introduced new trade-offs between security, scalability, and trust. This paper analyzes the architectural trajectory of bridges from 2016 through 2025, tracing how design shifts shaped vulnerability profiles and enabled major exploits. We propose a security evaluation framework grounded in eight architectural dimensions that move beyond descriptive attack taxonomies to provide a diagnostic tool for researchers, developers, and auditors. Applied through real-world case studies, the framework highlights how structural risks emerge from design choices and offers a path toward more resilient, trustworthy interoperability infrastructure.

1 Introduction

Cross-chain bridges have become foundational infrastructure in the blockchain ecosystem, enabling interoperability across networks that would otherwise remain siloed. As the number and diversity of blockchain platforms has grown, ranging from general-purpose Layer-1 blockchains like Ethereum to application-specific chains, so has the demand for seamless asset transfer, message passing, and synchronization across chains. A cross-chain bridge is an interoperability mechanism that facilitates the transfer of assets, messages, or state between two or more independent blockchain networks. Without such infrastructure, users and developers would be limited to fragmented liquidity, isolated applications, and constrained ecosystems.

However, the rapid proliferation of bridges has brought with it significant security challenges. In just the past five years, bridge-related vulnerabilities have resulted in billions of dollars in losses [23], including notable incidents such as the Ronin hack [15], the Wormhole exploit [16], and the Nomad bridge attack [14]. These incidents often stem from architectural design flaws - such as compromised multisig validators, improper message verification, or inadequate replay protection - rather than purely implementation bugs. Despite increased scrutiny, bridge attacks continue to represent one of the largest single categories of crypto-related exploits in both volume and impact, accounting for over 50 percent of all DeFi exploits in some years [22].

Recent research has cataloged various security challenges facing cross-chain bridges [6, 7, 9, 27], yet most treatments approach the subject from a vulnerability-first or component-centric perspective. While these surveys provide important classifications of attack types and trust models, they do not offer formal tools for practitioners to assess bridge security under real-world deployment conditions. Furthermore, many evaluations lack a coherent discussion of how bridge architectures have evolved, and how that evolution has influenced both the attack surfaces and resilience of cross-chain bridges.

In this paper, we analyze the architectural evolution of cross-chain bridges from 2016 to 2025, with a focus on how core design choices have shaped their security characteristics. The principal contributions of this work are threefold. First, we synthesize the generational design patterns of bridge architectures and show how those patterns drive distinct security trade-offs. Second, we present a practical, architecture-centric evaluation framework that links observed failures to concrete design dimensions usable by developers, auditors and researchers. Third, we demonstrate the framework's diagnostic value through applied case studies that surface both immediate exploit causes and deeper structural risk factors. Collectively, these contributions build on prior research on bridge security, shifting the focus from forensic cataloging toward prospective, design-driven assessment.

The remainder of this paper is structured as follows: Section 2 reviews the evolution of cross-chain bridge architectures. Section 3 presents a taxonomy of real-world bridge vulnerabilities. Section 4 introduces our security evaluation framework. Section 5 discusses selected case studies and applies our framework to assess different bridge models. Section 6 concludes with recommendations for future bridge design.

2 Evolution of Cross-Chain Bridge Architectures

To evaluate the performance and security landscape of cross-chain bridges, it is essential to examine the evolution of

their architectural implementations. This section analyzes the first decade of bridge designs, from early implementations through to the frontier in 2025. We divide this period into four generations, each marked by qualitative improvements in interoperability design and trust models. The generations we explore are: (i) simple relays and custodial bridges, (ii) multi-signature validator sets, (iii) modular messaging and proof-aware protocols, and (iv) cryptographic and trust-minimized bridges.

2.1 Simple Relays and Custodial Bridges

The earliest bridges delegated liquidity custody and minting functions to centralized operators, typically custodians or relay nodes. In these systems, a user deposited tokens into a smart contract or custodial vault; in the Bitcoin ecosystem, this could involve locking coins by committing block headers in a specified transaction [25]. A centralized set of operators then minted wrapped assets or relayed events on the destination chain, producing corresponding synthetic tokens.

While these implementations were simple and efficient, they relied heavily on trusted intermediaries. This created systemic limitations. First, these bridge architectures created a single point of failure, where the compromise of custodial keys could result in catastrophic loss. These cross-chain bridges also operated using opaque verification mechanisms where users had limited ability to audit underlying on-chain activity, including on-chain locking mechanisms necessary for the verified parity of synthetic tokens produced on the destination chain. In practice, users bore the burden of trust without meaningful fraud resistance.

The risks of custodian compromise and lack of auditability motivated the next generation of bridge designs, which sought to distribute control across multiple validators rather than relying on a single trusted validator node.

2.2 Multi-Signature Validator Sets

The second generation of blockchain bridge architectural implementations made progress on solving centralization constraints in relays and custodial bridges through the incorporation of multi-signature (multisig) vaults and small validator councils [4, 5]. Here, a group of n validator nodes collectively authorized cross-chain messages, asset minting, or transfers, with transactions requiring m-of-n signatures for validity. This distributed control improved resilience, since an attacker would need to compromise multiple validator keys rather than a single custodian.

Despite these improvements, significant vulnerabilities remained. Technically, validator key compromise still enabled bridge takeover, as demonstrated in later high-profile attacks on multisig-based systems [13, 15, 20]. From a governance perspective, many implementations lacked clear mechanisms for validator rotation, slashing, or accountability. Without automated penalties, validators could collude or sign fraudulent transactions without repercussions. Moreover, coordinated attacks against a small validator council remained feasible, particularly when validator sets were small or insufficiently decentralized.

These limitations highlighted the need for more flexible and modular systems that could accommodate different verification models beyond validator signatures alone.

2.3 Modular Messaging and Proof-Aware Protocols

As trust primitives, proof implementations and the ecosystem of blockchains expanded, cross-chain bridges moved towards more optionality in component implementations. The third generation of bridges reflected the growing diversity of blockchain ecosystems and trust primitives by adopting modular architectures. In these designs, the messaging, verification, and execution layers were separated, allowing protocol designers to select verification mechanisms that balanced security, performance, and cost [1, 2, 26]. Verification modules ranged from centralized oracles to decentralized relayers and, in some cases, light clients capable of directly validating source-chain headers.

This modularity offered several benefits. It enabled rapid adaptation to new chains, improved flexibility in responding to exploits, and allowed operators to tailor trust assumptions to their deployment context. However, modularity also expanded the attack surface: flaws in a single component could compromise the entire bridge. Notable exploits underscored this risk. For example, improper initialization logic in one modular bridge allowed attackers to bypass validation entirely [14], while signature verification errors in another led to losses exceeding $300M [16].

While modularity improved adaptability, these systems often continued to rely on validator sets or semi-trusted relayers. The persistence of human-operated verification bottlenecks underscored the demand for designs offering stronger, mathematically grounded guarantees of correctness.

2.4 Cryptographic and Trust-Minimized Bridges

The most recent generation of bridge architectures leverages advances in applied cryptography to achieve trust-minimized security. Rather than depending on validator honesty, these systems establish correctness through cryptographic proofs. Two dominant approaches have emerged: light-client verification and zero-knowledge proof systems.

In light-client bridges, each chain maintains a simplified client of the other, enabling direct verification of block headers and consensus proofs [8, 10]. This eliminates

reliance on external validators but faces scalability challenges, especially across heterogeneous consensus mechanisms.

Zero-knowledge-based bridges (zk-bridges) generate succinct proofs such as zk-SNARKs or zk-STARKs, often off-chain. These proofs represent the validity of block headers or transactions from the source chain [21, 24]. The destination chain can efficiently verify these proofs on-chain, compressing entire histories of activity into a single verifiable object. By anchoring security in mathematical correctness, zk-bridges eliminate entire classes of validator collusion and key-compromise risks. However, proof generation is computationally expensive and typically requires specialized off-chain infrastructure, constraining scalability.

A third emerging category, optimistic bridges, accepts state transitions as valid by default, subject to challenge during a dispute window [3]. While this offers efficiency, it introduces reliance on active monitoring and dispute timeliness challenges. It also creates concerns for network liveness and censorship resistance, given the potential for malicious nodes to flag legitimate transactions as invalid which at scale could inhibit transaction finality and network operations.

Together, these approaches represent a paradigm shift: away from reliance on human-governed validator sets and toward cryptographic guarantees of correctness. While widespread adoption remains limited by cost and performance bottlenecks, ongoing improvements in proof systems suggest that trust-minimized bridges will define the next frontier of interoperability.

3 Taxonomy of Real-World Bridge Vulnerabilities

Cross-chain bridges have suffered some of the most severe exploits in the blockchain ecosystem, accounting for over half of the total value lost in DeFi attacks between 2021 and 2023 [22]. While prior studies have documented these incidents in detail through attack-specific or component-specific taxonomies [9, 27], such approaches often remain descriptive. Our objective is not to replicate forensic-focused attack reports but to develop an architecturally grounded taxonomy that highlights how vulnerabilities emerge from the progression of design choices and trust model implementations. Each vulnerability category is therefore linked to evaluation dimensions that form the basis of an evaluative security framework for cross-chain bridge architectures.

3.1 Key Management and Validator Compromise

Validator compromise remains one of the most prevalent causes of cross-chain bridge exploits. Multisignature and validator-based bridges, while an improvement over single custodians, still have their security architectures hinged on private key integrity. When enough validator keys are compromised, attackers gain full control over cross-chain message authorization.

In March 2022, the Ronin Bridge was exploited for $600M after attackers compromised five of nine validator keys, enabling them to approve arbitrary withdrawals [15]. Similarly, the Harmony Horizon Bridge lost $100M due to the compromise of two out of five validator keys [13]. Architecture choices with a limited security posture as demonstrated in these cases resulted in the compromise of the cross-chain bridges. The incidents demonstrate structural limitations in validation soundness and in trust model design. They also show fragility in validator governance and rotation, where validator sets were too small or inadequately managed.

3.2 Consensus Weaknesses and Quorum Assumptions

Beyond outright key theft, validator-based bridges also face systemic risks from quorum design. Small validator sets with poorly defined rotation mechanisms are susceptible to collusion, censorship, or loss of liveness. Concentrated control undermines fault tolerance and creates brittle consensus assumptions. These weaknesses often arise even when no validator compromise occurs, underscoring systemic risks rather than operational failures.

The Multichain Bridge illustrates this fragility, where over-reliance on a few trusted nodes raised systemic concerns when operational disruptions led to user losses [19]. Such cases highlight the risks tied to trust model composition, consensus liveness and fault tolerance, and validator governance particularly in the case of assumed validator liveness without redundancies.

3.3 Smart Contract Logic Bugs

As bridge contracts became more modular, their complexity introduced new categories of vulnerabilities. Logic flaws in verification routines or contract initialization often allowed attackers to bypass intended security checks.

For example, the Nomad Bridge suffered a $190M exploit when an uninitialized root variable caused all messages to be treated as valid [14]. In the Wormhole Bridge attack, Wormhole lost $325M after attackers exploited a flaw in signature verification logic, minting wrapped ETH without valid proofs [16]. These incidents underscore the importance of validation soundness in contract-level verification mechanisms, along with associated smart contract specifications such as foundational variables. They also point to the importance of smart contract auditability and state transparency for improved security guarantees on cross-chain bridges.

3.4 Replay and Message Injection Attacks

Bridges are uniquely vulnerable to replay attacks, where the duplication of state across domains allows attackers to reuse previously valid proofs or messages. Without nonce management, proof uniqueness, or explicit cross-domain separation, attackers can replay transactions to mint duplicate assets.

Although fewer public incidents have stemmed purely from replay attacks, multiple audits have flagged such issues in bridge implementations, confirming that resistance to replay and injection is a necessary evaluative dimension. Audit reports from Quantstamp have repeatedly flagged replay vulnerabilities in cross-chain bridge contracts and other blockchain protocols [11,12], indicating a persistent challenge to be considered in bridge implementations. This category directly informs evaluating cross-chain bridges on their replay and message injection resistance as well as on validation soundness.

3.5 Oracle and Relayer Manipulation

Bridges relying on off-chain oracles or relayers inherit their security assumptions. Oracle reliance shifts the trust-model off-chain, reducing state transparency and weakening auditability. If these external actors are corrupted, collude, or otherwise submit malicious data, the bridge can be fed fraudulent proofs of state. While large-scale exploits in this category have not been widely documented, theoretical vulnerabilities in oracle-relayer models remain significant concerns. These limitations underscore technical challenges in validation soundness, state transparency and auditing. They also highlight the importance of consensus liveness and fault tolerance as security considerations for cross-chain bridges since if oracles fail to submit timely proofs, bridge liveness is disrupted.

3.6 Upgrade and Governance Exploits

Cross-chain bridge protocols often maintain upgradeable contracts for rapid patching, but these privileges also introduce attack vectors. If administrator keys are compromised, or if governance is captured, malicious upgrades can drain assets.

The bZx protocol [17] suffered such an attack when compromised private keys allowed adversaries to alter protocol contracts and exfiltrate funds. Similarly, reduced controls in the poly network on cross-chain smart contract calls caused an attack that exfiltrated more than $600 M in funds [18]. These attacks highlight the dangers of poor control safeguards and upgradeability, where flawed permission sets, upgrades or patching mechanisms compound security risks. Upgradeability remains a point of tension between agility and control risk for cross-chain bridges, with bridge designers and protocol developers needing to navigate the objectives of allowing for upgradeability, minimizing control risk and ensuring state transparency.

3.7 Cryptographic Proof and Infrastructure Risks

The most recent generation of bridges aims to minimize trust through light clients and zero-knowledge proofs. While these designs mitigate validator compromise risks, they introduce new vulnerabilities related to cryptographic soundness and proof infrastructure. A flawed proof circuit, insecure parameter setup, or compromised prover infrastructure could invalidate the system's security guarantees.

While these vulnerabilities are largely theoretical today, they represent frontier risks that may become more salient as zk-bridges achieve wider adoption. These risks inform security considerations on validation soundness, finality anchoring, and consensus liveness. More secure zk-bridge architectures could allow independent verification for generated zero-knowledge proofs to bolster finality anchoring.

This taxonomy highlights how architectural evolution shapes vulnerability profiles: custodial and multisig bridges are most exposed to validator compromise, modular contracts to logic flaws and oracle manipulation, and cryptographic bridges to proof infrastructure risks. The taxonomy shows that bridge vulnerabilities are not isolated flaws but systemic expressions of architectural trade-offs. Each cross-chain bridge architecture has benefits and drawbacks, with implementations accruing different benefits and limitations across dimensions such as scalability, agility and security. By framing architectural trade-offs through security-focused evaluative dimensions, the taxonomy serves as both a retrospective categorization of past failures and a forward-looking diagnostic lens. This prepares the foundation for a systematic security evaluation framework.

4 Security Evaluation Framework

The discussed taxonomy of vulnerabilities illustrates that bridge failures are not isolated coding mistakes or operational oversights, but systemic expressions of architectural trade-offs. To move from retrospective analysis toward forward-looking guidance, we introduce a security evaluation framework for cross-chain bridges. The framework provides a structured set of dimensions through which bridge architectures can be assessed, allowing both practitioners and researchers to identify strengths, weaknesses, and residual risks across designs. Each dimension reflects recurring fault lines observed in past exploits and directly addresses the sources of risk outlined in the vulnerability taxonomy.

4.1 Framework Dimensions

4.1.1 Validation Soundness

At the foundation of bridge security lies the question of how cross-chain messages are verified. Bridges must ensure that proofs of events or state commitments are both authentic and tamper-resistant. Weak validation mechanisms, such as incomplete signature checks or flawed verification routines, have been at the center of many high-value exploits. A secure bridge therefore requires not only correct cryptographic design, but also resilient implementation that withstands adversarial attempts to bypass verification.

4.1.2 Finality Anchoring

A bridge is only as strong as its ability to independently confirm the finality of the source chain's consensus. Without anchoring, users must trust validators or relayers to attest that an event has truly been finalized. This reliance creates trust bottlenecks, as demonstrated in validator-based systems. By contrast, light clients and zero-knowledge proof approaches introduce independent verification of finality, thereby minimizing reliance on intermediaries and aligning bridge security more closely with the underlying chain.

4.1.3 Trust Model Complexity

Trust models vary widely across bridge designs, from single custodians to distributed validator councils to cryptographic trust minimization. Greater complexity in trust assumptions typically increases the surface for collusion, capture, or compromise. Mapping the trust-critical actors, their control thresholds, and their interdependencies is therefore central to evaluating any bridge architecture's security robustness.

4.1.4 Replay and Message Injection Resistance

Because bridges replicate state across domains, they are uniquely exposed to replay and injection attacks. Without safeguards such as nonce management, unique message identifiers, or domain separation, an adversary can reuse previously valid messages to drain liquidity. Even though few public exploits have been attributed solely to replay vulnerabilities, their persistent presence in audits highlights their importance as a security dimension.

4.1.5 Validator Governance and Rotation

Validator-based bridges rely not only on validator honesty but also on robust governance of validator participation. Static or poorly rotated validator sets magnify the risk of compromise, since long-lived keys and concentrated control are prime attack targets. Effective governance mechanisms such as automated rotation, slashing and transparent selection criteria, are essential to strengthen validator-based models.

4.1.6 State Transparency and Auditing

Transparency in bridge state and proof handling allows independent auditors and users to verify the system's integrity. Opaque designs, particularly in custodial or multisig-based systems, reduce the ability of external actors to detect anomalies. Publicly verifiable proofs, accessible audit logs, and independent monitoring infrastructure all enhance bridge accountability and resilience.

4.1.7 Upgradeability and Control Risk

Because vulnerabilities can always surface over a system's lifecycle, bridges must often retain some ability to upgrade their logic. Yet upgradeability itself introduces new attack vectors, as privileged control over upgrades can be exploited. Striking the right balance between agility and control risk requires mechanisms such as multi-party governance, upgrade timelocks, and transparent proposals that mitigate unilateral changes.

4.1.8 Consensus Liveness and Fault Tolerance

Beyond safety, bridges must also remain operational and therefore live in the face of validator failures, censorship, or network disruptions. Systems with fragile liveness assumptions can halt user activity or lock assets even without direct compromise. Evaluating liveness involves examining the redundancy of validator sets, the fault tolerance thresholds of underlying consensus mechanisms, and the ability of the bridge to degrade gracefully under stress.

To operationalize these dimensions, Table 1 compares how different generations of bridge architectures perform across them. Rather than a ranking or quantitative score, the table serves as a diagnostic lens, illustrating the trade-offs inherent in custodial, validator-based, light client, and zero-knowledge bridge designs. By reading across the table, one can see how each architectural choice strengthens or weakens specific dimensions of security. This underscores the necessity of contextual evaluation rather than one-size-fits-all judgments.

As Table 1 illustrates, no bridge architecture dominates across all security dimensions. Custodial and multisig bridges reduce complexity but at the cost of validation soundness and transparency. Light client and zero-knowledge designs strengthen soundness and anchoring but raise new concerns related to liveness and proof infrastructure. These trade-offs reinforce the importance of evaluating bridges contextually, using the framework as a diagnostic tool.

Security Dimension	Custodial / Relay-Based	Multisig Validator Set	Light Client-Based	Zero-Knowledge / Trust-Minimized
Validation Soundness	Weak Relies on custodian honesty	Moderate Depends on validator integrity	Strong Operates on-chain consensus verification	Strong Utilizes cryptographic proof verificationn
Finality Anchoring	Rare Requires trust in custodian finality	Rare Often only requires validator attestation	Present Anchors to source consensus	Present Anchors via zk-proofs or light clients
Trust Model Complexity	Very low High reliance on single custodian	Moderate Subset(s) of validator councils	Low Relies on consensus protocols	Very low Cryptography replaces intermediaries
Replay and Injection Resistance	Weak Limited nonce controls	Variable Implementation-dependent	Strong Domain separation enforced	Strong zk-proofs and uniqueness checks
Validator Governance and Rotation	None Single custodian	Variable Implementation-dependent	Not applicable Direct validation from source chain	Not applicable Cryptographic verification of proofs
State Transparency and Auditing	Variable Dependent on custodial design	Variable Dependent on validator design	Strong Consensus state observable	Strong Proofs are publicly verifiable
Upgradeability / Control Risk	High Full custodian control possible	High Small councils control upgrades	Moderate Dependent on client governance implementation	Low Cryptographic core, though circuits may be upgradable
Consensus Liveness and Fault Tolerance	Weak to Moderate Liveness only dependent on custodian availability	Moderate Small validator set failures possible	Variable Liveness depends on underlying chain	Moderate Proof generation bottlenecks may delay liveness depending on implementation

Table 1: Security Evaluation Framework Across Cross-Chain Bridge Architectures.
Note: 'Variable' denotes cases where security outcomes depend heavily on specific implementation choices.

5 Case Studies and Security Framework Applications

5.1 Ronin Bridge (Axie Infinity)

The Ronin Bridge, developed by Sky Mavis to connect the Ethereum mainnet with the Ronin sidechain supporting the Axie Infinity ecosystem, was one of the largest cross-chain bridges in operation at the time of its exploit. On March 23, 2022, attackers compromised five of the bridge's nine validator nodes, authorizing fraudulent withdrawals that drained approximately $624 million in ETH and USDC [15]. This remains one of the largest single incidents in the history of decentralized finance. We proceed to evaluate Ronin Bridge's exploit across the security framework's dimensions.

5.1.1 Validation Soundness

Ronin relied on validator signatures for message authorization. While this provided moderate assurance in theory, the compromise of a majority threshold allowed attackers to generate signatures used to fraudulently withdraw funds. Validation soundness was therefore contingent on validator key security rather than cryptographic guarantees.

5.1.2 Finality Anchoring

The bridge lacked mechanisms to independently anchor transactions to Ethereum's consensus finality. Users trusted validator attestations without an on-chain verification of Ethereum state. This absence of anchoring meant that once validators were compromised, there were no external checks on fraudulent transactions.

5.1.3 Trust Model Complexity

Ronin's trust model was moderately complex but highly concentrated. Security depended on a small validator set operated primarily by entities close to Sky Mavis. While this reduced operational friction, it also increased systemic fragility, as control of just over half the validators conferred unilateral power over the bridge.

5.1.4 Replay and Message Injection Resistance

The Ronin Bridge included nonce management and message uniqueness checks sufficient to prevent replay attacks. No evidence suggests that replay or message injection contributed to the exploit. This dimension therefore rated adequately, though still dependent on validator honesty for enforcement.

5.1.5 Validator Governance and Rotation

Validator governance was notably weak. The validator set was static, with no robust procedures for regular rotation, slashing, or penalization of misbehavior. This created long-lived validator keys that were highly attractive attack targets. Once compromised, there was no automated mechanism to detect or replace malicious validators.

5.1.6 State Transparency and Auditing

Ronin's state verification was opaque to users. Since state commitments relied solely on validator signatures, external auditors and users could not independently verify bridge activity. This limited transparency delayed detection of the exploit, such that the fraudulent withdrawals were discovered only after users were unable to withdraw funds.

5.1.7 Upgradeability and Control Risk

Ronin Bridge contracts were upgradable under validator governance, meaning control was concentrated in the same set of entities responsible for validation. While this provided agility for patching, it also compounded systemic control risks, as compromised validators could in principle execute malicious upgrades.

5.1.8 Consensus Liveness and Fault Tolerance

The bridge's liveness depended on validator availability. With only nine validators, even partial downtime posed risks of transaction delays or censorship. The attack demonstrated that validator collusion could simultaneously compromise safety and undermine liveness, leaving the bridge unable to function as designed.

5.2 Nomad Bridge

While Ronin illustrates vulnerabilities stemming from validator governance and anchoring, the Nomad Bridge exploit highlights a different profile of weaknesses. Nomad employed a modular contract-based architecture in which validation occurred through message-passing contracts. In principle, this design reduced dependence on small validator councils and achieved better decentralization of trust. However, an initialization bug left a critical contract variable unconfigured, causing the system to treat all proofs as valid. Attackers exploited this flaw to drain approximately $190M in assets by copying transaction calls [14].

Through the lens of our framework, Nomad demonstrates how contract-centric architectures can achieve stronger outcomes in some dimensions, yet still fail significantly in others. Compared to Ronin, Nomad fared better on trust model complexity and validator governance, since it did not rely on a concentrated validator council. Yet its validation soundness was fatally undermined by flawed initialization, and its state transparency and auditing were limited. As a result, external users lacked mechanisms to detect that invalid proofs were being accepted until funds were already drained.

This comparison reinforces two key insights. First, vulnerabilities map differently across architectures: Ronin's risks clustered around validator compromise, while Nomad's failure stemmed from implementation gaps in validation logic. Second, the framework applies equally across both cases, surfacing distinct but equally critical weaknesses. Taken together, the Ronin and Nomad incidents show that no single architectural family is exempt from systemic risks, underscoring the value of structured, dimension-based evaluation.

The analysis of cross-chain bridge architecture evolutions and the security evaluation framework as structured and applied provide a foundation for a broader discussion on the implications of bridge architecture and future directions of secure cross-chain interoperability.

6 Recommendations and Future Directions for Secure Bridge Architectures

The analysis of bridge architectures and vulnerabilities presented in this paper highlights the need for more robust design principles that address security risks at their root. Several directions for future bridge design emerge from our evaluation framework. First, validator-based models should evolve beyond static councils. Stronger governance mechanisms, such as automated validator rotation, slashing penalties for misbehavior, and more transparent selection processes, are essential to reduce concentration risk. Second, validation soundness can be improved by integrating cryptographic verification methods such as light clients and zero-knowledge proofs, which minimize reliance on trusted

intermediaries. While these approaches remain challenged by computational cost and infrastructure complexity, continued advances in applied cryptography make them promising foundations for next-generation bridges.

Equally important is the need to balance flexibility with security in system upgrades. Upgradeability remains critical for patching vulnerabilities, yet poorly governed upgrade paths have repeatedly introduced new attack vectors. Future designs would benefit from adopting multi-party governance, timelocks, and transparent on-chain proposals to ensure that upgrades cannot be exploited as control backdoors. State transparency also requires greater attention: public audit logs, verifiable proofs, and independent monitoring tools should become standard features of bridge deployments to enhance accountability and early anomaly detection.

Future research should extend beyond individual bridge protocols toward systemic perspectives. Scalability of zero-knowledge-based bridges, decentralization of proof generation, and the integration of cross-bridge replay resistance in multi-chain ecosystems remain open challenges. There is also a need for benchmarking frameworks that can translate qualitative evaluation dimensions into measurable criteria for comparative analysis. By advancing these research directions, the community can develop not only more secure bridges but also more standardized approaches to assessing their resilience.

Ultimately, secure bridges are a prerequisite for the broader vision of interoperable blockchain ecosystems. By framing vulnerabilities through evaluative dimensions, our framework contributes to this effort by offering a structured diagnostic lens for both practitioners and researchers. As bridges continue to evolve, adopting these recommendations will be essential to aligning scalability and usability with the uncompromising need for security.

7 Conclusion

This paper traced the evolution of cross-chain bridge architectures from custodial relays to cryptographic, trust-minimized designs, showing how each generation's trust assumptions and implementation approaches shaped distinct vulnerabilities. We introduced a security evaluation framework of eight dimensions that links architectural choices to systemic risks, moving beyond attack catalogues toward a diagnostic tool for researchers, developers, and auditors. Case studies of the Ronin and Nomad exploits demonstrated the framework's value in surfacing points of validator governance fragility and contract-level validation flaws, underscoring the need for architecture-aware assessments. By synthesizing design patterns, developing a structured evaluative lens, and applying it to real-world incidents, our work contributes a foundation for more resilient bridge security. As interoperability becomes central to the blockchain ecosystem, adopting rigorous, dimension-based evaluation will be essential to align scalability and usability with durable safety.

References

[1] Connext. Connext - technical docs, 2024. Accessed: 2025-09-30. URL: https://docs.connext.network/.

[2] Cosmos. Cosmos whitepaper. Technical report, 2024. URL: https://cosmos.network/whitepaper/.

[3] Dénes László Fekete and Attila Kiss. Trust-minimized optimistic cross-rollup arbitrary message bridge. *Journal of Network and Computer Applications*, 221:103771, 2024. URL: https://www.sciencedirect.com/science/article/pii/S108480452300190X, https://doi.org/https://doi.org/10.1016/j.jnca.2023.103771.

[4] David Galindo and Jia Liu. *Robust Subgroup Multi-signatures for Consensus*, pages 537–561. 01 2022. https://doi.org/10.1007/978-3-030-95312-6_22.

[5] Jongbeen Han, Mansub Song, Hyeonsang Eom, and Yongseok Son. An efficient multi-signature wallet in blockchain using bloom filter. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 273–281, New York, NY, USA, 2021. Association for Computing Machinery. https://doi.org/10.1145/3412841.3441910.

[6] Sung-Shine Lee, Alexandr Murashkin, Martin Derka, and Jan Gorzny. Sok: Not quite water under the bridge: Review of cross-chain bridge hacks. In *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–14, 2023. https://doi.org/10.1109/ICBC56567.2023.10174993.

[7] Ningran Li, Minfeng Qi, Zhiyu Xu, Xiaogang Zhu, Wei Zhou, Sheng Wen, and Yang Xiang. Blockchain cross-chain bridge security: Challenges, solutions, and future outlook. *Distributed Ledger Technologies: Research and Practice*, 4(1), February 2025. https://doi.org/10.1145/3696429.

[8] Near. Eth-near rainbow bridge, 2020. Accessed: 2025-09-30. URL: https://www.near.org/blog/eth-near-rainbow-bridge.

[9] Jakob Svennevik Notland, Jingyue Li, Mariusz Nowostawski, and Peter Halland Haro. Sok: Cross-chain bridging architectural design flaws and mitigations. *Blockchain: Research and Applications*, page 100315, 2025. URL: https://www.sciencedirect.com/science/article/pii/

S2096720925000429, https://doi.org/https://doi.org/10.1016/j.bcra.2025.100315.

[10] Polkadot. Polkadot - light clients, 2025. Accessed: 2025-09-30. URL: https://docs.polkadot.com/develop/toolkit/parachains/light-clients/.

[11] Quantstamp. Nomad - quantstamp security assessment audit report, 2022. Accessed: 2025-09-30. URL: https://d33wubrfki0l68.cloudfront.net/a4aa692a35a2e6a05c09763229fe81a1011d85e7/2ebfb/nomad-audit.pdf.

[12] Quantstamp. Pine - quantstamp security assessment audit report, 2022. Accessed: 2025-09-30. URL: https://files.gitbook.com/v0/b/gitbook-x-prod.appspot.com/o/spaces%2Fx2kcknwVE1ikWzd0nPDQ%2Fuploads%2FeTag4DBkct3MxOr1Zpgo%2FPine%20-%20Final%20Report.pdf?alt=media&token=4859ad3c-0fdc-4542-ab94-db06bdf4c226.

[13] Rekt. Harmony bridge - rekt, 2022. Accessed: 2025-09-30. URL: https://rekt.news/harmony-rekt.

[14] Rekt. Nomad bridge - rekt, 2022. Accessed: 2025-09-30. URL: https://rekt.news/nomad-rekt.

[15] Rekt. Ronin network - rekt, 2022. Accessed: 2025-09-30. URL: https://rekt.news/ronin-rekt.

[16] Rekt. Wormhole - rekt, 2022. Accessed: 2025-09-30. URL: https://rekt.news/wormhole-rekt.

[17] Rekt. Bzx - rekt, 2023. Accessed: 2025-09-30. URL: https://rekt.news/bzx-rekt.

[18] Rekt. Poly network - rekt, 2023. Accessed: 2025-09-30. URL: https://rekt.news/polynetwork-rekt.

[19] Rekt. Rekt - multichain, 2023. Accessed: 2025-09-30. URL: https://rekt.news/multichain-r3kt.

[20] Rekt. Radiant capital - rekt ii, 2024. Accessed: 2025-09-30. URL: https://rekt.news/radiant-capital-rekt2.

[21] Uma Roy, John Guibas, Kshitij Kulkarni, Mallesh Pai, and Dan Robinson. Succinct network: Prove the world's software, 2025. Accessed: 2025-09-30. URL: https://pdf.succinct.xyz/.

[22] Chainalysis Team. 2022 biggest year ever for crypto hacking with $3.8 billion stolen, primarily from defi protocols and by north korea-linked attackers, 2022. Accessed: 2025-09-30. URL: https://www.chainalysis.com/blog/2022-biggest-year-ever-for-crypto-hacking/.

[23] Chainalysis Team. Vulnerabilities in cross-chain bridge protocols emerge as top security risk, 2022. Accessed: 2025-09-30. URL: https://www.chainalysis.com/blog/cross-chain-bridge-hacks-2022/.

[24] Tiancheng Xie, Jiaheng Zhang, Zerui Cheng, Fan Zhang, Yupeng Zhang, Yongzheng Jia, Dan Boneh, and Dawn Song. zkbridge: Trustless cross-chain bridges made practical, 2022. URL: https://rdi.berkeley.edu/zkp/uploads/paper.pdf, arXiv:2210.00264, https://doi.org/https://doi.org/10.48550/arXiv.2210.00264.

[25] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William Knottenbelt. Sok: Communication across distributed ledgers. In *Financial Cryptography and Data Security*, 2019. URL: https://eprint.iacr.org/2019/1128.pdf, https://doi.org/10.1007/978-3-662-64331-0_1.

[26] Ryan Zarich, Bryan Pellegrino, Isaac Zhang, Thomas Kim, and Caleb Banister. Layerzero. Technical report, 2024. URL: https://layerzero.network/publications/LayerZero_Whitepaper_V2.1.1.pdf.

[27] Mengya Zhang, Xiaokuan Zhang, Yinqian Zhang, and Zhiqiang Lin. Security of cross-chain bridges: Attack surfaces, defenses, and open problems. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, page 298–316, New York, NY, USA, 2024. Association for Computing Machinery. URL: https://arxiv.org/pdf/2312.12573, https://doi.org/10.1145/3678890.3678894.