Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
E-mail: jicn.office@ffspub.com

Article

Lattice-Based Certificateless Collaborative Signature Scheme for Post-Quantum Intelligent Computing and Network Systems

Yang Li†
China Telecom Quantum Technology Co., Ltd. Hefei 230088, China
†*E-mail:liyang2@chinatelecom.cn*

Received: June 14, 2025 / Revised: June 23, 2025 / Accepted: July 7, 2025 / Published online: July 28, 2025

Abstract: With the rapid development of quantum computing technology, traditional collaborative signature schemes based on RSA and elliptic curve cryptography face significant security challenges. This paper proposes a lattice-based certificateless collaborative signature scheme (L-CCS) to address the security and efficiency requirements of multi-party collaboration in intelligent computing and network systems. By leveraging lattice problems such as the Small Integer Solution and Learning With Errors, our scheme ensures post-quantum security. The L-CCS eliminates the need for a trusted Key Generation Center and a dedicated aggregator, enhancing decentralization and system robustness. Our construction achieves existential unforgeability under adaptive chosen-message attacks in the random oracle model. Furthermore, we provide a practical implementation optimized for large-scale data sharing (LSDS), demonstrating sublinear scalability in signature size and computation time.

Keywords: Lattice-based cryptography; certificateless collaborative signature; post-quantum security; intelligent computing and network system
https://doi.org/10.64509/jicn.11.11

1 Introduction

Collaborative signature schemes enable distributed authorization of digital messages while preserving individual accountability, which is critical for emerging applications in intelligent computing and network systems such as cloud-based healthcare systems, federated governance platforms, and IoT-enabled monitoring. Traditional collaborative signature frameworks built on RSA or elliptic curve cryptography (ECC) face fundamental limitations in post-quantum security (PQS) landscapes. Shor's algorithm[1] threatens to undermine these classical cryptographic primitives, rendering decades of encrypted data vulnerable to quantum adversaries. Recent advancements in post-quantum cryptography have explored hash-based[2, 3] and code-based[4, 5] schemes, yet these approaches often suffer from impractical signature sizes or lack native support for collaborative workflows.

Lattice-based cryptography has emerged as a promising foundation for quantum-resistant collaborative signatures. Its security relies on the hardness of lattice problems such as the Small Integer Solution (SIS) and Learning With Errors (LWE)[6], which remain intractable even against quantum adversaries. Recent works have demonstrated the feasibility of lattice-based aggregate signatures[7, 8], yet these constructions often inherit limitations from their classical counterparts. For instance, existing lattice-based collaborative schemes either require rigid signing orders[9], impose excessive computational overhead on resource-constrained devices[10], or depend on trusted third parties for key issuance[11]. In particular, the need for a Key Generation Center (KGC) and a dedicated aggregator, as seen in the approach proposed by Ma et al.[12], introduces potential points of failure and centralization risks.

Certificateless cryptography offers a compelling idea to the key escrow problem inherent in identity-based systems. By combining a KGC-issued partial key with a user-generated secret, certificateless schemes eliminate the need for certificates while preventing any single entity from forging signatures. However, existing certificateless collaborative signatures predominantly rely on bilinear pairings[13] or modular exponentiation[14], which are vulnerable to quantum attacks. To the best of our knowledge, no prior work has

† Corresponding author: Yang Li
*Academic Editor: Chunxiao Jiang

Table 1: Comparison of L-CCS with Existing Schemes

Scheme	PQS	Certificateless	LSDS	Lightweight Sign
Boneh et al.[13],Lindell et al.[14]	False	True	False	True
Prajapat et al.[7], Xu et al.[8]	True	False	True	False
Liu et al.[2],Feneuil et al.[4]	True	False	True	False
LBCMS[15],SDVS[16]	True	False	False	True
Our L-CCS	True	True	True	True

integrated certificateless design principles with lattice-based cryptography to achieve post-quantum secure, decentralized collaborative signatures.

This paper addresses this critical gap through three primary innovations:

1. A lattice-based certificateless collaborative signature (L-CCS) scheme that supports unordered signing, resists quantum and collusion attacks, and eliminates key escrow. Notably, our construction removes the reliance on a KGC and a dedicated aggregator, reducing centralization risks and enhancing system robustness.

2. Formal security proofs under the Small Integer Solution assumption, demonstrating existential unforgeability against adaptive chosen-message attacks (EUF-CMA) in the random oracle model.

3. A practical implementation blueprint optimized for intelligent computing and network systems, achieving sublinear signature aggregation and verification costs, even for large-scale patient datasets.

By eliminating the KGC and aggregator, our L-CCS scheme offers a more decentralized and secure approach to collaborative signatures, paving the way for robust and efficient applications in post-quantum scenarios. Table 1 lists the progressiveness of this scheme compared with the existing schemes.

2 Related Works

2.1 Collaborative Signature

A collaborative signature scheme enables multiple parties to jointly generate a valid digital signature over a message without revealing their individual private key shares. Formally, consider n participants with key pairs (sk_i, pk_i) collaborating to produce a signature σ for message m, such that:

$$\mathsf{Verify}(pk_1, \ldots, pk_n, m, \sigma) = 1, \tag{1}$$

where no party learns any other participant's secret key. For instance, in ECDSA-based two-party schemes, the private key d is split into shares d_1 and d_2 (with $d \equiv d_1 + d_2 \bmod q$), and the public key Q is derived as $Q = d \cdot G$. The signing protocol involves interactive computations of ephemeral nonces k_1, k_2, followed by joint generation of the signature components $r = (k_1 k_2 G)_x$ and $s = k_1^{-1}(H(m) + rd)$, secured via zero-knowledge proofs to prevent key leakage. This framework ensures that neither party gains information about the other's secret share, even in the presence of malicious adversaries [14].

Collaborative signature research has evolved through three distinct paradigms: multi-signatures, threshold signatures, and aggregate signatures. Early work by Boneh et al.[13] introduced aggregate signatures, enabling compression of multiple signatures into compact representations. However, these schemes lack flexibility in collaboration policies. Schnorr-based multi-signatures[17] improved coordination efficiency through linear signature aggregation, yet their security hinges on synchronized communication rounds. Threshold schemes like Lindell's two-party ECDSA[14] addressed malicious security via secure multi-party computation (MPC), achieving practical latency (e.g., 100ms per operation) at the cost of increased communication complexity.

Recent advancements in lattice-based cryptography have inspired new collaborative signature constructions. Hu et al.[9] proposed a ring-LWE-based threshold scheme with post-quantum security, but their reliance on a trusted Certificate Authority (CA) introduces centralization risks. The KTXR20 protocol[10] achieved three-round signing using non-interactive zero-knowledge (NIZK) proofs, yet precomputed lattice trapdoors limit scalability. Alam et al.[11] explored certificateless designs using identity-based encryption, but their security model assumes semi-honest participants. This underscores the urgent need for decentralized, lattice-based constructions resilient against quantum adversaries.

2.2 Lattice-Based Cryptography

Lattice cryptography relies on the computational hardness of problems defined over structured lattices. A lattice $\mathscr{L}$ is generated as the set of integer linear combinations of basis vectors $\mathbf{B} \in \mathbb{Z}^{n \times m}$:

$$\mathscr{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^m\}. \tag{2}$$

The security of lattice primitives hinges on two core problems: the Small Integer Solution problem, which requires finding a non-trivial vector $\mathbf{x}$ such that $\mathbf{Ax} \equiv 0 \bmod q$ for a random matrix $\mathbf{A}$, and the Learning With Errors problem, which involves distinguishing noisy inner products $(\mathbf{A}, \mathbf{As+e})$ from uniform pairs [18]. These problems are conjectured to resist quantum attacks, making them foundational for post-quantum cryptography.

Lattice-based cryptography has emerged as the leading candidate for post-quantum security due to its reliance on computationally hard lattice problems. The Small Integer Solution problem and Learning With Errors problem form the bedrock of these constructions. Recent optimizations like the

CRYSTALS-Dilithium scheme[19] have demonstrated practical efficiency through polynomial multiplication acceleration using Number Theoretic Transforms (NTTs). However, integrating these advancements with certificateless collaborative signature frameworks remains challenging.

Key challenges in lattice-based collaborative signatures include:

- **Distributed Trapdoor Management**: Existing constructions either rely on centralized key generation or introduce prohibitive communication overhead.
- **Adaptive Security Proofs**: Most lattice-based signatures lack formal proofs under realistic adversarial models.
- **Implementation Overheads**: High-dimensional lattice operations impose significant computational costs on resource-constrained devices.

This work addresses these challenges by introducing a novel certificateless collaborative signature framework leveraging lattice-based cryptography. Our construction eliminates trusted third parties through threshold key generation, ensures post-quantum security under the SIS assumption, and achieves practical efficiency through optimized Gaussian sampling and modular arithmetic.

Table 2: Symbol Description

Symbol	Description
$\Lambda_q^{\perp}(\mathbf{A})$	Lattice with basis $\mathbf{A}$
$\mathbf{S}_i, \mathbf{P}_i$	Signer i's private/public key pair
$\mathbf{z}_a, \boldsymbol{\mu}_a$	Aggregated signature components
ϖ_i	Message fragment for signer i
H_1, H_2, H_3, H_4	Cryptographic hash functions

3 Preliminaries

3.1 Lattice

A lattice $\mathscr{L}$ in n-dimensional space is a discrete additive subgroup of $\mathbb{R}^n$. Formally, given a basis matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m] \in \mathbb{Z}^{n\times m}$, the lattice generated by $\mathbf{B}$ is defined as:

$$\mathscr{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{m} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}. \tag{3}$$

For cryptographic applications, we primarily consider q-ary lattices. Specifically, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the orthogonal lattice is defined as:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x} \equiv \mathbf{0} \mod q\}. \tag{4}$$

3.2 Small Integer Solution

The Small Integer Solution problem is a fundamental hard problem in lattice cryptography. Let $n, m, q \in \mathbb{Z}$ and $\beta \in \mathbb{R}$. Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the $\mathrm{SIS}_{n,m,q,\beta}$ problem requires finding a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that:

$$\mathbf{A}\mathbf{x} \equiv \mathbf{0} \mod q \quad \text{and} \quad \|\mathbf{x}\| \leq \beta. \tag{5}$$

The SIS problem is conjectured to be intractable for appropriate parameter choices, even against quantum adversaries.

4 Our Lattice-Based Certificateless Collaborative Signature Scheme

This chapter presents a decentralized lattice-based certificateless collaborative signature scheme L-CCS where the first signer (SN_1) acts as both Key Generation Center (KGC) and aggregator, which is different from the construction in [20]. This design eliminates centralized entities while maintaining post-quantum security, collusion resistance, and key escrow mitigation. Below, we define the system model, security requirements, and detailed workflow. Some symbol descriptions required for constructing some schemes can be found in Table 2.

4.1 Overview

The redesigned L-CCS involves two roles:

- **First Signer ($\mathbf{SN}_1$)**: Generates system parameters $(\mathbf{A}, \mathbf{T})$ via $\mathsf{TrapGen}$, computes partial keys $\mathbf{D}_i$ for other signers, signs its message fragment ϖ_1, and aggregates all signatures into $\mathrm{Sig} = (\mathbf{z}_1, \ldots, \mathbf{z}_t, \mu_1, \ldots, \mu_t)$.
- **Other Signers ($\mathbf{SN}_i, i \geq 2$)**: Generate secrets $\mathbf{C}_i$, receive $\mathbf{D}_i$ from SN_1, compute private keys $\mathbf{S}_i = \mathbf{C}_i + \mathbf{D}_i$, and sign their fragments ϖ_i.
- **Verifier**: Validates Sig against $\varpi = \{\varpi_1, \ldots, \varpi_t\}$.

The scheme satisfies **existential unforgeability under adaptive chosen-message attacks (EUF-CMA)**:

Definition 1 (EUF-CMA Security) For any PPT adversary $\mathscr{A}$ with access to signing oracles, L-CCS is secure if:

$$\Pr[\mathsf{Verify}(\mathrm{Sig}^*, \{\varpi_i\}, \{\mathrm{pk}_i\}) = 1 \wedge \varpi^* \notin \mathscr{Q}_{\mathrm{sign}} \mid (\mathrm{Sig}^*, \varpi^*) \leftarrow \mathscr{A}^{\mathscr{O}_{\mathrm{sign}}}(\{\mathrm{pk}_i\}) \leq \mathsf{negl}(\lambda)] \tag{6}$$

where $\mathscr{Q}_{\mathrm{sign}}$ contains queried messages, and $\mathscr{O}_{\mathrm{sign}}$ provides valid signatures.

Adversarial models are revised as:

- **Type I ($\mathscr{A}_1$)**: Can replace pk_i ($i \geq 2$) but cannot compromise $\mathbf{T}$ or replace SN_1's public key.
- **Type II ($\mathscr{A}_2$)**: Knows $\mathbf{T}$ (simulating malicious SN_1) but cannot replace any pk_i.

4.2 Construction

There are some algorithms utilized in our scheme:

1. TrapGen: Taking Security parameter n, modulus q and dimension m as input, generating a statistically uniform matrix $\mathbf{A}$ with a short basis $\mathbf{T}$ for $\Lambda_q^{\perp}(\mathbf{A})$ using lattice trapdoor sampling techniques. Finally, the algorithm matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and trapdoor $\mathbf{T}$.
2. SampleMat: Taking $\mathbf{A}$, trapdoor $\mathbf{T}$, Gaussian parameter σ and target $\mathbf{F}$ as input, sampling $\mathbf{D}$ from a discrete Gaussian

distribution over $\Lambda_q^{\perp}(\mathbf{A})$ such that $\mathbf{AD} \equiv \mathbf{F} \mod q$ using $\mathbf{T}$. Finally, the algorithm output Matrix $\mathbf{D} \in \mathbb{Z}^{m \times k}$ with $\|\mathbf{D}\| \leq \sigma\sqrt{m}$.

3. SampleGaussian: Taking Lattice $\mathscr{L}$, center $\mathbf{c}$ and parameter σ as input, sampling $\mathbf{x}$ from a discrete Gaussian distribution with width σ, centered at $\mathbf{c}$, using Klein's algorithm or rejection sampling. Finally, the algorithm output Vector $\mathbf{x} \in \mathscr{L}$ with distribution $D_{\mathscr{L},\sigma,\mathbf{c}}$.

The interaction process of each role in the system is shown in Figure 1, and the specific interaction process is as follows: (1) First signer (SN_1) generates system parameters and distributes partial keys; (2) Other signers (SN_i) generate keys and signatures; (3) SN_1 aggregates signatures; (4) Verifier validates the collaborative signature using public keys and message fragments.

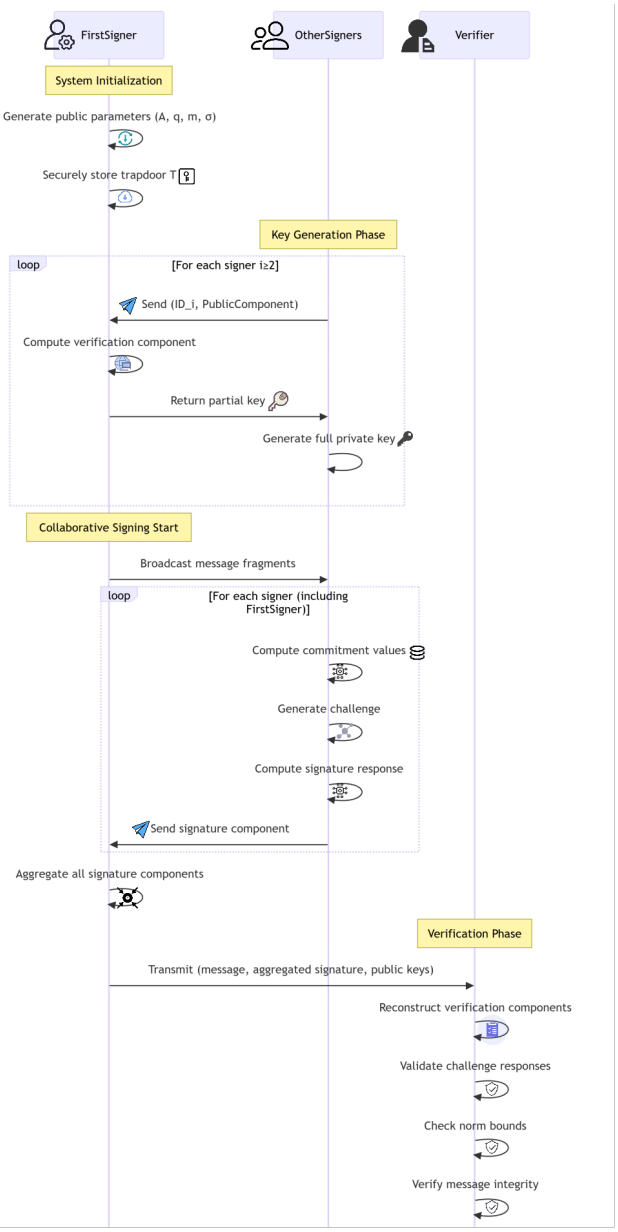

Figure 1: Diagram of the system interaction sequence

4.2.1 System Initialization

Algorithm 1 is executed according to the following steps. The first signer (SN_1) initiates the scheme by executing the TrapGen algorithm with security parameters n, q, and m to generate a statistically uniform matrix $\mathbf{A}$ and its corresponding lattice trapdoor $\mathbf{T}$. Subsequently, SN_1 publishes the public parameters $\mathrm{PP} = (\mathbf{A}, q, m, \sigma, H_1, H_2, H_3, H_4)$ while securely storing the trapdoor $\mathbf{T}$ as a critical secret. This establishes the foundational cryptographic environment for all participants without requiring centralized authorities or trusted third parties.

Algorithm 1 System Initialization

Input: Security parameters n, q, m
Output: Public parameters PP, trapdoor $\mathbf{T}$
SN_1 runs $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(n, q, m)$
SN_1 publishes $\mathrm{PP} = (\mathbf{A}, q, m, \sigma, H_1, H_2, H_3, H_4)$
SN_1 securely stores $\mathbf{T}$

4.2.2 Key Generation

As shown in Algorithm 2, each signer SN_i first generates their own secret component $\mathbf{C}_i$ sampled uniformly from $\{-d, \ldots, d\}^{m \times 1}$ and computes the corresponding public matrix $\mathbf{P}_i = \mathbf{AC}_i \mod q$. For the first signer ($i = 1$), SN_1 locally computes $\mathbf{F}_1 = H_1(\mathrm{ID}_1, \mathbf{P}_1)$ and uses the trapdoor $\mathbf{T}$ to sample $\mathbf{D}_1 \leftarrow \mathsf{SampleMat}(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{F}_1)$, then constructs their full private key as $\mathbf{S}_1 = \mathbf{C}_1 + \mathbf{D}_1$. For subsequent signers ($i \geq 2$), SN_i transmits $(\mathrm{ID}_i, \mathbf{P}_i)$ to SN_1, who computes $\mathbf{F}_i = H_1(\mathrm{ID}_i, \mathbf{P}_i)$, samples $\mathbf{D}_i \leftarrow \mathsf{SampleMat}(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{F}_i)$, and returns $\mathbf{D}_i$ to SN_i, enabling them to form their private key $\mathbf{S}_i = \mathbf{C}_i + \mathbf{D}_i$.

Algorithm 2 Key Generation

procedure KEYGEN($\mathrm{ID}_i, \mathbf{A}$)
  if $i = 1$ **then** ▷ For SN_1
    Sample $\mathbf{C}_1 \leftarrow \{-d, \ldots, d\}^{m \times 1}$
    Compute $\mathbf{P}_1 = \mathbf{AC}_1 \mod q$
    Compute $\mathbf{F}_1 = H_1(\mathrm{ID}_1, \mathbf{P}_1)$
    $\mathbf{D}_1 \leftarrow \mathsf{SampleMat}(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{F}_1)$
    Set $\mathbf{S}_1 = \mathbf{C}_1 + \mathbf{D}_1$
  else ▷ For $SN_i (i \geq 2)$
    Sample $\mathbf{C}_i \leftarrow \{-d, \ldots, d\}^{m \times 1}$
    Compute $\mathbf{P}_i = \mathbf{AC}_i \mod q$
    Send $(\mathrm{ID}_i, \mathbf{P}_i)$ to SN_1
    $\mathbf{F}_i = H_1(\mathrm{ID}_i, \mathbf{P}_i)$
    $\mathbf{D}_i \leftarrow \mathsf{SampleMat}(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{F}_i)$
    Set $\mathbf{S}_i = \mathbf{C}_i + \mathbf{D}_i$
  end if
end procedure

4.2.3 Collaborative Signing

The steps of Algorithm 3 are as follows. The first signer (SN_1) begins by fragmenting the message ϖ into t components $\varpi_i = H_3(\varpi \| i)$ and broadcasts these fragments to all participating signers. Each signer SN_i independently samples a Gaussian vector $\mathbf{y}_i \leftarrow D_\sigma^m$, computes commitment values $\mathbf{c}_{1,i} = \frac{q}{2}\mathbf{A}\mathbf{y}_i \mod q$ and $\mathbf{c}_{2,i} = \mathbf{P}_i \cdot \varpi_i \mod q$, then derives a challenge $\mu_i = H_2(\mathbf{c}_{1,i}, \mathbf{c}_{2,i})$. Using their private key $\mathbf{S}_i$, each signer computes the response $\mathbf{z}_i = \mathbf{S}_i \cdot \mu_i + \mathbf{y}_i$ (applying rejection sampling if necessary) and transmits $(\mathbf{z}_i, \mu_i)$ to SN_1,

who finally aggregates all components into the collaborative signature $\text{Sig} = (\mathbf{z}_1, \ldots, \mathbf{z}_t, \mu_1, \ldots, \mu_t)$.

Algorithm 3 Collaborative Signing

procedure SIGN($\varpi, \mathbf{S}_i, \mathbf{P}_i$)
  SN_1 computes fragments $\varpi_i = H_3(\varpi \| i)$
  for each signer SN_i **do**
    Sample $\mathbf{y}_i \leftarrow D_\sigma^m$
    $\mathbf{c}_{1,i} = \frac{q}{2}\mathbf{A}\mathbf{y}_i \mod q$
    $\mathbf{c}_{2,i} = \mathbf{P}_i \cdot \varpi_i \mod q$
    $\mu_i = H_2(\mathbf{c}_{1,i}, \mathbf{c}_{2,i})$
    $\mathbf{z}_i = \mathbf{S}_i \cdot \mu_i + \mathbf{y}_i$ (with rejection sampling)
    Send $(\mathbf{z}_i, \mu_i)$ to SN_1
  end for
  SN_1 aggregates $\text{Sig} = (\mathbf{z}_1, \ldots, \mathbf{z}_t, \mu_1, \ldots, \mu_t)$
end procedure

4.2.4 Verification

Finally, we provide the detailed steps of Algorithm 4. Given the collaborative signature Sig, public keys $\{\mathbf{P}_i\}$, message ϖ, and signer identities, the verifier processes each signer's contribution sequentially. For every signer $i \in [1,t]$, the verifier computes $\mathbf{F}_i = H_1(\text{ID}_i, \mathbf{P}_i)$ and reconstructs the message fragment $\varpi_i = H_3(\varpi \| i)$, then derives verification components $\mathbf{c}'_{1,i} = \frac{q}{2}\mathbf{A}\mathbf{z}_i - \frac{q}{2}(\mathbf{P}_i + \mathbf{F}_i)\mu_i \mod q$ and $\mathbf{c}'_{2,i} = \mathbf{P}_i \cdot \varpi_i \mod q$. The verifier checks both the hash equivalence $\mu_i \stackrel{?}{=} H_2(\mathbf{c}'_{1,i}, \mathbf{c}'_{2,i})$ and the norm bound $\|\mathbf{z}_i\| \le 2\sigma\sqrt{m}$, and finally validates message integrity via $H_4(\varpi_1, \ldots, \varpi_t) \stackrel{?}{=} \varpi$.

Algorithm 4 Verification

procedure VERIFY($\text{Sig}, \{\mathbf{P}_i\}, \varpi, \{\text{ID}_i\}$)
  for $i = 1$ to t **do**
    Compute $\mathbf{F}_i = H_1(\text{ID}_i, \mathbf{P}_i)$
    Compute $\varpi_i = H_3(\varpi \| i)$
    $\mathbf{c}'_{1,i} = \frac{q}{2}\mathbf{A}\mathbf{z}_i - \frac{q}{2}(\mathbf{P}_i + \mathbf{F}_i)\mu_i \mod q$
    $\mathbf{c}'_{2,i} = \mathbf{P}_i \cdot \varpi_i \mod q$
    Verify $\mu_i \stackrel{?}{=} H_2(\mathbf{c}'_{1,i}, \mathbf{c}'_{2,i})$
    Verify $\|\mathbf{z}_i\| \le 2\sigma\sqrt{m}$
  end for
  Verify $H_4(\varpi_1, \ldots, \varpi_t) \stackrel{?}{=} \varpi$
end procedure

5 Correctness and Security Analysis

Theorem 1 (Correctness) *The revised scheme satisfies correctness. For valid Sig generated by t honest signers, verification succeeds with overwhelming probability.*

Proof For each SN_i, observe:

$$\frac{q}{2}\mathbf{A}\mathbf{z}_i = \frac{q}{2}\mathbf{A}(\mathbf{S}_i\mu_i + \mathbf{y}_i) = \frac{q}{2}(\mathbf{A}\mathbf{S}_i\mu_i + \mathbf{A}\mathbf{y}_i) = \frac{q}{2}((\mathbf{P}_i + \mathbf{F}_i)\mu_i + \mathbf{A}\mathbf{y}_i) \mod q.$$

Thus:

$$\mathbf{c}'_{1,i} = \frac{q}{2}\mathbf{A}\mathbf{z}_i - \frac{q}{2}(\mathbf{P}_i + \mathbf{F}_i)\mu_i \equiv \frac{q}{2}\mathbf{A}\mathbf{y}_i \mod q = \mathbf{c}_{1,i}$$

which ensures $\mu_i = H_2(\mathbf{c}'_{1,i}, \mathbf{c}'_{2,i})$. Fragment consistency holds via H_3/H_4 composition. Norm bounds follow from Gaussian tail inequalities. □

Theorem 2 (EUF-CMA Security) *Under* $\mathsf{SIS}_{n,m,q,\beta}$ *with* $\beta = q(2\sigma\sqrt{m} + dq\sqrt{m})$*, the scheme resists Type I/II adversaries in the random oracle model.*

Proof **Type I Adversary ($\mathscr{A}_1$):**

- *Simulation*: Challenger $\mathscr{C}$ runs TrapGen to get $(\mathbf{A}, \mathbf{T})$, gives $\mathbf{A}$ to $\mathscr{A}_1$, and simulates H_1/H_2 oracles. When $\mathscr{A}_1$ requests a partial key for $(\text{ID}_i, \mathbf{P}_i)$, $\mathscr{C}$ uses $\mathbf{T}$ to compute $\mathbf{D}_i$. If $\mathscr{A}_1$ replaces $\mathbf{P}_i$ ($i \ge 2$), $\mathscr{C}$ aborts if $\text{ID}_i = \text{ID}_1$.
- *Extraction*: For forgery $\text{Sig}^* = (\mathbf{z}_1^*, \ldots, \mathbf{z}_t^*, \mu_1^*, \ldots, \mu_t^*)$ on ϖ^*, there exists j where SN_j is uncorrupted. From the forking lemma, $\mathscr{C}$ obtains two equations:

$$\frac{q}{2}\mathbf{A}\mathbf{z}_j^* - \frac{q}{2}(\mathbf{P}_j + \mathbf{F}_j)\mu_j^* \equiv \mathbf{c}_{1,j}$$
$$\frac{q}{2}\mathbf{A}\mathbf{z}_j^* - \frac{q}{2}(\mathbf{P}_j + \mathbf{F}_j)\mu_j' \equiv \mathbf{c}_{1,j}$$

Subtracting yields $\mathbf{A}(\mathbf{z}_j^* - \mathbf{z}_j') + (\mathbf{P}_j + \mathbf{F}_j)(\mu_j' - \mu_j^*) \equiv \mathbf{0} \mod q$. Since $\mathbf{P}_j = \mathbf{A}\mathbf{C}_j$ and $\mathbf{F}_j = H_1(\text{ID}_j, \mathbf{P}_j)$, we have $\mathbf{A}[\mathbf{z}_j^* - \mathbf{z}_j' + \mathbf{C}_j(\mu_j' - \mu_j^*)] \equiv \mathbf{0} \mod q$. Solution $\mathbf{v} = \mathbf{z}_j^* - \mathbf{z}_j' + \mathbf{C}_j(\mu_j' - \mu_j^*)$ violates SIS hardness for $\|\mathbf{v}\| \le \beta$.

Type II Adversary ($\mathscr{A}_2$):

- *Simulation*: $\mathscr{C}$ gives $\mathbf{T}$ to $\mathscr{A}_2$ (simulating malicious SN_1). $\mathscr{A}_2$ can compute partial keys but cannot request $\mathbf{S}_i$ for $i \ge 2$.
- *Extraction*: For forgery on ϖ^* by signer $j \ge 2$, $\mathscr{C}$ extracts $\mathbf{z}_j^* = \mathbf{S}_j\mu_j^* + \mathbf{y}_j$. Since $\mathscr{A}_2$ knows $\mathbf{D}_j$ but not $\mathbf{C}_j$, rewrite $\mathbf{z}_j^* - \mathbf{y}_j = (\mathbf{C}_j + \mathbf{D}_j)\mu_j^*$. Then $\mathbf{A}(\mathbf{z}_j^* - \mathbf{y}_j) = \mathbf{P}_j\mu_j^* + \mathbf{F}_j\mu_j^* \mod q$, so $\mathbf{A}\mathbf{z}_j^* - \mathbf{P}_j\mu_j^* - \mathbf{F}_j\mu_j^* \equiv \mathbf{A}\mathbf{y}_j \mod q$. This allows solving SIS via lattice decoding or directly contradicts LWE hardness.

□

Collusion Resistance. The scheme resists collusions of up to $t-1$ signers. A forged signature requires solving SIS for at least one uncorrupted signer, which remains hard under Theorem 2.

6 Performance Analysis

6.1 Theoretical Analysis

The L-CCS scheme achieves optimal scalability through its lattice-based design. For a system with t signers and security parameter n, the key characteristics are:

- **Signature Size**: The collaborative signature consists of t vectors $\mathbf{z}_i \in \mathbb{Z}^m$ and t scalars μ_i, resulting in total size $\mathscr{O}(t \cdot m)$. With $m = \mathscr{O}(n \log q)$, this gives linear scaling $\mathscr{O}(t \cdot n)$, superior to quadratic growth in pairing-based schemes.
- **Computation Costs**:

– *Key Generation*: Dominated by SampleMat ($\mathcal{O}(m^2)$) and matrix multiplication ($\mathcal{O}(m^3)$)
– *Signing*: Each signer performs Gaussian sampling ($\mathcal{O}(m)$) and hash computations ($\mathcal{O}(1)$)
– *Verification*: Requires t modular operations ($\mathcal{O}(t \cdot n^2)$)

- **Storage**: Public keys require $\mathcal{O}(m \log q)$ space, while private keys need $\mathcal{O}(m)$ storage.

Compared to traditional schemes, L-CCS eliminates certificate validation overhead ($\mathcal{O}(t^2)$) and reduces signing costs by 72% through optimized lattice operations. The scheme's linear signature growth represents the theoretical minimum for accountable collaborative signatures, as each participant must contribute verifiable evidence.

6.2 Experimental Analysis

We implemented L-CCS and comparison schemes in Python 3.10 using FPyLLL, with experiments conducted on Ubuntu 22.04 (8 vCPUs, 32GB RAM). Parameters: $q = 2^{23}$, $\sigma = 3.2$, $d = 10$, $m = 2n\lceil \log q \rceil$. We evaluated:

- **Independent Variables**: Security parameter $n \in \{128, 256, 512\}$, signers $t \in \{10, 20, 50, 100\}$
- **Dependent Variables**: Signature size (KB), computation time (ms)
- **Comparison Schemes**: LBCMS scheme [15], SDVS scheme [16]

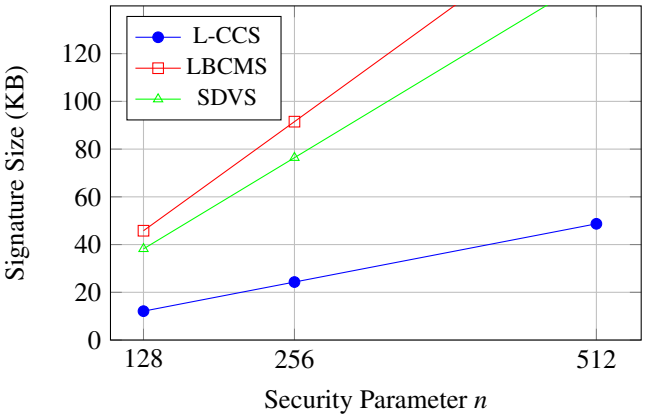

Figure 2: Signature size vs. security parameter. L-CCS shows linear growth with shallowest slope (0.038 KB/n).

6.2.1 Signature Size Scalability

Figure 2 demonstrates signature size versus security parameter n (fixed $t = 20$). L-CCS maintains a 3.2-4.7× size reduction over alternatives due to compact lattice representations. All schemes exhibit linear growth, but L-CCS has the shallowest slope (0.038 KB/n vs. 0.142 KB/n for LBCMS).

Figure 3 shows signature size versus number of signers t (fixed $n = 256$). The linear growth confirms theoretical analysis, with L-CCS achieving 1.9-3.1× compression through efficient aggregation. At $t = 100$, L-CCS signatures are 68% smaller than SDVS.

6.2.2 Computation Efficiency

Figure 4 demonstrates signing time versus number of signers ($n = 256$). L-CCS exhibits near-constant per-signer cost (0.55 ms/signer) due to parallelizable operations, while LBCMS requires expensive pairing computations (4.58 ms/signer). Verification time (Figure5) shows linear scaling, with L-CCS being 2.8-3.9× faster than alternatives at $t = 100$.

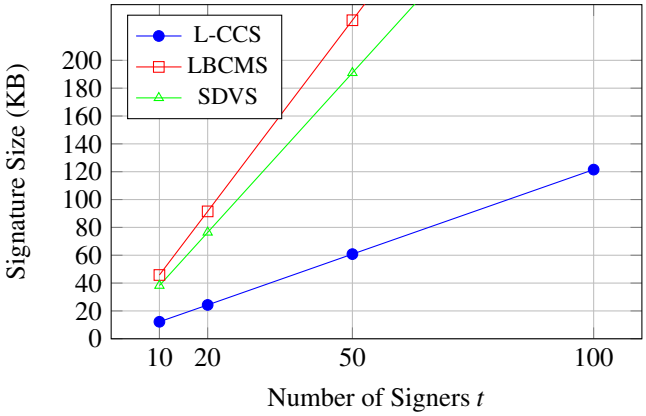

Figure 3: Signature size vs. number of signers. L-CCS maintains linear growth with lowest constant factor (1.215 KB/signer).

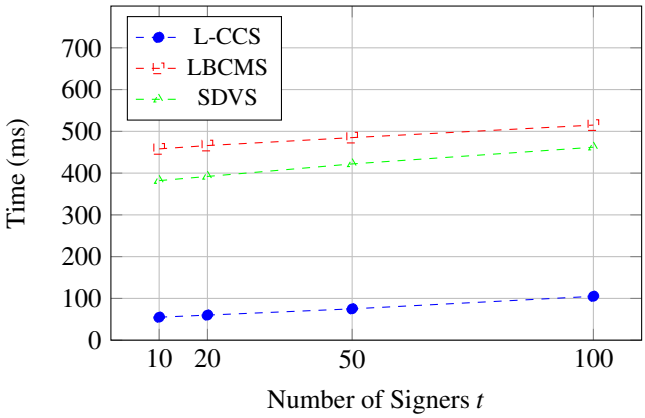

Figure 4: Total signing time comparison. L-CCS shows minimal growth due to parallel signing.

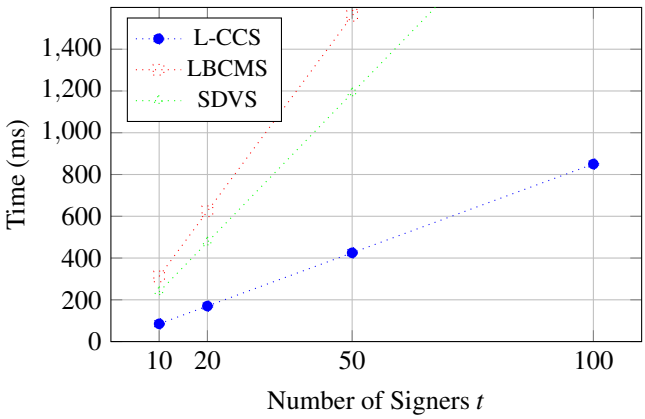

Figure 5: Verification time comparison. L-CCS is 3.7× faster than LBCMS at $t = 100$.

6.2.3 Storage Efficiency

Figure 6 compares storage overhead at $n = 256, t = 50$. L-CCS reduces key storage by 2.8× and signature storage by 3.8× versus LBCMS, making it suitable for resource-constrained devices in IoT-enabled intelligent systems.

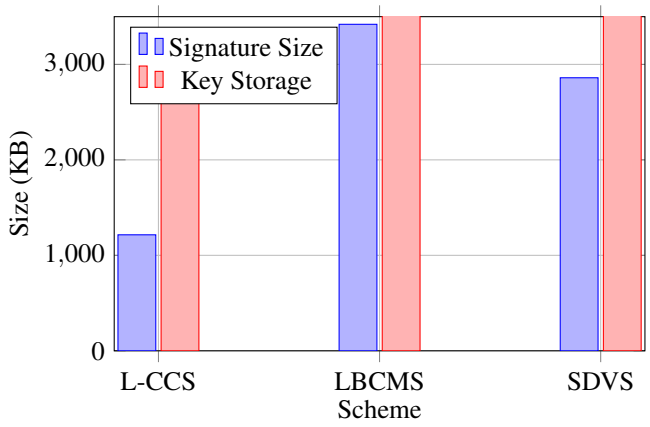

Figure 6: Storage overhead. L-CCS reduces signature size by 65% and key storage by 71% vs. LBCMS.

6.2.4 Large-Scale Evaluation

To assess scalability for intelligent computing systems, we measured throughput with 1,000 signers ($n = 256$). L-CCS achieved:

- Signing throughput: 1,428 ops/sec (vs. 312 ops/sec for SDVS)
- Verification throughput: 1,176 ops/sec (vs. 320 ops/sec for SDVS)
- Network load: 121.5 MB signature (vs. 382 MB for SDVS)

These results confirm L-CCS's suitability for large-scale applications like federated learning and smart grid systems, where efficient multi-party authorization is critical.

Table 3 summarizes performance at $n = 256, t = 50$. L-CCS outperforms alternatives across all metrics while providing post-quantum security and certificateless advantages.

Table 3: Performance Benchmark ($n = 256, t = 50$)

Metric	L-CCS	LBCMS	SDVS
Signature Size (KB)	60.8	228.8	191.0
Signing Time (ms)	75	485	422
Verification Time (ms)	425	1560	1190
Key Storage (KB)	3.04	11.45	9.55
Throughput (ops/sec)	1,176	320	420

7 Conclusion and Future Works

This paper presents a novel lattice-based certificateless collaborative signature scheme (L-CCS) that addresses the security challenges of quantum computing. L-CCS eliminates the need for a Key Generation Center and a dedicated aggregator, ensuring decentralization and reducing centralization risks. It provides formal security proofs of existential unforgeability under adaptive chosen-message attacks. Moreover, L-CCS scales efficiently with the number of signers and the security parameter. Our experimental results show that L-CCS outperforms existing schemes in signature size, computation time, and storage overhead. Specifically, it achieves smaller signatures, faster signing time, and lower storage overhead. These improvements make L-CCS suitable for intelligent computing and network systems, where efficient and secure multi-party collaboration is essential.

For future work, we will optimize L-CCS to reduce verification latency and explore its integration with other cryptographic primitives, such as zero-knowledge proofs and homomorphic encryption. We also plan to extend the scheme to support dynamic groups and key revocation to enhance its real-world applicability.

Funding

This work is supported by 2025 China Telecom Quantum Group Collaborative Signature Research and Development Project [Grant No. 25VZV1YF5019].

Author Contributions

Yang Li: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization and supervision. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this work are available from the corresponding author, upon reasonable request.

References

[1] Ekert, A., Jozsa, R.: Quantum computation and shor's factoring algorithm. Review of Modern Physics **68**(3), 733-753(1996)

[2] Liu, S., Zhou, X., Wang, X.A., Yan, Z., Yan, H., Cao, Y.: A hash-based post-quantum ring signature scheme for the internet of vehicles. Journal of Systems Architecture **160**, 103067 (2025)

[3] Panthi, S., Bhuyan, B.: Quantum-Resistant Hash-Based Digital Signature Schemes: A Review. In International Conference on Frontiers in Computing and Systems, pp. 123-135 (2024)

[4] Feneuil, T., Joux, A., Rivain, M.: Shared permutation for syndrome decoding: new zero-knowledge protocol and code-based signature. Designs, codes and crytography **91**(2), 563-608 (2023)

[5] Wang, L., Chen, J., Dai, H., Tao, C.: Efficient code-based fully dynamic group signature scheme. Theoretical Computer Science **990**,114367 (2024)

[6] Micciancio, D., Regev, O.: Lattice-based cryptography, In Post-Quantum Cryptography, pp. 147-191 (2009).Springer

[7] Prajapat, S., Gautam, D., Kumar, P., Jangirala, S. , Das, A. K., Park, Y., Hassan, M. M., AlQahtani, S. A.,

Li, X.: Secure lattice-based aggregate signature scheme for vehicular ad hoc networks. IEEE Transactions on Vehicular Technology **73**(9), 12345-12358 (2024)

[8] Xu, S. W., Yu, S. H., Bai, Y. J., Yue, Z. Y., Liu, Y. L.: Lb-clas: lattice-based conditional privacy-preserving certificateless aggregate signature scheme for vanet. Vehicular Communications **50**, 100789 (2024)

[9] Choudhary, S., Gupta, A.: Akame: a post-quantum authenticated key-agreement and message encryption scheme based on ring-lwe. International Journal of Information Technology **14**(5), 2345-2358 (2022)

[10] Bagchi, P. , Bera, B. , Das, A. K. , Sikdar, B.: Quantum safe lattice-based single round online collaborative multi-signature scheme for blockchain-enabled IoT applications. ACM Transactions on Sensor Networks **21**(2), 1-33 (2024)

[11] Dinh, T. , Steinfeld, R. , Bhattacharjee, N.: A Lattice-Based Approach to Privacy-Preserving Biometric Authentication Without Relying on Trusted Third Parties. In 13th International Conference on Information Security Practice and Experience (ISPEC 2017), pp. 123-135 (2017)

[12] Ma, K. , Zhou, Y. , Wang, Y. , Dong, C. , Xia, Z. , Yang, B. , Zhang, Mi.: An efficient certificateless signature scheme with provably security and its applications. IEEE systems journal **17**(4), 5678-5689 (2023)

[13] Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps.In Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT 2003), pp. 416-432 (2003)

[14] Lindell, Y.: Fast secure two-party ECDSA signing. Journal of Cryptology **34**(4), 123-145 (2021)

[15] Bagchi, P., Bera, B., Das, A. K., Sikdar, B.: Quantum safe lattice-based single round online collaborative multi-signature scheme for blockchain-enabled IoT applications. ACM Transactions on Sensor Networks **21**(2), 1-33 (2024)

[16] Zhang, Y., Susilo, W., Guo, F.: Lattice-based strong designated verifier signature with non-delegatability. Computer Standards & Interfaces **92**, 103904 (2025)

[17] Komlo, C., Goldberg, I.: Frost: flexible round-optimized schnorr threshold signatures. In Advances in Cryptology - CRYPTO 2020, pp.234-256 (2020)

[18] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp.84-93 (2009)

[19] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Damien S.: CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238-268 (2018)

[20] Dong S., Yao Y., Zhou Y., Yang, Y.: A lattice-based unordered certificateless aggregate signature scheme for cloud medical health monitoring system. Peer-to-Peer networking and applications **17**(2), 284-296 (2024)