BILINGUAL PUBLISHING GROUP

Journal of Electronic & Information Systems

https://journals.bilpubgroup.com/index.php/jeis

ARTICLE

P-CSNKS: Post-Quantum Collaborative Signature Scheme with Non-Linear Private Key Splitting Technique

Fei Long*, *Yang Li*

China Telecom Quantum Technology Co., Ltd, Hefei 230088, China

ABSTRACT

Traditional collaborative signature schemes face significant challenges in resisting quantum computing attacks, securing private keys in distributed architectures, and balancing operational efficiency, which are critical requirements for modern electronic and information systems like IoT, blockchain, and federated learning. This paper proposes P-CSNKS, a novel post-quotum collaborative signature scheme featuring a non-linear private key splitting technique. Unlike linear secret sharing, P-CSNKS partitions the master private key into multiple interdependent subkeys using multiplicative inverses and modular arithmetic, ensuring algebraic interdependencies prevent full key reconstruction even if attackers compromise sufficient shares. Simultaneously, the scheme embeds hash-based post-quantum signature components directly into the collaborative ECDSA signing workflow. This hybrid design maintains backward compatibility with standard ECDSA verification while establishing dual security layers: one for classical security and another providing provable existential unforgeability against quantum adversaries in the quantum random oracle model. Crucially, P-CSNKS achieves this quantum resistance without incurring prohibitive computational costs. Rigorous experimental evaluations demonstrate that P-CSNKS significantly outperforms lattice-based while also showing efficiency gains against hash-based scheme. The optimized algorithms for key generation, signing, and verification ensure lightweight performance suitable for latency-sensitive applications. Thus, P-CSNKS delivers enhanced security against both classical and quantum threats while meeting the stringent efficiency demands of next-generation distributed systems.

Keywords: Collaborative Signature; Post-Quantum; Quantum Computing Attack; Private Key Splitting

*CORRESPONDING AUTHOR:

Yang Li, China Telecom Quantum Technology Co., Ltd, Hefei 230088, China ; Email: liyang2@chinatelecom.cn

ARTICLE INFO

Received: 25 May 2025 | Revised: 16 June | Accepted: 23 June 2025 | Published Online: 16 July 2025

DOI: https://doi.org/10.30564/jeis.v7i2.10217

CITATION

Long, F.,Li, Y., 2025. P-CSNKS: Post-Quantum Collaborative Signature Scheme with Non-Linear Private Key Splitting Technique. Journal of Electronic & Information Systems. 7(2): 1–12. DOI: https://doi.org/10.30564/jeis.v7i2.10217

1. Introduction

In the era of intelligent electronic systems and ubiquitous information networks, securing distributed architectures, such as the Internet of Things (IoT), blockchain platforms, and federated learning frameworks, has become a critical challenge. These systems rely heavily on digital signature techniques to ensure authentication, data integrity, and non-repudiation across heterogeneous devices and communication channels. However, the rapid evolution of quantum computing poses an existential threat to classical cryptographic schemes, particularly in resource-constrained environments such as embedded circuits, distributed sensor networks, and edge computing nodes. Traditional digital signature algorithms depend on discrete logarithm assumptions that are vulnerable to Shor's algorithm [1], jeopardizing the security foundations of modern electronic and information systems.

Collaborative signature schemes, designed to decentralize signing authority across multiple parties, offer a promising solution to mitigate single points of failure in distributed systems, as shown in **Figure 1**. Over the past decade, significant efforts have been devoted to developing collaborative variants of ECDSA and SM2 [2–4], aiming to strike a balance between efficiency and decentralized trust. Nevertheless, existing solutions face critical limitations. First, most schemes adopt linear secret sharing [5] for private key splitting, where the master key can be reconstructed through simple linear combinations of subkeys. This approach exposes systems to collusion risks and partial key compromise, as intercepting a sufficient number of shares directly reveals the secret. Second, while post-quantum collaborative signature schemes [6,7] have been proposed to address quantum vulnerabilities, their computational overhead—stemming from complex polynomial operations and large key sizes—renders them impractical for latency-sensitive applications. Third, current collaborative frameworks often neglect seamless integration of post-quantum components, resulting in either incompatibility with legacy systems or weakened security guarantees.

The advent of quantum computing exacerbates these challenges. Quantum algorithms, such as Grover's and Shor's [1], threaten to break classical cryptographic hardness assumptions within polynomial time, necessitating urgent adoption of quantum-resistant techniques. Among post-quantum candidates, hash-based signatures [8,9] offer provable security rooted in the collision resistance of cryptographic hash functions, a property considered robust even against quantum adversaries. However, integrating such mechanisms into collaborative signing protocols while maintaining efficiency remains an open problem.

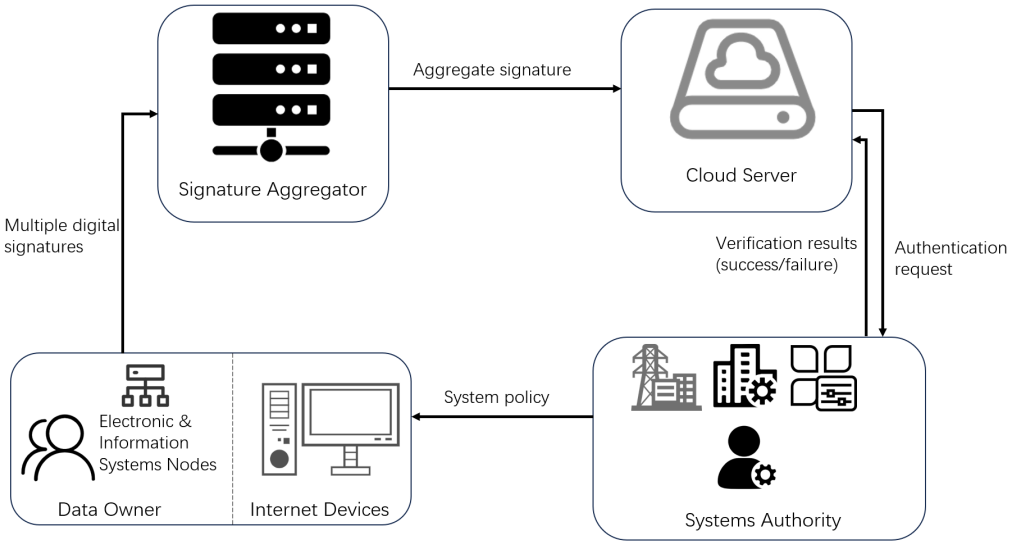

Figure 1. Data protection model in electronic and information systems.

To address these challenges, we propose P-CSNKS, a post-quantum collaborative signature scheme that synergizes non-linear private key splitting with hash-based quantum resistance. Unlike linear secret sharing, our non-linear segmentation mechanism employs multiplicative inverses and modular arithmetic to partition the master private key into interdependent subkeys, ensuring that partial compromises cannot lead to full key recovery. Simultaneously, we embed hash-based signature components into the collaborative signing workflow, achieving dual verification layers: one for classical ECDSA compatibility and another for post-quantum security. This hybrid design not only resists quantum attacks but also retains operational efficiency.

Figure 2 illustrates the technical roadmap of the scheme proposed in this paper. The roadmap outlines two primary development paths addressing critical requirements. The first path focuses on ensuring the integrity and security of the private key, with the objective of preventing partial reconstruction of the private key, achieved through the use of non-linear private key segmentation technology. The second path addresses the requirement for quantum resistance and compatibility, to achieve post-quantum security without loss of compatibility, realized by embedding a hash-based post-quantum signature. These two paths are interconnected through Mutual Improvement, highlighting the synergistic design of the scheme. In summary, our contributions include the following three points:

1. Non-Linear Private Key Splitting: A novel segmentation technique that thwart partial key reconstruction through algebraic interdependencies among subkeys, enhancing resilience against collusion and side-channel attacks.

2. Post-Quantum Collaborative Signing: A protocol integrating hash-based signatures into the ECDSA framework, enabling dual verification without sacrificing compatibility with existing infrastructure.

3. Practical Efficiency: Optimized algorithms for key generation, signing, and verification, validated through rigorous experimentation to outperform state-of-the-art post-quantum solutions.

The proposed scheme not only enhances security against quantum and classical threats but also achieves practical efficiency, making it deployable in latency-sensitive electronic and information systems.

Organization: This paper begins with an introduction that provides background, motivation, and an overview of the P-CSNKS scheme. Chapter 2 covers preliminaries, including the ECDLP assumption, two-party collaborative signature, and hash-based post-quantum signature. Chapter 3 presents the security model, focusing on EUF-CMA security and additional goals. Chapter 4 details the scheme's detailed construction, including key generation, collaborative signing, and verification processes. Chapter 5 includes correctness and security analysis with a proof of existential unforgeability. Chapter 6 evaluates performance by comparing computational metrics with those of existing schemes. Finally, Chapter 7 concludes the paper and suggests future work.

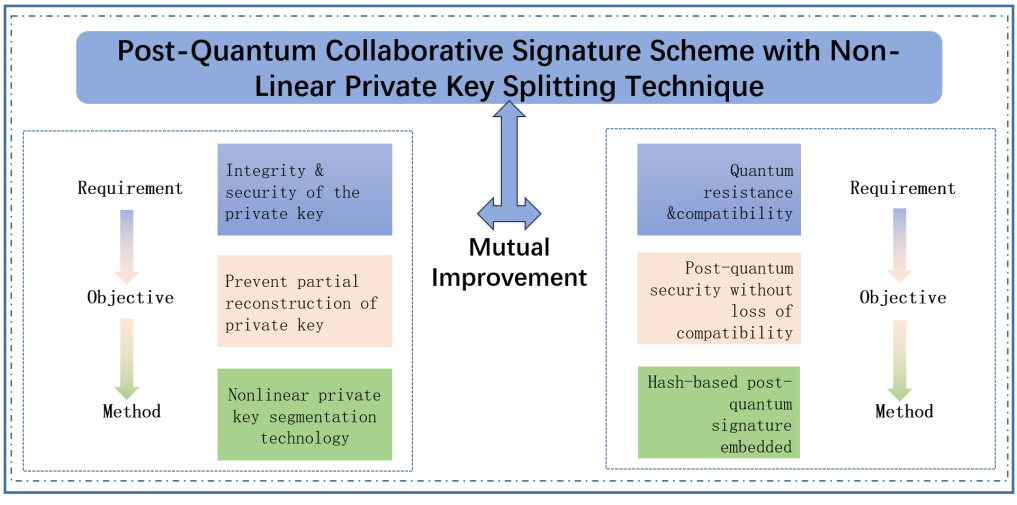

Figure 2. Technology roadmap of P-CSNKS.

2. Preliminaries

2.1. Elliptic Curve and ECDLP Assumption

To provide a comprehensive understanding of the elliptic curve cryptography (ECC) foundation for the P-CSNKS scheme, this supplementary section delves deeper into the mathematical constructs and properties of elliptic curves over finite fields, with a particular emphasis on elements directly relevant to the subsequent formulas and security assumptions utilized in the paper. Concretely, an elliptic curve E over a field $\mathbb{F}_p$ (where p is a large prime number) is defined by the equation: $y^2 \equiv x^3 + ax + b \bmod p$, where $a, b \in \mathbb{F}_p$ are curve parameters satisfying the condition $4a^3 + 27b^2 \neq 0 \bmod p$. This condition ensures that the curve is non-singular, meaning it has no cusps or self-intersections, which is essential for cryptographic applications.

The set of points $E(\mathbb{F}_q)$ on the elliptic curve consists of all pairs $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ that satisfy the curve equation, along with a special point $\mathcal{O}$, referred to as the point at infinity. This point at infinity serves as the identity element for the elliptic curve group operation. The elliptic curve group operation, often called point addition, is defined as follows:

1. Identity Element: For any point $P \in E(\mathbb{F}_q)$, $P + \mathcal{O} = \mathcal{O} + P = P$.

2. Inverse Element: For each point $P = (x, y) \in E(\mathbb{F}_q)$, there exists a point $-P = (x, -y \bmod p)$ such that $P + (-P) = \mathcal{O}$.

3. Point Addition: Let $P, Q \in E(\mathbb{F}_q)$ be two distinct points. The sum $R - P + Q$ is computed geometrically by drawing a line through P and Q, finding the third intersection point R' with the curve, and then reflecting R' over the x-axis to obtain R. Algebraically, this is expressed as:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P} \bmod p,$$

$$x_R = \lambda^2 - x_P - x_Q \bmod p,$$

$$y_R = \lambda(x_P - x_R) - y_R \bmod p.$$

4. Point Doubling: When adding a point P to itself ($R = P + P = 2P$), the process is called point doubling. The formula for point doubling is:

$$\lambda = \frac{3x_P^2 + a}{2y_P} \bmod p,$$

$$x_R = \lambda^2 - 2x_P \bmod p,$$

$$y_R = \lambda(x_P - x_R) - y_P \bmod p.$$

These operations endow $E(\mathbb{F}_q)$ with an abelian group structure, where the group operation is associative, commutative, and has an identity element and inverse elements.

Definition 1 (ECDLP Assumption) Let E be an elliptic curve defined over a finite field $\mathbb{F}_p$, where p is a large prime. Let G denote a cyclic subgroup of $E(\mathbb{F}_q)$ with prime order n. The Elliptic Curve Discrete Logarithm Problem (ECDLP) Assumption is defined as follows: Given two points P and $Q = k \cdot G$ on E, where $k \in [1, n-1]$, it is computationally infeasible to determine the integer k. The hardness of ECDLP forms the foundational security assumption for elliptic curve-based cryptographic schemes, including ECDSA. This assumption asserts that no polynomial-time adversary can solve ECDLP with non-negligible probability under classical computational models.

2.2. Two Party Collaborative Signature

Taking the collaborative signature scheme based on ECDSA digital signature [10] as an example. A two-party collaborative signature scheme extends the classical ECDSA protocol to distribute the signing authority between two entities (e.g., a client and a server). The scheme comprises three polynomial-time algorithms defined at a high level as follows:

(i) $(sk_{share}, pk) \leftarrow \mathbf{KenGen}(1^{\lambda})$:

Input: Security parameter λ.

Output: A set of private key shares $sk_{share} = (sk_1, sk_2)$ and public key pk.

The master private key is generated and split into two shares using a secure secret-sharing mechanism. The corresponding public key $pk_i = sk_i \cdot G$ is computed and published.

(ii) $\sigma \leftarrow \mathbf{Sign}(M, \{sk_i\}_{i \in \{0,1\}})$:

Input: Message M, private key shares sk_1 (held by the client) and sk_2 (held by the server).

Output: A collaborative signature σ.

The client and server interactively compute partial

signatures using their respective shares. The client generates a nonce k_1 , computes r as the x-coordinate of the elliptic curve point $Q = k_1 \cdot G$. The server uses sk_2 to compute intermediate values s_1 , s_2. The client then computes s using s_1, s_2. The final signature $\sigma = (r, s)$ is derived through secure combination of these components.

(iii) $\{0,1\} \leftarrow$ **Verify**(M, σ, pk):

Input: Message M, signature σ, and public key pk.

Output: A bit $b \in \{0,1\}$, where 1 indicates validity.

The verifier reconstructs the elliptic curve point R from r and s. The signature is valid if $R.x \bmod n = r$.

2.3. Hash-Based Post-Quantum Signature

Hash-based post-quantum signatures represent a class of cryptographic methods that leverage the inherent security properties of cryptographic hash functions to resist attacks from quantum computers. These schemes are fundamentally different from traditional digital signatures, which often rely on the difficulty of mathematical problems, such as integer factorization or discrete logarithms—problems that quantum algorithms, like Shor's, can efficiently solve. Instead, hash-based signatures derive their security from the collision resistance and preimage resistance of hash functions, properties that are currently believed to be robust even against quantum adversaries.

At the core of hash-based signatures is the concept of hash chains. A hash chain is constructed by iteratively applying a cryptographic hash function to an initial secret value, creating a sequence of hash values. The beauty of this approach lies in its simplicity and security. While it is easy to compute each subsequent hash value in the chain given the previous one, it is computationally infeasible to reverse the process and determine earlier values in the chain from later ones. This one-way property ensures that even if a portion of the hash chain is exposed, the entire chain—and thus the underlying secret key—remains secure.

One of the most well-known hash-based signature schemes is the Lamport one-time signature (OTS) scheme. In a Lamport OTS, the private key consists of multiple pairs of random values, and the public key is derived by hashing each of these values. To sign a message, the signer reveals a subset of the private key values corresponding to the bits of the message hash. The security of this scheme stems from the fact that each private key can only be used to sign a single message; reusing the key would compromise security. While Lamport OTS is simple and secure, its drawback is the relatively large size of the public key and signatures, making it less practical for some applications.

To address these limitations, more advanced hash-based signature schemes have been developed, such as the Merkle signature scheme and the SM3-OTS scheme [11] mentioned in our paper. The Merkle scheme introduces a hierarchical structure, allowing for the signing of multiple messages with a single public key while maintaining security. It achieves this by combining multiple OTS keys under a single public key using a Merkle tree, a binary tree where each leaf node contains the hash of an OTS public key and each internal node is the hash of its children. The root of the tree serves as the master public key. When signing a message, the signer reveals the OTS public key used for that specific signature, along with the authentication path (a set of hash values) that connects the OTS public key to the root of the tree. This approach significantly reduces the size of the public key and enables the signing of multiple messages, albeit with some increase in signature size and computational overhead.

The SM3-OTS scheme, which we reference in our paper, builds upon these principles but incorporates the SM3 hash function, a Chinese standard cryptographic hash function. It follows a similar approach to other hash-based OTS schemes but is optimized for performance and security in specific environments. In SM3-OTS, the master private key is split into multiple secret blocks, each of which is processed through a hash chain of a fixed length. When signing a message, specific nodes from these hash chains are selected based on the hash digest of the message. Verification involves reconstructing the public key fragments through complementary hash iterations. This design ensures that even if part of the hash chain is compromised, the entire key remain intact, thereby providing strong security guarantees.

3. Security Model

The security of P-CSNKS is formalized via a game between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$:

Setup: $\mathcal{C}$ generates system parameters (G, n), splits sk into (sk_1, sk_2, sk_3) using non-linear private key splitting technique and publishes $pk = sk \cdot G$.

Queries: $\mathcal{A}$ adaptively requests signatures on message M_i.$\mathcal{C}$ simulates the collaborative signing protocol and hash-based post-quantum signature mechanism, returning valid signatures σ_i.

Forgery: $\mathcal{A}$ outputs a forgery σ^* on a message $M^* \notin \{M_i\}$.

The adversary wins the game above if the signature σ^* is a legitimate signature of the message M. Our P-CSNKS achieves existential unforgeability under adaptive chosen-message attacks (EUF-CMA) if no PPT adversary $\mathcal{A}$ wins with non-negligible probability.

Additional security goals include keyword secrecy and quantum resistance.

(i) Key Secrecy: Even if $\mathcal{A}$ corrupts the client or the server, it cannot recover sk from leaked subkeys sk_1, sk_2 or sk_3.

(ii) Quantum Resistance: Hash-based components in hash-based post-quantum signature mechanism remain secure against quantum adversaries, assuming the used hash function H is a quantum-random oracle.

4. Constructions

This chapter will give the detailed structure of the scheme. The flow chart of interaction between roles is shown in **Figure 3**, which depicts the three main phases of interaction: key generation, where the master private key is split into subkeys; collaborative signing, where the client and server interactively compute the signature components; and signature verification, where the validity of the signature is confirmed using reconstructed public key fragments.

4.1. Key Generation

The system initializes by selecting elliptic curve parameters (G, n), where G is the base point an n is the curve order. A master private key $sk \in [1, n]$ is generated, and its corresponding public key $pk = sk \cdot G$ is computed. Using a non-linear segmentation mechanism, sk is split into three subkeys sk_1, sk_2, sk_3 satisfying **Equation (1)**:

$$sk \equiv \left(sk_2 \cdot sk_1^{-1} + sk_2 \cdot sk_3^{-1}\right) \bmod n. \quad (1)$$

where sk_1, sk_3 are non-zero random values. Subkey sk_1 is securely distributed to the client, while sk_2 and sk_3 are allocated to the server, and sk is irrevocably deleted to eliminate central storage risks①. The public key pk is broadcast globally.

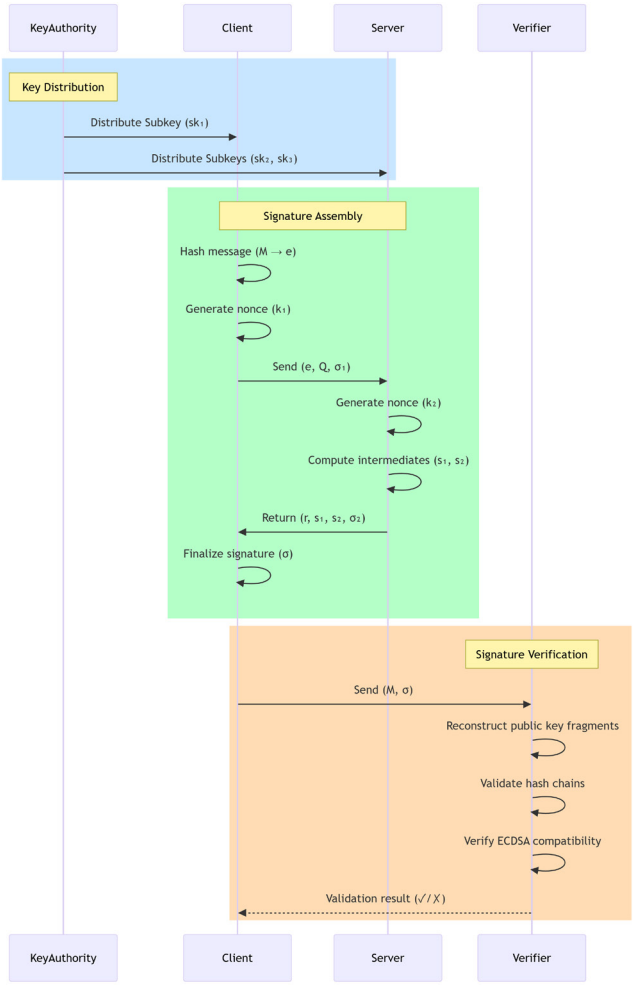

Figure 3. Flow chart of interaction between roles in the system.

4.2. Collaborative Signing

To sign a message M, the client and the server engage in a dynamic three-phase protocol:

Phase 1 (Client): The client hashes M to obtain $e = H(M)$, generates a random nonce $k_1 \in [1, n - 1]$ and com-

① Secure deletion follows industry standards (e.g., NIST SP 800-88) enforced via trusted hardware modules.

putes $Q = k_1 \cdot G$. with $r = Q.x \bmod n$. Using sk_1, the client calls Algorithm 1 to generate a hash-based post-quantum signature component σ_1. The tuple (e, Q, σ_1) is sent to the server.

Phase 2 (Server): The server generates $k_2 \in [1, n-1]$, computes intermediates $s_1 = k_2 \cdot (sk_3 \cdot e + r \cdot sk_3) \bmod n$ and $s_2 = k_2 \cdot sk_2 \cdot sk_3 \cdot r \bmod n$, and generates its post-quantum component σ_2 by calling Algorithm 1 with input sk_2, M, r similar to the SM3-OTS scheme [11]. The server returns (r, s_1, s_2, σ_2) to the client.

Phase 3 (Client): The client computes $s = k_1^{-1} (s_1 + sk_1^{-1} \cdot s_2) \bmod n$ then finalizes the signature as $(r, s, \sigma_1, \sigma_2)$.

4.3. Signature Verification

As shown in **Equations (2)** and **(3)**, the verifier reconstructs the client's and the server's public key fragments pk_1' and pk_2' by applying residual hash iterations to σ_1 and σ_2.

$$pk_1' = \{H^{255-a_0}(\sigma_{1,0}), \cdots\cdots, H^{255-a_{31}}(\sigma_{1,31}), H^{255-b_0}(\sigma_{1,32}), \cdots\cdots, H^{255-b_{15}}(\sigma_{1,47})\}. \quad (2)$$

$$pk_2' = \{H^{255-a_0}(\sigma_{2,0}), \cdots\cdots, H^{255-a_{31}}(\sigma_{2,31}), H^{255-b_0}(\sigma_{2,32}), \cdots\cdots, H^{255-b_{15}}(\sigma_{2,47})\}. \quad (3)$$

If pk_1' and pk_2' are consistent with pre-broadcast pk_1 and pk_2 confirms the legitimacy of the post-quantum components. Then, the verifier computes $u_1 = e \cdot s^{-1} \bmod n$ and $u_2 = r \cdot s^{-1} \bmod n$, derives $R' = u_1 \cdot G + u_2 \cdot pk$. The signature is valid iff $R'.x \bmod n = r$.

Algorithm 1 Hash-Based Post-Quantum Signature Generation

Input: Secret key sk_i ($i \in \{0, 1\}$), Message M, The abscissa r of point Q on the elliptic curve.
Output: Signature σ_i ($i \in \{0, 1\}$).

1: Split sk_i as $\{sk_{i,0}, sk_{i,1}, \cdots\cdots, sk_{i,47}\}$;
2: Iteratively compute 255 hash-chain layers for each of the 47 secret key segments as
3: $pk_i = \{H^{255}(sk_{i,0}), H^{255}(sk_{i,1}), \cdots\cdots, H^{255}(sk_{i,47})\}$, then broadcast the pk_i;
4: Calculate the joint hash value $h = H(M \| r)$ then encode h into
5: binary and hexadecimal forms, respectively;
6: Convert binary form of h to 32 decimal numbers $a_0, \cdots\cdots, a_{31}$ and
7: sum the position index of each character in hexadecimal form and modulo 255 to obtain 16 step sizes $b_0, \cdots\cdots, b_{15}$;
8: Compute signature $\sigma_i = \{H^{a_0}(sk_{i,0}), \cdots\cdots, H^{a_{31}}(sk_{i,31}), H^{b_0}(sk_{i,32}), \cdots\cdots, H^{b_{15}}(sk_{i,47})\}$.

5. Correctness and Security Analysis

5.1. Correctness Analysis

The correctness relies on the deterministic reconstruction of public key fragments (pk_1, pk_2) and the algebraic consistency of the non-linear private key splitting mechanism. Firstly, If σ_1 and σ_2 are generated correctly, applying $H^{255-aj}(\sigma_{i,j})$ recovers the pre-broadcast pk_i, validating the hash chains. Secondly, during collaborative signing, the client computes $s = k^{-1} \cdot (s_1 + sk_1^{-1} \cdot s_2) \bmod n$, where $s_1 = k_2 \cdot (sk_3 \cdot e + r \cdot sk_2) \bmod n$ and $s_2 = k_2 \cdot sk_2 \cdot sk_3 \cdot r \bmod n$. Substituting s_1 and s_2 into s, we derive $s \equiv k_1^{-1} \cdot [k_2 \cdot sk_2 \cdot (sk_3 \cdot e + r \cdot sk_2) + sk_1^{-1} \cdot k_2 \cdot sk_2 \cdot sk_3 \cdot r] \bmod n$. Simplifying using **Equation (1)**, this reduces to $s \equiv k_1^{-1} \cdot k_2 \cdot sk \cdot (e + sk \cdot r) \bmod n$, which matches the standard ECDSA structure [12]. Verification reconstructs $R' = u_1 \cdot G + u_2 \cdot pk$ where $u_1 = e \cdot s^{-1} \bmod n$ and $u_2 = r \cdot s^{-1} \bmod n$. Substituting u_1, u_2, pk, we obtain $R' = (e + sk \cdot r)\, s^{-1} \cdot G = k_1 \cdot G$, ensuring $R'.x \bmod n = r$.

5.2. Security Proof

Theorem 1 (Existential Unforgeability): The proposed scheme P-CSNKS is existentially unforgeable under adaptive chosen-message attacks in the random oracle model, assuming the hardness of ECDLP.

Proof. Formal Definitions:

EUF-CMA Security: A signature scheme is EUF-CMA secure if no PPT adversary $\mathcal{A}$, given access to a signing oracle and hash oracle, can produce a valid signature on a new message $M^* \notin Q_{sign}$ with non-negligible probability ϵ, where Q_{sign} is the set of signing queries.

ECDLP Hardness: For G of order n, given $(G, Q = k \cdot G)$, finding $k \in [1, n-1]$ is intractable for all PPT algorithms.

Simulator Construction: S receives an ECDLP instance $(G, Q = k \cdot G)$ and aims to compute k. S interacts with $\mathcal{A}$:

Key Generation: S generates the master private key sk via the non-linear splitting mechanism in **Equation (1)**. It sets the public key $pk = sk \cdot G$ and splits sk into subkeys sk_1, sk_2, sk_3 as per Section 4.1. S provides $\mathcal{A}$ with pk^1.

Hash Queries: S maintains a list L_H of tuples (M_i, r, h) to simulate the random oracle H. For each query M_i, if $(M_i, \cdot$

$h) \in L_H$, S returns h. Otherwise, randomly samples $r \leftarrow \mathbb{Z}_n$ and adds (M_i, r, h) to L_H then returns h.

Sign Queries: For a message M, S simulates the collaborative signing protocol as follows. S computes $e = H(M)$ and generates nonce k_1, k_2 as in Section 4.2., then derives r from $Q = k_1 \cdot G$ and computes s using **Equation (1)** and the simulated subkey. S generates σ_1 , σ_2 via Algorithm 1, leveraging precomputed hash chains. Finally, S returns $\sigma = (r, s, \sigma_1, \sigma_2)$.

Probability Analysis: After $\mathcal{A}$ outputs a forgery $\sigma^* = (r^*, s^*, \sigma_1^*, \sigma_2^*)$ on $M \notin \{M_i\}$, S extracts $e^* = H(M)$ and checks consistency with L_H. S replies $\mathcal{A}$ with a different hash $e' \neq e^*$ to obtain another forgery $\sigma' = (r^*, s^*, \sigma_1', \sigma_2')$. S solves for sk algebraically, which reduces to solving the ECDLP instance. If $\mathcal{A}$ succeeds with probability ϵ, S solves ECDLP with probability ϵ^2/q_H, where q_H is the number of hash queries. Since ECDLP is hard, ϵ must be negligible.

Theorem 2 (Key Secrecy): Under the non-linear private key splitting mechanism, the master private key remains computationally hidden from any PPT adversary that corrupts one of the two parties (client or server) and obtains his subkey(s).

Proof. Without loss of generality, assume $\mathcal{A}$ corrupts the client, obtaining subkeys sk_1. To recover sk, $\mathcal{A}$ must solve **Equation (1)**: $sk \equiv (sk_2 \cdot sk_1^{-1} + sk_2 \cdot sk_3^{-1}) \bmod n$, where sk_2 and sk_3 remain unknown to $\mathcal{A}$. The **Equation (1)** introduces a non-linear dependency among the subkeys due to the multiplicative inverses sk_1^{-1} and sk_3^{-1}. Specifically, $\mathcal{A}$ faces a system of two equations with three variables (sk_2, sk_3, sk):

$$sk_2 \cdot sk_1^{-1} \equiv a \bmod n,$$
$$sk_2 \cdot sk_3^{-1} \equiv b \bmod n,$$

where $a + b \equiv sk \bmod n$. However, $\mathcal{A}$ only possesses sk_1, leaving sk_2 and sk_3 as independent unknowns. Even if $\mathcal{A}$ attempts brute-force attacks or algebraic manipulation, the non-linear structure prevents direct isolation of sk. For instance, solving for sk_2 requires knowledge of sk_3, and vice versa, creating a circular dependency. Additionally, the secure deletion of sk ensures no residual traces of the master key exist post-splitting. Even with partial subkey compromise, the algebraic interdependencies and information-theoretic gaps introduced by the non-linear splitting mechanism guarantee that sk remains hidden. Thus, the master private key sk remains computationally hidden from any PPT adversary.

Theorem 3 (Quantum Resistance): The hash-based post-quantum signature components in P-CSNKS are existentially unforgeable against quantum adversaries in the quantum random oracle model, provided the hash function H is post-quantum collision-resistant and one-way.

Proof. Let $\mathcal{A}$ be a quantum adversary with access to a quantum random oracle H. The hash-based signature components output by the Algorithm 1 rely on iterated hash chains $H^{aj}(sk_{i,j})$. To forge a signature, $\mathcal{A}$ must either find a preimage of $H^{255-aj}(\sigma_{i,j})$ to recover pk_i, or generate a valid hash chain σ_i without knowing sk_1. In quantum random oracle, quantum queries to H provide no advantage in finding collisions or preimages due to the lower-bound security of hash functions against Grover's algorithm. By the one-way function and collision resistance of H, the probability of $\mathcal{A}$ forging σ_1 or σ_2 is negligible. The hybrid design ensures that breaking quantum resistance requires simultaneously solving ECDLP and inverting the hash chains, which is computationally infeasible. Hence, P-CSNKS achieves quantum resistance under the quantum random oracle model.

6. Performance Analysis

To evaluate the practical efficiency of the proposed P-CSNKS scheme, we conduct a comprehensive performance analysis, comparing it with two state-of-the-art lattice-based schemes [6,7] and a hash-based signature scheme [8]. We focus on computational complexity, algorithm execution time, and resource utilization, with a particular emphasis on the claimed improvements in computational overhead.

The performance evaluation is conducted in a controlled environment, where both the client and server are implemented on standard computing platforms with comparable hardware specifications. The experimental setup ensures consistent conditions for all schemes, allowing for fair comparisons. The experiments were conducted on a standardized hardware platform equipped with an Intel(R) Core(TM) i5-10505 CPU @3.20 GHz processor and 32 GB of RAM. The software stack utilized the Java Runtime Environment (JRE) version 10.0.1, integrated with the Bouncy Castle cryptography library (v1.70) to en-

able efficient elliptic curve operations (e.g., ECDSA) and SM3 hash algorithm implementations. All schemes were implemented using the same elliptic curve parameters and security level. The results in **Figure 4** demonstrate that P-CSNKS achieves a significant reduction in time cost during the key generation phase, collaborative signing phase, and signature verification phase. This improvement is attributed to the lightweight nature of the non-linear private key splitting mechanism and the efficient integration of hash-based post-quantum components.

Table 1 summarizes the comparison of computational complexity and algorithm execution time (in milliseconds) for key generation, collaborative signing, and signature verification across the three schemes, where n denotes the bit length of the elliptic curve order or lattice dimension in the comparative schemes. The results demonstrate that P-CSNKS not only exhibits the lowest computational overhead but also demonstrates efficiency in terms of memory and bandwidth usage, making it highly suitable for resource-constrained environments.

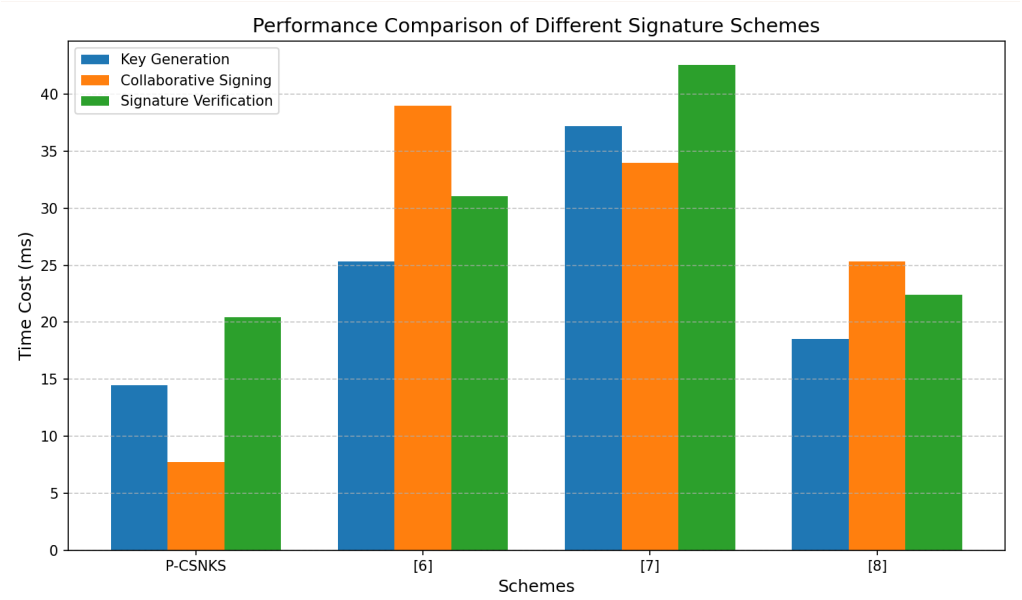

Figure 4. Time cost comparison of each scheme.

Table 1. Comparison of computational complexity and algorithm execution time.

Scheme	Client Computational Complexity	Server Computational Complexity	Key Generation	Collaborative Signing	Signature Verification	Memory Usage (MB)	Bandwidth Usage (KB/sig)
P-CSNKS	$\mathcal{O}(n)$	$\mathcal{O}(n)$	14.46	7.74	20.41	15.2	1.2
[6]	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2 \log n)$	25.33	39.02	31.05	23.7	2.1
[7]	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	37.23	33.97	42.56	32.4	2.8
[8]	$\mathcal{O}(n)$	$\mathcal{O}(n)$	18.50	25.31	22.44	19.8	1.8

7. Conclusions

This paper proposes P-CSNKS, a novel post-quantum collaborative signature scheme that utilizes a non-linear private key splitting technique. The scheme presents several remarkable advantages. Firstly, the non-linear private key splitting technique effectively prevents partial key reconstruction, significantly enhancing the resilience against collusion and side-channel attacks compared to traditional linear secret sharing. It ensures that even if attackers intercept a certain number of shares, they cannot directly reveal the secret, thereby significantly improving the security of private key management. Secondly, P-CSNKS successfully integrates hash-based post-quantum signature components into the collaborative signing process while maintaining compatibility with dynamic verification. This hybrid

design not only provides robust resistance to quantum attacks but also achieves provable security against quantum adversaries without requiring extensive cost. It effectively bridges the gap between classical cryptographic schemes and post-quantum cryptography, ensuring the security of digital signatures in the era of quantum computing. Thirdly, the scheme demonstrates practical efficiency. Experimental evaluations demonstrate that P-CSNKS incurs a lightweight computational overhead compared to lattice-based alternatives and the SM3-OTS-based Collaborative Signature Scheme. Specifically, P-CSNKS outperforms the lattice-based scheme in Chen et al. [7] by more than 40% and the scheme in Li et al. [6] by 30% in computational overhead. When compared to the hash-based signature scheme in Liu et al. [8], P-CSNKS also exhibits a slight computational efficiency advantage across all phases, making it highly suitable for latency-sensitive applications in electronic and information systems, where efficient cryptographic operations are crucial.

Compatibility with existing cryptographic frameworks is a key consideration. P-CSNKS has been designed to maintain compatibility with existing infrastructure by preserving the standard ECDSA structure and enabling seamless integration with legacy systems. This enables a smooth transition to post-quantum security without requiring a complete overhaul of current cryptographic implementations. However, there are still some challenges that need to be addressed. For instance, the adoption of new cryptographic schemes often requires updates to cryptographic libraries and APIs, which may involve significant development and testing efforts. Additionally, the integration of P-CSNKS with existing protocols may require careful consideration of key management and certificate infrastructure.

Regarding the feasibility of implementation within real-world security architectures, P-CSNKS shows great potential. In blockchain networks, the collaborative nature of the scheme can enhance transaction signing security by distributing signing authority among multiple participants, thereby reducing the risk of single points of failure. Its compatibility with existing verification mechanisms ensures that it can be integrated into blockchain systems with minimal modifications. For IoT networks, P-CSNKS's lightweight computational overhead and resistance to quantum attacks make it suitable for resource-constrained devices. The non-linear private key splitting technique also helps mitigate the security risks associated with device compromises in IoT environments. Nevertheless, the practical implementation of these architectures may require addressing additional considerations, such as network latency, scalability, and interoperability with other security protocols.

In future work, we plan to extend P-CSNKS from two-party to multi-party scenarios[13-16] by adapting the non-linear splitting mechanism to a hierarchical structure. Specifically, the master key will be split into m subkeys using modular multiplicative inverses, satisfying $sk \equiv \sum_{i=1}^{m} \left(sk_i \cdot \prod_{j \neq i} sk_j^{-1} \right) \bmod n$. This ensures that reconstructing sk requires colluding at least t+1 parties (with t as the threshold), while maintaining EUF-CMA security by extending the security proof to multi-party collusion. Efficiency will be optimized via batch hash chain computations and parallel signing protocols. For practical deployment, we can develop adapters for integrating P-CSNKS with existing blockchain frameworks (e.g., Ethereum 2.0) and IoT protocols (MQTT), preserving ECDSA verification interfaces, and propose a distributed key generation (DKG) protocol combining non-linear splitting with threshold cryptography to eliminate single points of failure in key distribution, validating performance on resource-constrained devices (e.g., ARM Cortex-M4) and evaluate latency in edge computing networks.

Challenges and Solutions. Security challenges in multi-party settings: Collusion risks increase with more parties. We propose a layered security model: 1)Dynamic threshold adjustment: Adapt the collusion threshold based on system trust levels using secure multi-party computation (MPC); 2)Post-quantum key insulation: Periodically refresh hash chains using quantum-resistant pseudorandom functions (QR-PRFs) to mitigate key exposure over time.

Scalability. Signature size and computation overhead may increase with the number of parties. Solutions include: 1)Aggregated hashing: Compress multi-party signatures via Merkle tree aggregation, reducing verification complexity to $\mathcal{O}$ (log$m$); 2)Hardware acceleration: Implement hash chain iterations on FPGAs to speed up post-quantum components by 2–3x.

In future evaluations, we will expand testing scenarios to include more diverse real-world applications. We will specifically assess the scheme's performance and security in complex environments such as large-scale IoT networks and cross-organizational blockchain systems. Furthermore, we will briefly discuss the scheme's resilience against side-channel attacks, such as timing analysis, by evaluating its performance under different timing conditions and analyzing the effectiveness of the non-linear private key splitting technique in mitigating such risks. These expanded tests will provide a more comprehensive understanding of P-CSNKS's practical effectiveness, robustness, and security against various attack vectors.

Authors Contribution

Conceptualization, F.L. and Y.L.; methodology, F.L.; formal analysis, F.L.; investigation, Y.L.; data curation, F.L.; writing—original draft preparation, F.L.; writing—review and editing, Y.L.; supervision, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Data may be provided if requested.

Conflict of Interest

The authors declared that there is no conflict of interest in this article.

References

[1] Shor, P.W., 2002. Introduction to quantum algorithms. Proceedings of Symposia in Applied Mathematics. 58, 143–160.

[2] Lindell, Y., 2017. Fast secure two-party ECDSA signing. Proceedings of The 37th Annual International Cryptology Conference–CRYPTO 2017; August 20–24, 2017; Santa Barbara, CA, USA. pp. 613–644.

[3] Tu, B., Chen, Y., Cui, H., et al., 2024. Fast two-party signature for upgrading ECDSA to two-party scenario easily. Theoretical Computer Science. 986, 114325. DOI: https://doi.org/10.1016/j.tcs.2023.114325

[4] Xiao, Y., Zhang, L., Yang, Y., et al., 2024. Provably secure multi-signature scheme based on the standard SM2 signature scheme. Computer Standards & Interfaces. 89, 103819. DOI: https://doi.org/10.1016/j.csi.2023.103819

[5] Beimel, A., 2011. Secret-sharing schemes: a survey. Proceedings of The International Conference on Coding and Cryptology; May 30–June 3 2011; Berlin, Germany. pp. 11–46. DOI: https://doi.org/10.1007/978-3-642-20901-7_2

[6] Li, Q., Luo, M., Hsu, C., et al., 2022. A quantum secure and noninteractive identity-based aggregate signature protocol from lattices. IEEE Systems Journal. 16(3), 4816–4826. DOI: https://doi.org/10.1109/JSYST.2021.3112555

[7] Chen, X., Huang, J., Xiao, K., et al., 2025. A non-interactive identity-based multi-signature scheme on lattices with public key aggregation. IEEE Transactions on Dependable and Secure Computing. (99), 1–11. DOI: https://doi.org/10.1109/TDSC.2025.3543425

[8] Liu, S., Zhou, X., Wang, X.A., et al., 2025. A hash-based post-quantum ring signature scheme for the Internet of Vehicles. Journal of Systems Architecture. 160, 103345. DOI: https://doi.org/10.1016/j.sysarc.2025.103345

[9] Suhail, S., Hussain, R., Khan, A., et al., 2020. On the role of hash-based signatures in quantum-safe internet of things: current solutions and future directions. IEEE Internet of Things Journal. 8(1), 1–17. DOI: https://doi.org/10.1109/JIOT.2020.3013019

[10] Doerner, J., Kondi, Y., Lee, E., et al., 2018. Secure two-party threshold ECDSA from ECDSA assumptions. Proceedings of The IEEE Symposium on Security and Privacy (SP); May 20–24, 2018; San Francisco, CA, USA. pp. 980–997.

[11] Yang, Y., Yin, F., Chen, L., et al., 2025. A compact post quantum one time signature scheme over SM3 algorithm. Journal of Software. 36(10), 1–13.

[12] Johnson, D., Menezes, A., Vanstone, S., 2001. The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security. 1, 36–63. DOI: https://doi.org/10.1007/s102070100002

[13] Jiang, S., Alhadidi, D., Khojir, H.F., 2025. Key-and-signature compact multi-signatures for blockchain:

A compiler with realizations. IEEE Transactions on Dependable and Secure Computing. 22(1), 579–596. DOI: https://doi.org/10.1109/TDSC.2024.3410695

[14] Wang, Y., Li, B., Wu, J., et al., 2025. An efficient multi-party signature for securing blockchain wallet. Peer-to-Peer Networking and Applications. 18(3), 1–20. DOI: https://doi.org/10.1007/s12083-025-01958-1

[15] Wang, Y., Xu, G.B., Jiang, D.H., 2025. A quantum image secret sharing scheme based on designated multi-verifier signature. Advanced Quantum Technologies. 8(1), 2400267. DOI: https://doi.org/10.1002/qute.202400267

[16] Xu, R., Zhou, Y., Yang, Q., et al., 2024. An efficient and secure certificateless aggregate signature scheme. Journal of Systems Architecture. 147, 103030. DOI: https://doi.org/10.1016/j.sysarc.2023.103030