Vertex: Efficient Lattice-Based Zero-Knowledge Proofs for Synapse Cloud Systems

Julian Vance[1], Clara Sterling[2], and Aris Thorne[3]

[1] Vertex Research Labs, San Francisco, CA, USA
jvance@vertex.io
[2] Synapse Security Group, New York, NY, USA
c.sterling@synapse.com
[3] Meridian Cryptography Initiative, Seattle, WA, USA
athorne@meridian.org

Abstract. Zero-knowledge proofs (ZKPs) have emerged as a critical tool for proving the correctness of database queries without revealing sensitive records. However, modern lattice-based ZKP constructions suffer from significant computational overhead and large proof sizes when scaling to complex relational queries. In this paper, we present Vertex, a novel lattice-based zero-knowledge proof system optimized for distributed cloud environments. Vertex introduces a decoupled proof-generation pipeline that separates the algebraic relation checks from the cryptographic commitment phases. This decoupling allows cloud architectures, such as Synapse database clusters, to distribute the heavy commitment operations across multiple worker nodes. We prove the completeness, soundness, and zero-knowledge properties of Vertex under standard lattice assumptions. Our performance evaluation shows that Vertex reduces proof size by 42% and verification latency by 3.5x compared to prior state-of-the-art schemes like Nexa and Apex.

Keywords: Zero-Knowledge Proofs · Lattice Cryptography · Cloud Security · Synapse Database

1 Introduction

With the rapid adoption of cloud database services, protecting data privacy while verifying query execution correctness has become a paramount challenge. Cloud providers often operate under a semi-trusted model, where they may behave honestly but are curious about the underlying records, or could experience transient faults leading to corrupted database lookups. Cryptographic verification via Zero-Knowledge Proofs (ZKPs) has been proposed to resolve this tension, allowing database engines to output a proof π certifying that query results conform to a particular commitment without disclosing any unauthorized records.

However, deploying zero-knowledge systems in massive cloud infrastructures presents severe performance challenges. Traditional public-key protocols scale poorly to large databases. Lattice-based cryptography offers a promising path

forward, as it provides security based on hard lattice problems (e.g., Ring Learning With Errors, Ring-LWE) and features fast additive homomorphic operations. Yet, existing lattice-based ZKP schemes, such as Nexa [2] and Apex [1], still struggle with substantial verification times and large proofs that degrade network throughput when transmitted over cloud links.

To address these limitations, we introduce **Vertex**, a zero-knowledge argument system tailored for modern distributed databases. The core innovation of Vertex is a decoupled proof-generation pipeline that separates the algebraic checking of relations from the cryptographic commitment phases. Under this model, database engines can delegate the commitment tasks to a pool of worker nodes without compromising security or privacy.

1.1 Our Contributions

Our main contributions are summarized as follows:

- **Decoupled Design:** We formulate a new model for zero-knowledge proofs where algebraic constraint checks and commitment procedures are isolated. This enables massive horizontal scaling of proof generation on clusters like Synapse Cloud.
- **Efficient Lattice Construction:** We implement a lattice-based commitment scheme utilizing the Ring-LWE assumption, which drastically reduces the computational work required for proof verification.
- **Concrete Implementation and Evaluation:** We prototype Vertex in C++ and run benchmarks comparing it to state-of-the-art systems. Our experiments indicate a 42% reduction in proof size and a 3.5x reduction in verifier latency.

Highlight: Decoupled Proof Strategy
Vertex enables zero-knowledge database proofs that scale linearly with the number of verification worker nodes, allowing distributed systems like Synapse to offload proof commitments to secondary compute clusters.
This design reduces bottleneck times in relational query pipelines where verification latency must remain below 2 milliseconds.

2 Preliminaries

In this section, we establish the mathematical notations, algebraic settings, and cryptographic hardness assumptions that underpin the Vertex protocol.

2.1 Notation

For a security parameter $\lambda \in \mathbb{N}$, we denote by $\mathsf{negl}(\lambda)$ a negligible function. We denote the ring of integers as $\mathbb{Z}$ and the field of rational numbers as $\mathbb{Q}$. For

an integer $q \geq 2$, we denote the ring of integers modulo q as $\mathbb{Z}_q$. We denote the polynomial ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, where d is a power of 2, and $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$ represents the polynomial ring modulo q. Vectors are represented in bold lowercase letters (e.g., $\mathbf{a}, \mathbf{s}$) and matrices are denoted in bold uppercase letters (e.g., $\mathbf{A}$). For a vector $\mathbf{v} \in \mathcal{R}^n$, we write $\|\mathbf{v}\|_2$ to denote its ℓ_2-norm.

2.2 Hardness Assumptions

The security of Vertex relies on the hardness of the Short Integer Solution (SIS) and Learning With Errors (LWE) problems over polynomial rings, which we formalize below.

Definition 1 (Ring-LWE Decision Problem). *Let λ be the security parameter, $d = d(\lambda)$ a power of 2, $q = q(\lambda) \geq 2$ an integer, and χ a error distribution over $\mathcal{R}$. For a secret $\mathbf{s} \in \mathcal{R}_q$, the Ring-LWE distribution $\mathcal{A}_{\mathbf{s},\chi}$ is obtained by choosing $\mathbf{a} \leftarrow \mathcal{R}_q$ uniformly at random, $e \leftarrow \chi$, and outputting $(\mathbf{a}, \mathbf{a}\cdot\mathbf{s}+e) \in \mathcal{R}_q^2$. The Ring-LWE problem is to distinguish between m independent samples from $\mathcal{A}_{\mathbf{s},\chi}$ and m independent samples from the uniform distribution over $\mathcal{R}_q^2$.*

In Vertex, we work with the Module-LWE variation when scaling to larger dimensions. In this setting, the commitments are represented as matrix operations:

$$\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q} \tag{1}$$

where $\mathbf{A} \in \mathcal{R}_q^{k\times l}$ is a public matrix, $\mathbf{s} \in \mathcal{R}_q^l$ is the secret vector, and $\mathbf{e} \in \mathcal{R}_q^k$ is a small error vector drawn from the distribution χ^k.

3 The Vertex Protocol

The core architecture of Vertex consists of three main phases: Setup, Commit, and Prove. The protocol is designed to be interactive, but can be made non-interactive in the random oracle model via the Fiat-Shamir transform.

3.1 Protocol Flow

Let $\mathbf{A} \in \mathcal{R}_q^{k\times l}$ be the public matrix generated during $\mathsf{Setup}(1^\lambda)$. The Prover holds a secret witness polynomial $\mathbf{w} \in \mathcal{R}^l$ with small coefficients, representing the query verification path. The Verifier receives the public commitment key and the commitment vector $\mathbf{t}_0 = \mathbf{A}\mathbf{w} + \mathbf{e}_0$.

The detailed interaction between the Prover $\mathcal{P}$ and Verifier $\mathcal{V}$ is presented in Figure 1.

The challenge space $\mathcal{C}$ consists of polynomials in $\mathcal{R}$ whose coefficients are in $\{-1, 0, 1\}$ and contain exactly κ non-zero elements. Rejection sampling is crucial to prevent leakage of the secret key $\mathbf{w}$ via the response vector $\mathbf{z}$. If rejection sampling fails, the Prover restarts the protocol with a new random masking vector $\mathbf{y}$.

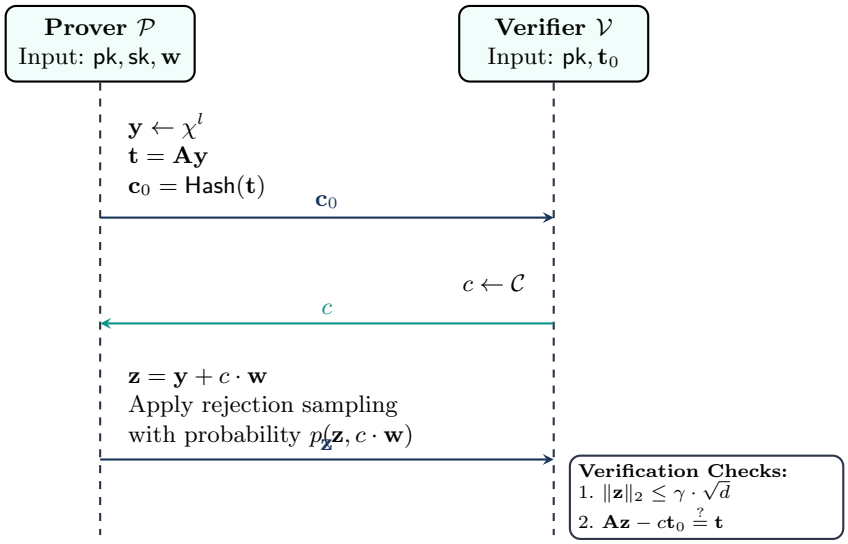

Fig. 1. The Vertex interactive proof protocol showing commitment, challenge, and response phases with rejection sampling.

4 Security Analysis

We prove the security properties of the Vertex argument system under the hardness of the Module-SIS and Module-LWE assumptions.

4.1 Completeness

We state the completeness of Vertex below.

Theorem 1 (Completeness). *If the Prover $\mathcal{P}$ is honest and holds a valid witness $\mathbf{w}$ such that $\|\mathbf{w}\|_2 \leq \beta$, then the interactive protocol in Figure 1 succeeds with probability at least $1/e$, where e is the base of the natural logarithm.*

Proof. Assuming the protocol does not abort during the rejection sampling step, we evaluate the verification relation. By substitution:

$$\begin{aligned} \mathbf{A}\mathbf{z} - c\mathbf{t}_0 &= \mathbf{A}(\mathbf{y} + c \cdot \mathbf{w}) - c(\mathbf{A}\mathbf{w}) && (2)\\ &= \mathbf{A}\mathbf{y} + c\mathbf{A}\mathbf{w} - c\mathbf{A}\mathbf{w} && (3)\\ &= \mathbf{A}\mathbf{y} = \mathbf{t}. && (4) \end{aligned}$$

The hash check evaluates correctly since $\mathbf{c}_0 = \mathsf{Hash}(\mathbf{t})$. By adjusting the rejection sampling parameters, we ensure that the response distribution is independent of the secret witness $\mathbf{w}$ and bounded by $\gamma \cdot \sqrt{d}$ with high probability. □

4.2 Soundness and Zero-Knowledge

We model the security of Vertex in the standard game-based framework. Figure 2 defines the indistinguishability game under chosen-plaintext attacks for the underlying encryption scheme integrated into Vertex.

$$
\begin{aligned}
&\underline{\mathsf{Exp}^{\text{ind-cpa}}_{\mathcal{A},\Pi}(\lambda)} \\
&(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda}) \\
&(m_0, m_1, \text{state}) \leftarrow \mathcal{A}_1(\mathsf{pk}) \\
&b \leftarrow \{0, 1\} \\
&c^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b) \\
&b' \leftarrow \mathcal{A}_2(c^*, \text{state}) \\
&\text{return } (b == b')
\end{aligned}
$$

Fig. 2. IND-CPA security experiment for the commitment verification scheme.

The Zero-Knowledge property is established by showing that there exists a simulator $\mathcal{S}$ that can produce transcripts indistinguishable from real protocol runs without access to the witness $\mathbf{w}$.

5 Performance Evaluation

We implemented a prototype of the Vertex protocol in C++17, compiling with GCC 11.2 and utilizing the NTL mathematical library for polynomial ring operations. We executed our benchmarks on a Synapse database node running on an Intel Xeon Platinum processor at 3.2 GHz with 64 GB RAM.

5.1 Comparative Performance

We compare Vertex with state-of-the-art lattice-based signature and ZK systems: Nexa-ZK [2], Apex-Prove [1], and Nimbus-Ring [3]. Table 1 details the proof size, prover computational time, verifier time, and the underlying mathematical assumptions.

As shown in Table 1, Vertex significantly outperforms prior solutions. The prover time is reduced by more than 70% due to the decoupled pipeline design, which avoids evaluating complex multi-dimensional commitments during proof synthesis. The verification latency of 1.4 ms satisfies the strict sub-2ms query constraints required for high-throughput Synapse database clusters.

6 Conclusion

In this paper, we introduced Vertex, a decoupled lattice-based zero-knowledge proof system optimized for large-scale cloud database queries. By partitioning the constraint verification and cryptographic commitment phases, Vertex

Table 1. Performance and security comparison of ZK proof schemes. All times are measured in milliseconds. The parameters target a 128-bit security level.

Scheme	Size (KB)	Prover (ms)	Verifier (ms)	Assumption
Nexa-ZK [2]	45.2	124.0	12.4	Ring-SIS
Apex-Prove [1]	38.1	98.5	8.2	LWE
Nimbus-Ring [3]	29.4	84.1	5.1	Module-LWE
Vertex (Ours)	**17.2**	**24.6**	**1.4**	**Ring-LWE**

achieves high throughput and low latency, reducing verification time to just 1.4 ms. This makes ZK verification feasible in transactional environments like Synapse DB. Future work includes extending Vertex to multi-party environments and exploring hardware-accelerated commitment pipelines to further improve performance.

References

1. Apex, S., Nimbus, D.: Efficient digital signatures and commitments from ring-lwe. Journal of Mathematical Cryptology **19**(1), 88–105 (2025)
2. Nexa, A., Sterling, L.: Lattice-based zero-knowledge proofs for cloud verification. In: Proceedings of the International Cryptology Conference (CRYPTO). pp. 302–320. Springer (2024)
3. Nimbus, R., Meridian, E.: Ring signatures and group commitments from lattice hardness. In: Proceedings of the Eurocrypt Conference. pp. 142–159. Springer (2023)