Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER
PUBLISHER

Research Article

An Efficient Post-Quantum Secure Two-Tier Signature Scheme Based on Chameleon Hashing

Yong Wang, Eddie Shahril Ismail*

Department of Mathematical Sciences, Faculty of Science and Technology, Universiti Kebangsaan Malaysia, 43600, UKM Bangi, Selangor, Malaysia
E-mail: esbi@ukm.edu.my

Received: 6 June 2025; **Revised:** 11 June 2025; **Accepted:** 26 June 2025

Abstract: In light of the growing threat from quantum adversaries to traditional digital signatures, the paper investigates a post quantum secure signature scheme integrates chameleon hash functions within a two-tier signing framework. The proposed construction achieves essential security features such as non-repudiation and recipient-specific verification, while maintaining a non-interactive structure compatible with efficient hash-and-sign paradigms. A novel enhancement of the Kabatianskii-Krouk-Smeets (KKS) code-based scheme is incorporated, leveraging rank metric techniques over large finite fields to bolster resistance against both classical and quantum attacks. The layered architecture, separating long-term and ephemeral keys, supports multi-message authentication and improved key lifecycle management. Comprehensive security definitions and reductions are provided to demonstrate the scheme's unforgeability under realistic assumptions, establishing its viability for robust post-quantum applications.

***Keywords*:** code-based signatures, chameleon hashing, two-tier cryptographic frameworks, rank metric security

MSC: 94A60

1. Introduction

Rivest-Shamir-Adleman (RSA), as an influential public-key encryption schemes, initiated a major paradigm shift in modern cryptographic theory [1]. The concept of "non-repudiation" or "invisibility" of signatures [2] mentioned in this cryptosystem tells us that the recipient can only verify the signature with the cooperation of the signer. However, once the signature is verified, the signer cannot deny its authenticity.

The design of digital signature schemes that allow controlled flexibility in message authentication has become a central research focus in modern cryptography. Among the notable contributions addressing this challenge are chameleon signatures, which provide a structured means to enable selective verifiability under recipient-specific constraints [3–6]. It allows the recipient to modify a given signature σ for a different message m' using a specific trapdoor. One advantage of this approach is its increased flexibility. The key feature of Chameleon signatures are non-transferability, meaning that a signature created for one recipient cannot be verified by others, thereby ensuring strong non-repudiation. These unique properties provide a higher level of security, particularly in situations where authenticity and non-repudiation are crucial.

DOI: https://doi.org/10.37256/cm.6420257389

Chameleon Signatures are unique in that they are non-transferable, meaning a signature from one recipient can't be verified by someone else, which ensures strong non-repudiation. This feature boosts security, especially when it comes to confirming authenticity and preventing denial of signature. At the core of Chameleon Signatures lies the Chameleon hash function, which reinforces security by enabling controlled collisions in a trapdoor-based setting. It enables effective collision computation when using a trapdoor, while still resisting collisions even without it. This capability is essential for non-interactive signature protocols, where the recipient can verify the signature's validity independently, without needing to communicate directly with the signer. Moreover, by adding an extra security layer on top of traditional hash functions, the Chameleon hash offers a two-layer security approach, making it a powerful tool in modern encryption systems.

The latest advancements have further driven research momentum and development. There are some schemes such as lattice-based signatures and hash-based signatures, that have become promising candidates for post quantum cryptography. These schemes also have limitations regarding efficiency, storage requirements, and flexibility. In addition to extensively discussed lattice-based and hash-based signature schemes, recent developments in code-based post-quantum signatures provide alternative designs with distinct performance and security characteristics. For instance, Wave [7] utilizes rank metric codes with generalized error decoding to produce short signatures and strong unforgeability guarantees, although it requires complex parameter tuning and has relatively large keys. Meanwhile, the scheme [8, 9] proposes a lightweight structure suitable for embedded environments, balancing signature size and computation cost using low-error syndrome decoding. These code-based schemes broaden the spectrum of Post-Quantum Cryptography (PQC) designs and further motivate the proposed integration of Chameleon hashing with code-based primitives. As is known, lattice-based schemes offer strong quantum security, they have high computational complexity, making them unsuitable for many practical applications. Hash-based schemes may result in large signature sizes and high memory consumption, posing challenges in resource-constrained environments. As a result, many schemes are limited in applicability to scenarios requiring multiple signature instances or reusing public keys across different messages. Recent efforts have also focused on integrating cryptographic techniques with image privacy and embedded hardware design. For example, nonlinear dynamics and transform-domain compression have been applied to privacy-preserving image processing [10, 11], while hardware/software co-design techniques have significantly improved the performance and configurability of post-quantum schemes such as CRYSTALS-Dilithium [12].

The two-tier signature architecture improves key usage by separating it into long-term and temporary components. Traditional encryption schemes rely on a static master key, with auxiliary keys used for individual messages. Early models [13] employed one-time auxiliary keys to enhance security. In previous work [14], we proposed a new approach that combined Hidden Field Equations (HFE) and Internal Perturbation Hidden Field Equations (IPHFE) with code-based and multivariate encryption to address limitations in both. This paper advances that work by enabling auxiliary keys to be used across multiple messages, improving flexibility and scalability. Additionally, integrating Chameleon hash functions and lattice-based tools significantly enhances security, making the system well-suited for secure code-based encryption applications.

We propose an innovative post-quantum signature scheme that leverages Chameleon hash functions and a dual-signature framework to balance quantum resistance, efficiency, and flexibility. Our design addresses key challenges in post-quantum signatures, such as high computational complexity and large storage requirements. We also demonstrate how this dual-layer architecture can be extended to multi-message security and provide formal security proofs against key leakage and forgery attacks.

2. Mathematical background of cryptography

Let λ be the security parameter, and assume that all algorithms used in this scheme are probabilistic in nature. When we write $x \leftarrow \mathscr{A}(x_1, \ldots, x_n)$, we mean that algorithm $\mathscr{A}$ is executed on inputs $x_1, \ldots, x_n$ using fresh internal randomness, and the result of the computation is assigned to the variable x.

2.1 *Hard problems*

The foundational work by Berlekamp in 1978 [15] demonstrated that decoding linear error-correcting codes and identifying low-weight codewords are both Nondeterministic Polynomial time (NP)-complete tasks. These problems have since formed the cornerstone of various code-based cryptographic constructions. One of the primary formulations is the Computational Syndrome Decoding (CSD) problem, outlined below.

Problem 2.1 (CSD-Problem). For a matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $y \in \mathbb{F}_q^{n-k}$, and an integer w, the goal is to find a vector $x \in \mathbb{F}_q^n$ such that $Hx^T = y^T$ and $w_H(x) = w$.

Definition 2.1 (Syndrome Decoding Distribution). For parameters n, k, and w, the distribution $SD(n, k, w)$ samples $H \leftarrow \mathbb{F}_{q^m}^{(n-k)\times n}$ and a vector $x \leftarrow \mathbb{F}_{q^m}^n$ of Hamming weight w, then outputs the pair (H, Hx^T).

Problem 2.2 (DSD-Problem). Given $(H, y^T) \in \mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^{n-k}$, distinguish whether it was sampled from the $SD(n, k, w)$ distribution or from the uniform distribution over $\mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^{n-k}$.

Replacing the Hamming metric with the rank metric in the above definitions yields the rank-based variants of these problems.

Problem 2.3 (RSD-Problem). Let $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $y \in \mathbb{F}_{q^m}^{n-k}$. Given a positive integer w, find a vector $x \in \mathbb{F}_{q^m}^n$ satisfying $Hx^T = y^T$ and rank weight of x is exactly w.

Definition 2.2 (Rank Syndrome Decoding Distribution). In the $RSD(n, k, w)$ distribution, a matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is sampled along with a vector $x \in \mathbb{F}_{q^m}^n$ of rank weight w, and the pair (H, Hx^T) is returned.

Problem 2.4 (DRSD-Problem). Given $(H, y) \in \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{n-k}$, determine whether the instance originates from the $RSD(n, k, w)$ distribution or is drawn uniformly at random from the product space.

To model broader parameter ranges, we consider the extended decoding distributions, which generalize the fixed-weight constraint.

Definition 2.3 (extSD-Distribution). Let $a \leq b$ be two integers. The distribution $extSD(n, k, a, b)$ samples $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and a vector $x \in \mathbb{F}_{q^m}^n$ satisfying $a \leq \|x\| \leq b$ (under the Hamming metric), and outputs (H, Hx^T).

Problem 2.5 (extDSD-Problem). Given an $(H, y^T) \in \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{n-k}$, decide whether it originates from $extSD(n, k, a, b)$ or a uniform distribution.

Definition 2.4 (Extended Rank Syndrome Decoding Distribution). For the distribution $extRSD(n, k, a, b)$ samples $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and a vector $x \in \mathbb{F}_{q^m}^n$ such that $a \leq \|x\| \leq b$ under the rank metric, then outputs the pair (H, Hx^T).

Problem 2.6 (extDRSD- Problem). Given $(H, y^T) \in \mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^{n-k}$, distinguish whether the instance is from $extRSD(n, k, a, b)$ or sampled uniformly.

We treat the $extDSD(n, k, a, b)$ and $extDRSD(n, k, a, b)$ problems as computationally intractable, specifically assuming that they are NP-hard and, under certain conditions, NP-complete [16].

Definition 2.5 (SIS-Problem). Let $q \in \mathbb{Z}$ be a modulus, and let $A \in \mathbb{Z}_q^{n\times m}$ be a matrix drawn uniformly at random. Given a bound β on the norm, the SIS problem is to determine a non-zero vector $x \in \mathbb{Z}^m$ such that:

$$Ax \equiv 0 \pmod{q} \quad \text{and} \quad \|x\| \leq \beta,$$

where $\|x\|$ may be defined under different norms, such as:

- Euclidean norm: $\|x\|_2 = \sqrt{\sum_{i=1}^m x_i^2}$,
- Infinity norm: $\|x\|_\infty = \max_i |x_i|$.

The SIS problem is widely regarded as computationally hard and is fundamental to the design of secure lattice-based cryptographic constructions.

2.2 Trapdoor-based chameleon hashing

Next, we will introduces the concept of Chameleon Hash Functions (CHFs), outlining their formal definition and highlighting their essential cryptographic properties.

Definition 2.6 A Chameleon Hash Function is defined as a tuple

$$CHF = (\mathsf{CHamGen}();\ \mathsf{CHamHash}();\ \mathsf{Coll}()),$$

where:

1. $\mathsf{CHamGen}(1^{\lambda})$, with security parameter λ, outputs a public hash key *chhk* and a corresponding trapdoor *chtd*.
2. $\mathsf{CHamHash}(chhk;\ x;\ s)$ computes the hash output h for a message x using randomness s.
3. $\mathsf{Coll}(chtd;\ (x,\ s);\ \hat{x})$ returns a value $\hat{s}$ such that:

$$\mathsf{CHamHash}(chhk;\ x;\ s) = \mathsf{CHamHash}(chhk;\ \hat{x};\ \hat{s}).$$

Definition 2.7 A Chameleon Hash Function CHF is said to be $(t,\ \varepsilon)$-collision resistant if no adversary $\mathscr{F}$, running in time at most t, can find two distinct pairs $(x_1,\ s_1) \neq (x_2,\ s_2)$ such that:

$$\Pr_{\substack{(chhk,\ chtd) \leftarrow \mathsf{CHamGen}(1^{\lambda}) \\ (x_1,\ s_1),\ (x_2,\ s_2) \leftarrow \mathscr{F}(chhk)}} [\mathsf{CHamHash}(chhk;\ x_1;\ s_1) = \mathsf{CHamHash}(chhk;\ x_2;\ s_2)] \leq \varepsilon.$$

The trapdoor *chtd* must satisfy the following security guarantees:

1. **Collision Resistance:** It is computationally infeasible to efficiently generate two message-randomness pairs $(x_1,\ s_1)$ and $(x_2,\ s_2)$ with $x_1 \neq x_2$ that hash to the same output:

$$\mathsf{CHamHash}(chhk;\ x_1;\ s_1) = \mathsf{CHamHash}(chhk;\ x_2;\ s_2).$$

2. **Trapdoor Collisions:** Knowing the trapdoor *chtd* enables efficient computation of a collision. Specifically, given $(x_1,\ s_1)$ and any message x_2, one can compute s_2 such that:

$$\mathsf{CHamHash}(chhk;\ x_1;\ s_1) = \mathsf{CHamHash}(chhk;\ x_2;\ s_2).$$

3. **Uniformity:** The distribution of $\mathsf{CHamHash}(chhk,\ x,\ s)$ should appear uniform for any fixed input x, even when s is selected at random. A relaxed requirement allows the hash outputs for different messages to be computationally indistinguishable. That is, for $x_1 \neq x_2$, we require:

$$\{\mathsf{CHamHash}(chhk;\ x_1;\ s_1) \mid s_1 \leftarrow D_1\} \equiv \{\mathsf{CHamHash}(chhk;\ x_2;\ s_2) \mid s_2 \leftarrow D_2\},$$

where D_1 and D_2 are distributions over the randomness, and are computationally indistinguishable.

2.3 *Design and implementation of two-tier signature mechanisms*

The concept of a d-bounded two-tier signature scheme, a construction that enables secure signing over multiple messages through a hierarchical key architecture. The scheme consists of four randomized algorithms, described as follows.

Definition 2.8 [13] A d-time two-tier signature scheme $TTSig$ is defined by a tuple of probabilistic algorithms:

$$TTSig = (\mathsf{PriGen},\ \mathsf{SecGen},\ \mathsf{TTSign},\ \mathsf{TTVerify})$$

with the following functionality:

• $\mathsf{PriGen}(1^{\lambda},\ d)$: Takes the security parameter λ and a positive integer d, and returns a primary key pair $(\mathsf{psk},\ \mathsf{ppk})$ for signing and verification.

• $\mathsf{SecGen}(\mathsf{ppk},\ \mathsf{psk})$: Generates a fresh secondary key pair $(\mathsf{spk},\ \mathsf{ssk})$ associated with a specific signing session.

• $\mathsf{TTSign}(\mathsf{psk},\ \mathsf{ssk},\ x)$: Produces a signature σ on message x using the primary and secondary signing keys. A stateful version $\mathsf{TTSign}(\mathsf{psk},\ \mathsf{ssk},\ x,\ j)$ incorporates a counter or session index j.

• $\mathsf{TTVerify}(\mathsf{ppk},\ \mathsf{spk},\ x,\ \sigma)$: Deterministically returns 1 if the signature is valid, 0 otherwise. The stateful verifier $\mathsf{TTVerify}(\mathsf{ppk},\ \mathsf{spk},\ x,\ \sigma,\ j)$ uses the state index j for session tracking.

This construction generalizes the one-time signature scheme when $d = 1$ and allows for multi-message signing under key isolation.

Definition 2.9 We say that a two-tier signature scheme $TTSig$ is $(t,\ q,\ d,\ \varepsilon)$-existentially unforgeable under a non-adaptive chosen-message attack (TT-EUF-NCMA) if the success probability of any PPT adversary $\mathscr{F}$ in the corresponding experiment is at most ε. Formally:

$$\Pr\left[\mathsf{Exp}^{TT\text{-}EUF\text{-}NCMA}_{TTSig,\ \mathscr{F},\ q}(\lambda,\ d) = 1\right] \le \varepsilon$$

for all adversaries $\mathscr{F}$ running in time at most t and allowed to issue at most q queries.

2.3.1 *Security experiments for two-tier signature schemes*

We describe the unforgeability experiments for two-tier signature schemes under non-adaptive and adaptive chosen-message attacks, denoted as $\mathrm{Exp}^{\mathrm{TT\text{-}EUF\text{-}NCMA}}_{\mathscr{F},\ q}(\lambda,\ d)$ and $\mathrm{Exp}^{\mathrm{TT\text{-}EUF\text{-}CMA}}(\lambda,\ d)$, respectively.

Non-Adaptive experiment $\mathbf{Exp}^{\mathbf{TT\text{-}EUF\text{-}NCMA}}_{\mathscr{F},\ q}(\lambda,\ d)$

1. The challenger generates the primary key pair by running $(\mathrm{ppk},\ \mathrm{psk}) \leftarrow \mathsf{PriGen}(1^{\lambda},\ d)$.

2. The adversary $\mathscr{F}$ issues a single query to the oracle $\mathsf{NTTSign}(m_1,\ \dots,\ m_d)$, which performs the following steps:

- Increments the session index i;
- Computes $(\mathrm{spk}_i,\ \mathrm{ssk}_i) \leftarrow \mathsf{SecGen}(\mathrm{ppk},\ \mathrm{psk})$;
- For each $j = 1,\ \dots,\ d$, computes $\sigma_j \leftarrow \mathsf{TTSign}(\mathrm{psk},\ \mathrm{ssk}_i,\ m_j)$;
- Stores the queried message set $\{m_1,\ \dots,\ m_d\}$ as $\mathscr{Q}_i$;
- Returns the tuple $(\mathrm{spk}_i,\ \sigma_1,\ \dots,\ \sigma_d)$.

3. The adversary outputs a tuple $(m^*,\ \sigma^*,\ i^*)$.

4. The experiment returns 1 if the signature verifies successfully, i.e.,

$$\mathsf{TTVerify}(\mathrm{ppk},\ \mathrm{spk}_{i^*},\ m^*,\ \sigma^*) = 1$$

and the message m^* was not among those in $\mathscr{Q}_{i^*}$. Otherwise, the experiment returns 0.

Adaptive experiment $\mathbf{Exp}^{\textbf{TT-EUF-CMA}}(\lambda,\ d)$

1. The challenger initializes the system with $(\mathrm{ppk},\ \mathrm{psk}) \leftarrow \mathsf{PriGen}(1^{\lambda},\ d)$.

2. The adversary is given access to two oracles:

• $\mathsf{OSKey}()$: On invocation, increments index i, sets $j_i = 0$, generates $(\mathrm{spk}_i,\ \mathrm{ssk}_i) \leftarrow \mathsf{SecGen}(\mathrm{ppk},\ \mathrm{psk})$, and returns spk_i.

• $\mathsf{TTSign}(i',\ m)$: Increments $j_{i'}$ and stores the message m as $m_{j_{i'}}$. If $j_{i'} > d$ or $(\mathrm{spk}_{i'},\ \mathrm{ssk}_{i'})$ is undefined, returns $\perp$. Otherwise, computes $\sigma \leftarrow \mathsf{TTSign}(\mathrm{psk},\ \mathrm{ssk}_{i'},\ m_{j_{i'}})$, stores $m_{j_{i'}}$ in $\mathscr{Q}_{i'}$, and returns σ.

3. The adversary eventually outputs a candidate forgery $(m^*,\ \sigma^*,\ i^*)$.

4. The experiment outputs 1 if:

$$\mathsf{TTVerify}(\mathrm{ppk},\ \mathrm{spk}_{i^*},\ m^*,\ \sigma^*) = 1 \quad \text{and} \quad m^* \notin \mathscr{Q}_{i^*}$$

indicating a successful forgery. Otherwise, it returns 0.

The security notion of existential unforgeability under adaptive chosen-message attacks (TT-EUF-CMA) for two-tier signature schemes aligns conceptually with its traditional counterpart in standard signature frameworks. Similarly, strong unforgeability is modeled for both adaptive and non-adaptive settings-namely, TT-SUF-CMA and TT-SUF-NCMA-following corresponding formal structure incorporated into conventional signature scheme definitions.

2.4 *Rank metric and rank codes*

Let $\mathbb{F}_{q^m}$ be an extension field of degree m over $\mathbb{F}_q$, with $m,\ n \in \mathbb{Z}^+$. A basis $B = \{\alpha_1,\ \ldots,\ \alpha_m\}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ allows any $\alpha \in \mathbb{F}_{q^m}$ to be uniquely written as:

$$\alpha = \sum_{i=1}^{m} \lambda_i \alpha_i, \quad \lambda_i \in \mathbb{F}_q.$$

For $x = (x_1,\ \ldots,\ x_n) \in \mathbb{F}_{q^m}^n$, define its matrix representation $M \in \mathbb{F}_q^{n\times m}$ by expanding each x_i in basis B, such that:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = M \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}.$$

This yields a one-to-one correspondence between x and M over $\mathbb{F}_q$.

Definition 2.10 The *rank weight* of $x \in \mathbb{F}_{q^m}^n$ is defined as the rank of its expansion matrix $A \in \mathbb{F}_q^{n\times m}$, may be represented as $\mathrm{rank}(x)$, or simply $||x||$ or $w_R(x)$ when the context is clear.

The *rank distance* between $x,\ y \in \mathbb{F}_{q^m}^n$ is $d(x,\ y) = \mathrm{rank}(x-y)$. For comparison, the *Hamming weight* of x is $w_H(x)$, and the Hamming distance is $d_H(x,\ y) = w_H(x-y)$.

Remark 2.1 The rank of x equals the dimension over $\mathbb{F}_q$ of the span of its coordinates, or equivalently, the rank of its matrix representation. By construction, $\mathrm{rank}(x) \leq w_H(x)$.

Definition 2.11 Let $G \in \mathbb{F}_{q^m}^{k\times n}$ have rank k. A rank metric code $C \subseteq \mathbb{F}_{q^m}^n$ with length n and dimension k is defined as:

$$C = \{xG \mid x \in \mathbb{F}_{q^m}^k\},$$

where G is the generator matrix. Equivalently, using a parity-check matrix H, we have

$$C = \{x \in \mathbb{F}_{q^m}^n \mid Hx^T = 0\}.$$

The *minimum rank distance* of C is:

$$d_C = \min\{\text{rank}(x) \mid x \in C \setminus \{0\}\}.$$

Definition 2.12 Suppose $B = \{\alpha_1, \ldots, \alpha_m\}$ be a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, and S be an invertible matrix of order m over the base field. Define a map $\varphi_B : \mathbb{F}_{q^m}^n \to \mathbb{F}_q^{m\times n}$, which associates each vector $x \in \mathbb{F}_{q^m}^n$ with its associated matrix over $\mathbb{F}_q$ in relation to basis B.

The transformed vector $S \star x$ is then given by:

$$S \star x = \varphi_B^{-1}(S \cdot \varphi_B(x)),$$

where the action is performed by applying S on the row space of the matrix representation, followed by inverse mapping back to $\mathbb{F}_{q^m}^n$.

Remark 2.2 The transformation $S \star x$ serves as a form of rank-preserving permutation, where the coordinates of x are restructured through a linear transformation over the base field. This operation is analogous to general coordinate permutations in Hamming-based settings, but is adapted to the rank metric.

2.5 *Affine transformations*

Affine transformations over extension fields are frequently used in the design of structured cryptographic functions. Below, we present both the standard matrix-based formulation and its univariate extension using polynomial representations.

Definition 2.13 Let $M_T \in GL(n, \mathbb{F}_{q^m})$ be an invertible $n \times n$ matrix and let $v_T \in \mathbb{F}_{q^m}^n$. For a vector $x \in \mathbb{F}_{q^m}^n$, the affine transformation T is defined by:

$$T(x) = M_T \cdot x^T + v_T,$$

where the rank of the transformation is inherited from the matrix M_T, and all operations are carried out over the extension field $\mathbb{F}_{q^m}$.

To express this transformation in univariate polynomial form, consider an irreducible polynomial $g(x) \in \mathbb{F}_{q^m}[x]$ of degree n, and define the extension field $\mathbb{E} = \mathbb{F}_{q^m}[x]/(g(x))$. An isomorphism $\phi : \mathbb{F}_{q^m}^n \to \mathbb{E}$ can be established by:

$$\phi(a_1, \ldots, a_n) = \sum_{i=1}^{n} a_i x^{i-1}.$$

Under this isomorphism, the affine transformation T can be written in its Frobenius form over $\mathbb{E}$:

$$T(X) = \sum_{i=0}^{n-1} B_i X^{q^i} + A,$$

where B_i, $A \in \mathbb{E}$, and $X \in \mathbb{E}$ is the image of x under ϕ. This univariate representation is particularly useful for encoding structure while preserving the algebraic properties of the transformation.

2.6 *Permutation transformations*

Permutations play a role in modifying both the input and output of multivariate polynomial functions. Let $\sigma \in S_n$ act on an input vector $x \in \mathbb{F}_q^n$ by rearranging its coordinates, and let $\tau \in S_m$ apply a similar transformation to the output space of a polynomial mapping $\mathscr{P}(x)$.

The composite function $\tau \circ \mathscr{P} \circ \sigma$ represents a sequence where the input is permuted, passed through $\mathscr{P}$, and then the result is permuted again. This structure provides a mechanism for reshaping polynomial mappings while preserving algebraic properties, and can be useful for obfuscation or symmetry analysis in cryptographic constructions.

2.7 *Gauss transformations*

Gaussian elimination-comprising elementary row and column operations-forms the foundation for matrix reduction techniques over $\mathbb{F}_q$. These operations are invertible and can be represented via elementary matrices, which constitute a subset of affine transformations.

Due to their invertibility, Gaussian transformations preserve algebraic properties relevant to matrix rank and polynomial systems. This makes them an essential component in solving linear equations and analyzing matrix-based cryptographic constructions, such as those found in code-based schemes.

Example 2.1 Gaussian Transformation Consider the matrix $A \in \mathbb{F}_5^{3\times 3}$:

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 3 & 1 \\ 4 & 0 & 5 \end{pmatrix}$$

To reduce A, we perform Gaussian elimination:
1. Use row 1 as the pivot.
2. Update row 3: subtract $4 \times \text{row}_1$, yielding:

$$(4,\ 0,\ 5) - 4 \cdot (1,\ 2,\ 0) = (0,\ -8,\ 5) \equiv (0,\ 2,\ 5) \pmod 5.$$

The resulting matrix is:

$$U = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 3 & 1 \\ 0 & 2 & 5 \end{pmatrix}.$$

This matrix is full-rank and thus suitable for use in the KKS key generation process.

3. The newly proposed signatures schemes

3.1 *An improved KKS scheme*

Otmani and Tillich [17] identified an exponential-time cryptanalytic attack targeting the original KKS signature scheme over $\mathbb{F}_2$. Although the attack is computationally intensive, it reveals the necessity of selecting parameters with greater care. In response, the KKS assumption, which asserts the existence of a parameter regime under which one-time Existential Unforgeability under Chosen Message Attack security can be achieved [18].

The improved KKS scheme employs two rank metric codes. The first is specified via a parity-check matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$, and the second is a linear code $C \subseteq \mathbb{F}_{q^m}^{n}$, characterized by length n and dimension k, and generated using a matrix G. A notable feature of code C is, $\forall$ nonzero $x \in C$, the rank weight $\|x\|$ falls within the interval $(l_1,\ l_2)$ with high probability, where $l_1,\ l_2 \in \mathbb{Z}^+$ and $l_1 < l_2$. The scheme's construction is described in what follows.

3.2 *Key generation (Gen(1^λ))*

Input: A parameter λ indicating desired security level.

Output: A public/private key pair $(ppk,\ psk)$.

1. Construction of the parity-check matrix H:

• Generate a sparse, full-rank matrix $H \in \mathbb{F}_{q^m}^{(N-K)\times N}$ with a sparsity parameter ρ_H, where $0.1 \le \rho_H \le 0.3$.

• Validate H for full rank using LU decomposition. If it fails, regenerate H iteratively until the condition is met.

• Store H using the Compressed Sparse Row (CSR) format to optimize storage and computation.

• Derive a submatrix H_J by randomly selecting a subset $J \subseteq \{1,\ \ldots,\ N\}$ such that $|J| = n > N-K$. The generation of J should be random but reproducible for key consistency.

2. Generator matrix G construction:

• Construct G_1, a sparse matrix with a sparsity parameter ρ_{G_1}.

• Generate a noise matrix E sampled from a Gaussian distribution $\mathscr{N}(0,\ \sigma^2)$, where $\sigma^2 = O(\sqrt{n})$. Ensure the randomness used for E is cryptographically secure.

• Define the generator matrix $G = G_1 + E$, combining structural sparsity and randomness.

3. Computation of the verification matrix Q:

$$Q = H_J \cdot G^T.$$

Utilize the sparsity of H_J and G to achieve computational efficiency with complexity $O(\rho_H \cdot \rho_{G_1} \cdot n^2)$.

4. Generation of the SIS matrix A_{SIS}:

• Construct A_{SIS} to meet the SIS problem requirements, optionally maintaining a sparsity parameter $\rho_{A_{SIS}}$.

5. Key outputs:

• Public key: $ppk = (H,\ Q,\ A_{SIS})$.

• Private key: $psk = (J,\ G)$.

3.3 *Sparse matrix generation algorithm*

To construct the parity-check matrix H in a rank-metric cryptographic context, we employ a randomized sparse matrix generation procedure, designed to ensure full rank while preserving a specified sparsity level. The algorithm iteratively samples matrices until the desired rank condition is satisfied.

Algorithm 1: GenerateSparseMatrix $(n,\ m,\ \rho)$

Input: Number of rows n, number of columns m, sparsity rate ρ

Output: Full-rank sparse matrix $A \in \mathbb{F}_q^{n\times m}$

```
repeat
  Initialize $A \leftarrow 0_{n\times m}$;
  for $i \leftarrow 1$ to $n$ do;
    for $j \leftarrow 1$ to $m$ do;
      Draw $r \sim \mathscr{U}(0,\ 1)$;
      if $r \le \rho$ then;
        $A[i][j] \leftarrow$ RandomElement($\mathbb{F}_q \setminus \{0\}$);
```

```
until rank$(A) = n$;
return $A$
```

Example 3.1 (Sparse Matrix Sampling with Full-Rank Verification)

Consider the parameter setting $n = 4$, $m = 6$, and sparsity rate $\rho = 0.2$. The count of non-zero elements within the matrix generated equals $n \cdot m \cdot \rho = 4 \cdot 6 \cdot 0.2 = 4.8$.

Hence, approximately 5 out of the 24 entries will be non-zero, randomly distributed across the matrix. After each generation, the algorithm verifies whether the matrix A has full rank (i.e., $\text{rank}(A) = n$) via Gaussian elimination. If the rank is insufficient, the matrix is discarded and regenerated.

This process continues until a full-rank matrix is found. Empirically, for moderate values of ρ (e.g., $0.1 \leq \rho \leq 0.4$), the process converges in few steps. The output matrix is suitable for use as a parity-check matrix H in code-based cryptographic schemes such as KKS.

3.4 *Signature generation (Sign(psk, m))*

Input: Message m, private key $psk = (J, G)$.

Output: Signature $\sigma = (J, s_J, x)$.

1. Derive the digest vector $x \in \mathbb{F}_{q^m}^k$ via the application of a cryptographic hash function H to the message m:

$$x \leftarrow H(m),$$

where H can be realized using a secure hash algorithm like SHA-3 or BLAKE2.

2. Generate the codeword u: Calculate u as:

$$u = x \cdot G = x \cdot (G_1 + E) = xG_1 + xE.$$

3. Compute the SIS short vector x_{SIS}:

• Solve for x_{SIS} such that: $A_{SIS} \cdot x_{SIS} = x \pmod q$.

• Ensure $\|x_{SIS}\| \leq l_2$ using lattice basis reduction techniques, such as LLL or BKZ algorithms [19, 20].

4. Construct the signature vector s_J:

$$s_J = u + x_{SIS}.$$

5. Output the signature:

$$\sigma = (J, s_J, x).$$

3.5 *Signature verification (Verify(ppk, (m, σ)))*

Input: Public key $ppk = (H, Q, A_{SIS})$, message m, signature $\sigma = (J, s_J, x)$.

Output: Accept or reject the signature.

1. Recompute the message digest x':

$$x' = H(m).$$

Verify $x' = x$. If not, reject the signature.

2. Recover the full signature vector s:

$$s_i = \begin{cases} s_{J,\ i}, & \text{if } i \in J, \\ \\ 0, & \text{otherwise.} \end{cases}$$

3. Verify the norm constraint:

$$l_1 \leq \|s\| \leq l_2.$$

4. Validate the core equation:

$$H \cdot s^T = Q \cdot x^T.$$

5. Check the SIS condition:

$$A_{SIS} \cdot x_{SIS} = x (\text{mod } q).$$

3.6 *Parameter considerations*

The next part discusses key parameters that shape both the security level and practical efficiency of the proposed KKS signature scheme. Each parameter has a direct impact on how well the scheme can defend against various attacks while ensuring feasible performance.

1. Sparsity Parameters:

The sparsity parameters ρ_H, ρ_{G_1}, and $\rho_{A_{SIS}}$ govern the density of the matrices involved in the scheme. These parameters influence both the computational complexity and the vulnerability to lattice-based attacks, particularly in the context of decoding or cryptanalysis. It is essential to carefully select these parameters to balance efficiency with security.

• For the parity-check matrix H, a sparsity parameter in the range $0.1 \leq \rho_H \leq 0.3$ is typically used based on empirical tests.

• The sparsity parameters for the generator matrix G_1 and the SIS matrix A_{SIS} must also be carefully chosen and tested to ensure their robustness against attacks such as lattice reduction attacks.

The selection of sparsity parameters impacts the security margin and efficiency. Excessively high sparsity may lead to easier attack vectors, while too low sparsity could result in performance bottlenecks. Empirical testing in the context of specific attacks should be performed to fine-tune these values.

2. Security of SIS:

The security of the SIS problem hinges on the careful selection of parameters n, q, and the bound β. To ensure the infeasibility of computing a short, non-trivial vector x satisfying

$$A_{\text{SIS}} \cdot x \equiv b \pmod{q} \quad \text{and} \quad \|x\| \leq \beta,$$

the chosen parameters must fall outside the range susceptible to efficient lattice reduction algorithms.

• The dimension n, modulus q, and bound β should be selected in such a way that solving the SIS instance remains computationally infeasible for potential adversaries, even under the best-known lattice basis reduction algorithms [19, 20].

• Empirical security analysis is required to test the difficulty of the SIS problem for different parameter values. These evaluations help determine whether the chosen parameters yield instances that resist known algorithmic attacks.

The complexity associated with solving the SIS problem is influenced by factors such as the lattice dimension, modulus size, and the size of the shortest solution. In practice, these parameters should be selected to resist known lattice-based attacks such as those based on lattice basis reduction or quantum attacks.

3. Noise matrix E:

The noise matrix E is sampled from a Gaussian distribution $\mathscr{N}(0,\ \sigma^2)$, where $\sigma^2 = O(\sqrt{n})$. This noise is a critical component for ensuring the security of the scheme by introducing sufficient randomness to make it computationally infeasible for an attacker to exploit the structure of the code.

• The value of σ^2 must be chosen carefully to ensure that the noise is large enough to introduce randomness and obscure patterns in the code but not so large as to degrade the efficiency of the system.

• The noise variance σ^2 directly influences both the security and operational efficiency of the scheme. An small variance increases vulnerability to decoding attacks, whereas an excessively large noise may burden the signing and verification routines

3.7 *Security analysis and parameter tradeoffs*

The KKS signature scheme, which is known to be vulnerable to exponential-time attacks such as the one outlined by Otmani and Tillich [17]. To mitigate this, our parameter selection prioritizes larger values for the field extension degree m and code length n, which directly impact the complexity of rank decoding attacks.

In particular, the attack described in [17] scales with the Gaussian elimination cost for low-weight codeword searches in rank metric, requiring:

$$\mathscr{O}\left(q^{(m-r)(n-r)}\right)$$

operations for decoding rank-r errors. By selecting parameters satisfying

$$(m-r)(n-r) \gg \lambda,$$

where λ is the security level (e.g., 128-bit), we ensure that even exponential attacks remain computationally infeasible.

Furthermore, we incorporate sparsity in both the parity-check matrix H and the generator matrix G, setting sparsity rates $\rho_H = 0.2$ and $\rho_G = 0.3$. This design choice reduces the exploitable algebraic structure and storage requirements, while still preserving the necessary rank properties to maintain security.

However, increasing m and n also results in larger public key and signature sizes due to the use of dense representations over $\mathbb{F}_{q^m}$. This introduces a design tradeoff: higher rank weight thresholds improve resistance to structural and decoding attacks, but inflate key sizes and reduce efficiency. Table 1 summarizes the selected parameters and the corresponding decoding complexity, ensuring a conservative security margin under both classical and quantum models.

Our security proof establishes polynomial-time reductions from the extended Rank Syndrome Decoding (extRSD) problem to the collision resistance of the proposed chameleon hash function. However, two limitations merit further study: (1) the concrete hardness of extRSD with sparse matrices requires deeper cryptanalytic analysis, and (2) the quadratic growth of key sizes with n may limit deployment in ultra-constrained environments. Future work will explore hybrid constructions combining rank and lattice-based primitives to optimize this tradeoff.

Table 1. Parameter selection and security analysis

n	m	r	Work factor (bits)	Public key (MB)	Signature (KB)
128	24	12	128	1.5	2.6
192	32	16	192	3.2	4.1
256	32	20	256	6.8	5.8

3.8 *A constructive framework for chameleon hash functions*

This section outlines a method for deriving Chameleon hash functions grounded in the KKS signature scheme over the finite extension field $\mathbb{F}_{q^m}$. To facilitate the construction, we introduce two subsets of $\mathbb{F}_{q^m}^N$ defined by their rank weight constraints:

$$\text{For } c \in \mathbb{F}_{q^m}^N, \quad L_d := \{c \mid \|c\| = d\}, \quad L_{[l_1,\, l_2]} := \{c \mid l_1 < \|c\| < l_2\}.$$

In particular, by applying the permutation techniques described in Section 2.8 and the structural properties formalized in Definition 2.11 we derive the functional mapping used in the hash construction.

Definition 3.1 For $x \in \mathbb{F}_{q^m}^k$, $c \in L_d$, let $f : \mathbb{F}_{q^m}^k \times L_d \to \mathbb{F}_{q^m}^{N-K}$ be defined as

$$f(ppk,\, x,\, c) = Q \cdot (S_1 \star xP_1)^T + H \cdot (S_2 \star cP_2)^T,$$

where $ppk = (H,\, Q)$ is the public key derived from the KKS framework (see Section 3.1). Here, $P_1 \in GL(k,\, \mathbb{F}_{q^m})$, $P_2 \in GL(N,\, \mathbb{F}_{q^m})$ are random invertible matrices. Additionally, $S_1,\, S_2 \in GL(m,\, \mathbb{F}_q)$ are linear operators over the base field.

In the special case where $P_1,\, P_2,\, S_1$ and S_2 are identity matrices, the function simplifies to:

$$f(ppk,\, x,\, c) = Q \cdot x^T + H \cdot c^T.$$

Solution 1 Given the function f as introduced in Definition 3.1, we now present a procedure for constructing a trapdoor collision.

Assume $(x_1,\, c_1) \in \mathbb{F}_{q^m}^k \times L_d$ and another input $x_2 \neq x_1$. We define the trapdoor collision function $\mathrm{Coll}(psk,\, (x_1,\, c_1),\, x_2)$ through the following process:

(i) Compute $v^T = (S_1 \star x_1 P_1 - S_1 \star x_2 P_1)^T \in \mathbb{F}_{q^m}^k$ using the trapdoor matrices S_1 and P_1.

(ii) Solve $Q \cdot v^T = H \cdot c^T$ to obtain $c \in L_{[l_1,\, l_2]}$. This aligns with the signature generation mechanism in the KKS framework, where c serves as the signature for vector v.

(iii) Compute c_2 by solving $c^T = (S_2 \star c_2 P_2 - S_2 \star c_1 P_2)^T$ using the trapdoor matrices $S_2,\, P_2$.

The final output is set as $c_2 = \mathrm{Coll}(psk,\, (x_1,\, c_1),\, x_2)$.

A collision in this setting refers to the case where two distinct inputs $(x_1,\, c_1) \neq (x_2,\, c_2)$ for $x_1,\, x_2 \in \mathbb{F}_{q^m}^k$, yield the same function output:

$$f(ppk,\, x_1,\, c_1) = f(ppk,\, x_2,\, c_2),$$

where $c_1 \in L_d$, and $c_2 \in L_{[l_1,\, l_2]}$.

To ensure that collisions do not undermine the security of the function, f must guarantee that the outputs resulting from inputs $c_1 \leftarrow L_d$ and $c_2 \leftarrow L_{[l_1,\, l_2]}$ are indistinguishable to any efficient adversary, satisfying

$$\{f(ppk,\, x_1,\, c_1) \mid c_1 \leftarrow L_d\} \equiv \left\{f(ppk,\, x_2,\, c_2) \mid c_2 \leftarrow L_{[l_1,\, l_2]}\right\}.$$

If this indistinguishability condition holds, then f can be regarded as a valid Chameleon hash function.

Let x_1, x_2 and c_1 be as above and $x_2 \neq x_1$, one can compute c_2 via the trapdoor mechanism as $c_2 = \mathrm{Coll}(psk,\, (x_1,\, c_1),\, x_2)$. By invoking the triangle inequality, we obtain:

$$l - l_2 < \|c_2\| < l + l_2.$$

Finally, we verify that:

$$\begin{aligned}
f(ppk,\, x_2,\, c_2) &= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot (S_2 \star c_2 P_2)^T \\
&= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot (S_2 \star (c_1 P_2 + c))^T \\
&= Q \cdot (S_1 \star (x_1 P_1 - v))^T + H \cdot (S_2 \star c_1 P_2)^T + Hc^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T - Q \cdot v^T + H \cdot (S_2 \star c_1 P_2)^T + Hc^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T + H \cdot (S_2 \star c_1 P_2)^T \\
&= f(ppk,\, x_1,\, c_1)
\end{aligned}$$

Let $\mathbb{E}$ be an extension field of degree N over $\mathbb{F} = \mathbb{F}_{q^m}$, where q denotes a prime or a prime power. Suppose $g(X) \in \mathbb{F}[X]$ is an irreducible polynomial of degree N over $\mathbb{F}$. Then the field $\mathbb{E} = \mathbb{F}[X]/(g(X))$ constitutes an extension of degree N over $\mathbb{F}$.

Proposition 3.1 [21] Let $C = [n;\, k]$ denote a binary linear code sampled uniformly at random, and let d be an integer such that $0 < d \leq \frac{n}{2}$. Then, the probability that the minimum distance d_C satisfies $d_C \geq d$ is at least

$$1 - 2^{-(n-k) + nh_2\left(\frac{d-1}{n}\right)},$$

where h_2 represents the binary entropy function.

Proposition 3.2 [21] Let $C = [n;\, k]$ denote a binary linear code in systematic form, chosen uniformly at random. Suppose the parameters satisfy $0 < l_1 \leq \frac{n}{2} \leq l_2 \leq n$ and $l_1 + l_2 = n$. Then, with high probability, every nonzero codeword in C has Hamming weight falling within the interval $[l_1,\, l_2]$ with probability at least

$$1-2^{-(n-k)+1+nh_2\left(\frac{l_2-l_1}{n}\right)},$$

where h_2 denotes the binary entropy function.

Theorem 3.1 Let $N,\ K,\ k,\ l_1,\ l_2,\ l$ be positive integers satisfying $l_1 < l_2 < l$, and let h_2 denote the binary entropy function. Assume the inequality holds. Additionally, assume that solving the problems $DSD(N,\ K,\ l)$ and $extDSD(N,\ K,\ l-l_2,\ l+l_2)$ is infeasible for polynomial-time adversaries, and that the KKS signature mechanism guarantees one-time unforgeability under the parameters $l_1,\ l_2,\ Q,\ H$.

$$N-K-N\cdot h_2\left(\frac{4l+2l_2}{N}\right)>\lambda$$

Under these conditions, the function f (as introduced in Definition 3.1) can be regarded as a secure Chameleon hash function.

Proof. We demonstrate f fulfills the three fundamental criteria of a Chameleon hash function.

Collision Resistance: Under the KKS assumption detailed in Section 3.1, any adversary capable of producing a collision for f would effectively violate the underlying security of the KKS scheme. Namely, if an adversary can find distinct pairs that hash to the same value, it would contradict the presumed hardness of the KKS assumption. Hence, f is collision resistant.

Trapdoor Collision: Assume there exists an adversary $\mathscr{A}$ that, given $(H,\ Q)$, outputs two distinct inputs

$$(x_1,\ c_1)\in\mathbb{F}_{q^m}^k\times L_l \quad \text{and} \quad (x_2,\ c_2)\in\mathbb{F}_{q^m}^k\times L_{[l-l_2,\ l+l_2]} \quad \text{where } x_1\neq x_2,$$

such that their evaluations under f coincide:

$$f(ppk,\ x_1,\ c_1)=f(ppk,\ x_2,\ c_2). \tag{1}$$

We now present a reduction that builds an algorithm $\mathscr{M}$ using $\mathscr{A}$ to forge a KKS signature with non-negligible advantage ε. Upon receiving $(H,\ Q)$ from the scheme's setup, $\mathscr{M}$ executes the following steps:

1. It forwards $(H,\ Q)$ to $\mathscr{A}$.
2. Upon receiving the pairs $(x_1,\ c_1)$ and $(x_2,\ c_2)$ from $\mathscr{A}$, it computes $\sigma=c_1-c_2$.
3. Using the KKS signing procedure, $\mathscr{M}$ generates a forged message-signature pair, thereby breaking the scheme.

Define

$$v=S_1\star x_1P_1-S_1\star x_2P_1.$$

Consider the equation

$$Q\cdot v^T=H\cdot c^T,\quad \text{for } 0<\|c\|<2l+2l_2. \tag{2}$$

Since $c_1 \in L_d$ and $c_2 \in L_{[l-l_2,\, l+l_2]}$, their difference $c = c_1 - c_2$ satisfies the required norm constraint, ensuring that Equation (2) is valid. Furthermore, by Proposition 3.1, the linear code associated with H achieves a minimum distance of at least $4l + 2l_2$ with overwhelming probability:

$$1 - 2^{-(N-K)+Nh_2\left(\frac{4l+2l_2}{N}\right)} > 1 - 2^{-\lambda}.$$

Thus, with high probability Equation (2) admits a unique solution, and by the trapdoor mechanism this solution lies in $L_{[l_1,\, l_2]}$. Consequently, the computed σ satisfies $0 < \|\sigma\| < l_2$, meaning that a valid collision (in terms of forging a signature) is produced only if the KKS scheme is broken. This contradiction establishes the trapdoor collision property.

Uniformity: Define the distribution

$$D_1 = \{f(ppk,\, x_1,\, c_1) \mid c_1 \leftarrow L_d\}.$$

Let W_i (for $i = 1,\, 2$) be the distribution induced by evaluating $f(ppk,\, x_i,\, c)$, where c is sampled uniformly from $\mathbb{F}_{q^m}^N$. Given that H has full rank, for any fixed $s \in \mathbb{F}_{q^m}^{N-K}$, the following holds:

$$\Pr_{c \leftarrow \mathbb{F}_{q^m}^N}[f(ppk,\, x_1,\, c) = s^T] = \frac{1}{q^{mK}},$$

and the same result applies when x_1 is replaced by x_2, implying $W_1 \equiv W_2$. Furthermore, under the assumed hardness of the $DSD(N;\, K,\, l)$ and $extDSD(N;\, K,\, l-l_2,\, l+l_2)$ problems, the output distributions of f remain computationally indistinguishable for distinct inputs. This confirms the uniformity property.

Since f meets collision resistance, trapdoor collision, and uniformity, it functions as a Chameleon hash function. □

Theorem 3.2 Let $C = [n;\, k]$ denote a randomly selected binary code, and let d be a positive integer such that $0 < d \le \dfrac{n}{2}$. Then, the likelihood that the minimum distance d_C satisfies $d_C \ge d$ is lower bounded by

$$1 - q^{-(n-k)+nh_q\left(\frac{d-1}{n}\right)},$$

where the q-ary entropy function $h_q(x)$ takes the form:

$$h_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x), \quad \text{for } 0 < x < 1.$$

Proof. Note that an $[n;\, k]$ linear code corresponds to a k-dimensional subspace of $\mathbb{F}_q^n$, where q is a prime or a prime power. For any nonzero $x \in \mathbb{F}_q^n$, define the Hamming ball of radius r centered at x by

$$B_r(x) = \{v \in \mathbb{F}_q^n \mid \|x - v\| \le r\}.$$

We first present the following lemma as a foundation for the proof.

Lemma 3.1 The total number of k-dimensional subspaces in $\mathbb{F}_q^n$ that contain a fixed nonzero vector x is

$$\frac{(q^n-q)(q^n-q^2)\cdots(q^n-q^{k-1})}{(q^k-q)(q^k-q^2)\cdots(q^k-q^{k-1})}.$$

Proof. One can construct

$$(q^n-1)(q^n-q)(q^n-q^2)\cdots(q^n-q^{k-1})$$

ways to choose an ordered set of k linearly independent vectors in $\mathbb{F}_q^n$. In any fixed k-dimensional subspace, the number of ordered bases is

$$(q^k-1)(q^k-q)(q^k-q^2)\cdots(q^k-q^{k-1}).$$

Thus, the total number of distinct k-dimensional subspaces is the ratio

$$\frac{(q^n-1)(q^n-q)(q^n-q^2)\cdots(q^n-q^{k-1})}{(q^k-1)(q^k-q)(q^k-q^2)\cdots(q^k-q^{k-1})}$$

of these two quantities. Restricting to subspaces containing a fixed vector x yields the stated formula. □

Using Lemma 3.1, the likelihood that a k-dimensional subspace $V \subseteq \mathbb{F}_q^n$, selected uniformly at random, includes a given nonzero vector x is

$$\Pr[x \in V] = \frac{q^k-1}{q^n-1}.$$

Consequently, the probability that $x \notin V$ is

$$\Pr[x \notin V] = 1 - \frac{q^k-1}{q^n-1} \geq 1 - \frac{q^k}{q^n} = 1 - \frac{1}{q^{n-k}}.$$

Now, for any subset $U \subset \mathbb{F}_q^n$ not containing 0, by applying the union bound (or a simple independence argument) we deduce that the likelihood that a randomly selected subspace V of dimension k fails to intersect U satisfies

$$\Pr[U \cap V = \emptyset] \geq \left(1 - \frac{1}{q^{n-k}}\right)^{|U|}.$$

In particular, let $U = B_{l-1}(0) \setminus \{0\}$. Then the probability that V contains no nonzero vector from $B_{l-1}(0)$ is at least

$$\Pr[B_{l-1}(0) \cap V = \{0\}] \geq \left(1 - \frac{1}{q^{n-k}}\right)^{|B_{l-1}(0)|-1} \geq 1 - \frac{|B_{l-1}(0)|-1}{q^{n-k}} \geq 1 - \frac{|B_{l-1}(0)|}{q^{n-k}}.$$

This completes the proof. □

Lemma 3.2 Let $l \leq \frac{n}{2}$ be a positive integer. In this case, $|B_l(0)| \leq q^{nh_q\left(\frac{l}{n}\right)}$ where $h_q(x) = xlog_q(q-1) - xlog_q(x)$-(1-$x$)$log_q(1-x)$ for $0 < x < 1$

Proof. Assume q is a prime number. In this case, we adopt the Hamming metric, we can obtain:

$$|B_l(0)| = \binom{n}{0} + \cdots + \binom{n}{l}.$$

Define the binary entropy function for $0 < x < 1$ as

$$h(x) = -x\log_q(x) - (1-x)\log_q(1-x).$$

Then, noting that the q-ary entropy function can be written as

$$q^{nh_q\left(\frac{l}{n}\right)} = q^{l\log_q(q-1)} \cdot q^{nh\left(\frac{l}{n}\right)} = (q-1)^l \cdot q^{nh\left(\frac{l}{n}\right)} \geq q^{nh\left(\frac{l}{n}\right)},$$

we next focus on expressing $q^{nh(l/n)}$ in closed form. By the definition of $h(x)$, we have

$$q^{nh\left(\frac{l}{n}\right)} = q^{n\left(-\frac{l}{n}\log_q\frac{l}{n} - \left(1-\frac{l}{n}\right)\log_q\left(1-\frac{l}{n}\right)\right)} = \left(\frac{\left(1-\frac{l}{n}\right)^{n-l}}{l^l}\right)^n,$$

which, after rearrangement, yields

$$q^{nh\left(\frac{l}{n}\right)} = \frac{n^n}{l^l(n-l)^{n-l}}.$$

For convenience, define

$$\mathscr{M}(n,\, l) := \frac{n^n}{l^l(n-l)^{n-l}}.$$

Thus, the inequality in the lemma is equivalent to

$$\sum_{i=0}^{l} \binom{n}{i} \leq \mathscr{M}(n,\, l).$$

To prove this, observe that when $l \leq \frac{n}{2}$, for any integer $0 \leq i \leq l$ we have

$$l^l(n-l)^{n-l} \leq l^i(n-l)^{n-i}.$$

Multiplying the sum by $l^l(n-l)^{n-l}$ gives

$$l^l(n-l)^{n-l} \cdot \sum_{i=0}^{l} \binom{n}{i} \leq \sum_{i=0}^{l} \binom{n}{i} l^i(n-l)^{n-i}.$$

Since for each i, the term $\binom{n}{i} l^i(n-l)^{n-i}$ is bounded above by n^n, the entire sum is strictly less than n^n. Dividing both sides by $l^l(n-l)^{n-l}$, we obtain

$$\sum_{i=0}^{l} \binom{n}{i} < \mathscr{M}(n,\, l),$$

which is the desired inequality.

Assume q is a power of a prime number p, we set $q = p^m$ and the metric is the rank metric. For a given $0 \neq x \in B_l(0)$, we set its corresponding matrix to be $M = (m_{ij})_{n\times m}$ and $m_{ij} \in \mathbb{F}_q$. We assume $n \leq m$, and therefore:

$$|B_l(0)| = \binom{n}{0} + \cdots + \binom{n}{l}.$$

By the definition of $h(x)$, $h_q(x)$ and the previous process of calculating $q^{nh_q\left(\frac{l}{n}\right)}$, we can also obtain the conclusion. If $n > m$, we have:

$$|B_l(0)| = \binom{m}{0} + \cdots + \binom{m}{l} < \binom{n}{0} + \cdots + \binom{n}{l}.$$

Thus, the conclusion is clearly established. □

Theorem 3.3 Let $C = [n;\, k]$ be a randomly generated binary code expressed in systematic form. Suppose that the positive integers l_1 and l_2 satisfy

$$0 < l_1 < \frac{n}{2} < l_2 < n, \quad \text{with } l_1 + l_2 = n.$$

Then, the likelihood that every nonzero codeword in C possesses a Hamming weight falling within the range $[l_1,\, l_2]$ is no less than

$$1 - q^{-(n-k)+nh\left(\frac{l_1-1}{n}\right)} - q^{-(n-k)+nh\left(\frac{n-l_2-1}{n}\right)},$$

Proof. Throughout this proof, the V is considered to be a random k-dimensional subspace of $\mathbb{F}_q^n$. If q is a prime number, the metric will be the Hamming metric. For $\frac{n}{2} < l < n$ and $x = (x_1, \ldots, x_n) \in V$, we define

$$e = (e_i) \quad \text{where} \quad e_i = \begin{cases} 1, & \text{for } x_i = 0, \\ \\ x_i, & \text{for } x_i \neq 0 \end{cases}$$

and

$$B_{n-l}(e) = \{x \in \mathbb{F}_q^n \mid w(x) \geq l\}.$$

Obviously, we have

$$B_{n-l}(e) = \{x \in \mathbb{F}_q^n \mid d(x, e) \leq n - l\}.$$

By Lemma 3.2, we have

$$|B_{n-l}(e)| \leq q^{nh\left(\frac{n-l}{n}\right)}.$$

It is easily obtained that $V \cap B_{n-l}(e) = \emptyset$ is equivalent to $V \subseteq B_{l-1}(e)$.
Set $l = l_2 + 1$ and

$$U = B_{l_1}(\mathbf{0}) \cup \left(B_{n-l_2}(\boldsymbol{e}) \setminus \{\mathbf{0}\}\right).$$

Then, the event

$$V \subseteq \left(B_{l_2}(\mathbf{0}) \setminus B_{l_1}(\mathbf{0})\right) \cup \{\mathbf{0}\}$$

can be rephrased as the condition

$$V \cap U = \emptyset.$$

From Equation (1), we note that

$$|U| \leq q^{nh\left(\frac{l_1-1}{n}\right)} + q^{nh\left(\frac{n-l_2-1}{n}\right)} - 1.$$

We obtain:

$$P\{V \cap U = \emptyset\} \geq \left(1 - \frac{1}{q^{n-k}}\right)^{q^{nh\left(\frac{l_1-1}{n}\right)} + q^{nh\left(\frac{n-l_2-1}{n}\right)} - 1}$$

$$\geq \left(1 - \frac{1}{q^{n-k}}\right)^{q^{nh\left(\frac{l_1-1}{n}\right)}} \left(1 - \frac{1}{q^{n-k}}\right)^{q^{nh\left(\frac{n-l_2-1}{n}\right)}}$$

$$\geq \left(1 - \frac{q^{nh\left(\frac{l_1-1}{n}\right)}}{q^{n-k}}\right)\left(1 - \frac{q^{nh\left(\frac{n-l_2-1}{n}\right)}}{q^{n-k}}\right)$$

$$\geq \left(1 - q^{-(n-k)+nh\left(\frac{l_1-1}{n}\right)}\right)\left(1 - q^{-(n-k)+nh\left(\frac{n-l_2-1}{n}\right)}\right)$$

$$\geq 1 - q^{-(n-k)+nh\left(\frac{l_1-1}{n}\right)} - q^{-(n-k)+nh\left(\frac{n-l_2-1}{n}\right)}.$$

Therefore,

$$\Pr\left\{V \subseteq \left(B_{l_2}(\mathbf{0}) \setminus B_{l_1}(\mathbf{0})\right) \cup \{\mathbf{0}\}\right\} \geq 1 - q^{-(n-k)+nh\left(\frac{l_1-1}{n}\right)} - q^{-(n-k)+nh\left(\frac{n-l_2-1}{n}\right)}.$$

We establish the theorem when q is prime, in which case the metric shall be the Hamming metric.

Consider rank metric codes over a finite field of order q, where q is taken to be a power of a prime. Given any $x = (x_1, \ldots, x_n) \in V$, we set its corresponding matrix to be $M_{n\times m}$, where $n \leq m$. By applying Gauss transformation, we can rewrite $M_{n\times m}$ in the following form:

$$M = \begin{bmatrix} E_r & * \\ 0 & 0 \end{bmatrix}$$

where E_r is an r-order unit matrix, and $r = \text{rank}\ (M_{n\times m})$.

$$M = \begin{bmatrix} E_r & * \\ 0 & 0 \end{bmatrix}$$

Next, consider M as the matrix corresponding to x.

We set

$$E_1 = \begin{bmatrix} E_n & * \end{bmatrix}_{n\times m}$$

and its corresponding vector is $e \in \mathbb{F}_q^n$. We can obtain that the corresponding matrix of $x - e$ is $M - E_1$.

For $\dfrac{n}{2} < l < n$, define

$$B_{n-l}(e) = \{x \in \mathbb{F}_q^n \mid w(x) \geq l\}.$$

Obviously, we have

$$B_{n-l}(e) = \{x \in \mathbb{F}_q^n \mid d(x,\ e) \leq n - l\}.$$

By Lemma 3.2 and repeat the previous steps, the conclusion can be easily obtained.

If $n > m$, set

$$E_1 = \begin{bmatrix} E_m \\ 0 \end{bmatrix}.$$

and its corresponding vector is $e \in \mathbb{F}_q^n$.

For $\dfrac{n}{2} < l < n$, define $B_{n-l}(e) = \{x \in \mathbb{F}_q^n |\ w(x) \geq l\}$.

Obviously, we have $B_{n-l}(e) = \{x \in \mathbb{F}_q^n |\ d(x,\ e) \leq m - l\} \subseteq \{x \in \mathbb{F}_q^n |\ d(x,\ e) \leq n - l\}$.

By Lemma 3.2, we have $|B_{n-l}(e)| \leq q^{nh(\frac{n-l}{n})}$. Repeat the previous steps, we can prove the theorem. □

Remark 3.1 According to Theorem 3.2, by replacing h_2 with h_q in Theorem 3.1, we can similarly demonstrate that the result of Theorem 3.1 remains valid. So we can rewrite Theorem 3.1 as follows.

Theorem 3.1$'$

Let $N,\ K,\ k,\ l_1,\ l_2,\ l$ be positive integers satisfying $l_1 < l_2 < l$ and h denote the entropy function. Suppose the inequality $N - K - N \cdot h_2\left(\dfrac{4l + 2l_2}{N}\right) > \lambda$ holds. Additionally, assuming $DSD(N,\ K,\ l)$ or $DRSD(N,\ K,\ l)$ and $extDSD(N,\ K,\ l - l_2,\ l + l_2)$ problem are computationally hard and that the KKS signature scheme is one-time secure for parameters $l_1,\ l_2,\ Q,\ H$.

Given these conditions, it can be concluded that the function f in Definition 3.1 functions as a Chameleon hash function.

Definition 3.2 Let $r,\ s,\ u,\ m,\ n$ be positive integers such that $m = s(u + v)$, and n satisfies the inequality

$$(n - r(u + v - s)) \cdot (n - r(u + v - s) + 1) \leq 2m.$$

We define the matrices as follows:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1r} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{sr} \end{pmatrix} \in \mathbb{F}^{s \times r},$$

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1u} \\ \vdots & \ddots & \vdots \\ b_{r1} & \cdots & b_{ru} \end{pmatrix} \in \mathbb{F}^{r\times u},$$

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1v} \\ \vdots & \ddots & \vdots \\ c_{r1} & \cdots & c_{rv} \end{pmatrix} \in \mathbb{F}^{r\times v}.$$

Here, A is of dimension $s \times r$, while B and C are both $r \times u$ and $r \times v$ matrices, respectively. Each entry a_{ij} is sampled uniformly from the set $\{x_1, \ldots, x_n\}$, whereas the entries b_{ij} and c_{ij} are linear combinations over the same variable set. When $u = v$, B and C share the same dimensions, forming a symmetric instance.

Define $E_1 = A \cdot B$ and $E_2 = A \cdot C$, and let the map $P = (E_1, E_2) : \mathbb{F}^n \to \mathbb{F}^m$. This map consists of $m = s(u+v)$ components, each being a homogeneous quadratic polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. The associated quadratic forms are designed to have rank approximately $2r$.

Let $S : \mathbb{F}^{2n} \to \mathbb{F}^{2n}$ and $T : \mathbb{F}^{2n} \to \mathbb{F}^{2n}$ be two affine invertible maps. We define the obfuscated polynomial map $\bar{P}$ by composition:

$$\bar{P} = S \circ P \circ T.$$

To use P or its disguised version $\bar{P}$ in our signature construction, we next introduce a method for finding a solution to the equation $P(x) = y$. The procedure is as follows.

Solution 2 Next, we demonstrate the solution to the equation $P(x) = y$ to obtain a $x = (x_1, \ldots, x_n)$ for given $y = (y_1, \ldots, y_m)$.

1. In the first, we define two matrices as follows:

$$\bar{E}_1 = \begin{pmatrix} y_1 & y_2 & \cdots & y_u \\ y_{u+1} & y_{u+2} & \cdots & y_{2u} \\ \vdots & \vdots & \vdots & \vdots \\ y_{(s-1)u+1} & y_{(s-1)u+2} & \cdots & y_{su} \end{pmatrix} \in \mathbb{F}^{s\times u}$$

and

$$\bar{E}_2 = \begin{pmatrix} y_{su+1} & y_{su+2} & \cdots & y_{su+v} \\ y_{su+v+1} & y_{su+v+2} & \cdots & y_{su+2v} \\ \vdots & \vdots & \vdots & \vdots \\ y_{su+(s-1)v+1} & y_{su+(s-1)v+2} & \cdots & y_{su+sv} \end{pmatrix} \in \mathbb{F}^{s\times v}.$$

2. In the second stage, we aim to locate a vector $x \in \mathbb{F}^n$ satisfying $P(x) = y$. Define $\bar{A} = A(x)$. If $\bar{A}$ has full row rank r, then there exists a matrix $T \in \mathbb{F}^{r\times s}$ such that $T \cdot \bar{A} = I$, with I being the $r \times r$ identity.

Given $\bar{E}_1 = \bar{A}B$, $\bar{E}_2 = \bar{A}C$, we obtain:

$$T \cdot \bar{E}_1 = B, \quad T \cdot \bar{E}_2 = C.$$

Interpreting the entries of T as fresh variables, we construct a system of $r(u+v)$ equations involving $sr+n$ variables. Eliminating sr entries associated with T, we derive approximately $r(u+v-s)$ equations among the residual variables.

This system typically possesses low dimensionality. By Gaussian reduction (as described in Section 2.9), we can eliminate most variables-say γ-and express them via the remaining unknowns. Substituting these into the polynomial system P, we obtain m quadratic constraints over $n-\gamma$ variables, solvable by the relinearization method [22].

If $\text{Rank}(\bar{A}) < r$, reinitialize $A,\ B,\ C$ and restart. In the case of $\bar{P}$, we solve $\bar{P}(x) = y$ by computing $S^{-1}(y) = z$, solving $P(w) = z$, then deriving $x = T^{-1}(w)$.

Definition 3.3 Given a pair $(x,\ c) \in \mathbb{F}_q^k \times L_d$, we define the mapping

$$f : \mathbb{F}_{q^m}^k \times L_d \to \mathbb{F}_{q^m}^{N-K}, \quad f(ppk,\ x,\ c) = Q \cdot (S_1 \star x P_1)^T + H \cdot \bar{P}(c^T),$$

where the tuple $ppk = (H,\ Q)$ originates from the KKS framework construction described in Section 3.1. The nonlinear transformation $\bar{P}$ corresponds to Definition 3.2. Furthermore, matrices satisfy $P_1 \in GL(k,\ \mathbb{F}_{q^m})$ and $S_1 \in GL(m,\ \mathbb{F}_q)$.

Solution 3 To establish the trapdoor collision property of the function f, we present a constructive procedure for generating such collisions.

Assume an input of the form $(x_1,\ c_1) \in \mathbb{F}_{q^m}^k \times L_d$ is available and and another vector x_2 such that $x_2 \neq x_1$. We then define the collision generation algorithm $Coll(psk,\ (x_1,\ c_1),\ x_2)$ as follows:

(i) Compute the vector

$$v^T = \Big(S_1 \star x_1 P_1 - S_1 \star x_2 P_1\Big)^T \in \mathbb{F}_{q^m}^k,$$

using the trapdoor matrices S_1 and P_1.

(ii) Solve the linear equation

$$Q \cdot v^T = H \cdot c^T,$$

to determine a value $c \in L_{[l_1,\ l_2]}$. This step employs the signing procedure of the KKS scheme as described in Section 3.1, where c serves as the signature corresponding to v.

(iii) Obtain c_2 by solving

$$\bar{P}(c_2^T) = c^T + \bar{P}(c_1^T).$$

The solution method for this equation follows the approach detailed in Solution 2 for equations of the form $\bar{P}(x) = y$. Finally, set

$$c_2 = Coll(psk,\ (x_1,\ c_1),\ x_2).$$

The resulting collision is formed between the (x_1, c_1) and (x_2, c_2), with x_1, x_2, and c_1 defined as above, and $c_2 \in L_{[l_1, l_2]}$. To ensure uniformity, we require that for uniformly chosen x_1 and x_2, when c_1 is sampled from L_d and c_2 from $L_{[l_1, l_2]}$, then the output distributions of f remain indistinguishable to any efficient adversary:

$$\{f(ppk, x_1, c_1) \mid c_1 \leftarrow L_d\} \approx_{\text{comp}} \left\{f(ppk, x_2, c_2) \mid c_2 \leftarrow L_{[l_1, l_2]}\right\}.$$

Thus, under the trapdoor framework, f operates as a chameleon hash function.

Additionally, suppose we are provided with a specific input $(x_1, c_1) \in \mathbb{F}_{q^m}^k \times L_d$ and an alternative vector $x_2 \neq x_1$. Then, by invoking the trapdoor-based algorithm $\mathsf{Coll}(psk, (x_1, c_1), x_2)$, one can determine the corresponding c_2.

By the triangle inequality, it follows that

$$l - l_2 < \|c_2\| < l + l_2.$$

To verify correctness, observe that

$$\begin{aligned}
f(ppk, x_2, c_2) &= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot \bar{P}(c_2^T) \\
&= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot \bar{P}(c_1^T) + H \cdot c^T \\
&= Q \cdot \left(S_1 \star (x_1 P_1 - v)\right)^T + H \cdot \bar{P}(c_1^T) + H \cdot c^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T - Q \cdot v^T + H \cdot \bar{P}(c_1^T) + H \cdot c^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T + H \cdot \bar{P}(c_1^T) \\
&= f(ppk, x_1, c_1).
\end{aligned}$$

This confirms the validity of the trapdoor collision construction.

Theorem 3.4 Let N, K, k, l_1, l_2, l be positive integers such that $l_1 < l_2 < l$, and let $h(\cdot)$ denote the q-ary entropy function. Suppose the following inequality holds:

$$N - K - N \cdot h\left(\frac{4l + 2l_2}{N}\right) > \lambda.$$

Assume that $\mathscr{P}$ is one of the following hard problems:

DSD(N, K, l), *DRSD*(N, K, l), *extDSD*$(N, K, l - l_2, l + l_2)$, or *extDRSD*$(N, K, l - l_2, l + l_2)$. Also, the KKS signature scheme is assumed to be one-time strongly unforgeable for (l_1, l_2, Q, H). and that the KKS signature scheme is one-time strongly unforgeable under the parameters l_1, l_2, Q, H. Under these assumptions, the function f as introduced in Definition 3.8 satisfies the Chameleon hash function properties.

Proof. The conclusion follows directly from the construction of trapdoor collisions described in Solution 3, together with the security framework established in Theorem 3.8. The collision-resistance and trapdoor properties of f are thereby rigorously demonstrated.

Denote $\mathbb{F} := \mathbb{F}_q$, the finite field with q elements, let $o,\ v \in \mathbb{Z}$ be positive integers with $n = o + v$. Define the sets

$$V = \{1,\ 2,\ \ldots,\ v\} \quad \text{and} \quad O = \{v+1,\ v+2,\ \ldots,\ n\}.$$

Definition 3.4 Consider the map $\Gamma = (p_1,\ ...,\ p_o) : \mathbb{F}^n \to \mathbb{F}^o$, where the polynomials $p_1,\ ...,\ p_o$ are given by

$$p_i = \sum_{j,\ k \in V} \alpha_{j,\ k}^{(i)} x_j x_k + \sum_{j \in V,\ k \in O} \beta_{j,\ k}^{(i)} x_j x_k + \sum_{j \in V \cup O} \gamma_j^{(i)} x_j + \delta^{(i)}$$

where the coefficients $\alpha_{j,\ k}^{(i)}$, $\beta_{j,\ k}^{(i)}$, $\gamma_j^{(i)}$, and $\delta^{(i)}$ are randomly chosen from the field $\mathbb{F}$. Notably, the variables $x_1,\ ...,\ x_n$ are not fully mixed in the polynomials $p_1,\ ...,\ p_o$.

The polynomials $p_1,\ \ldots,\ p_o$ each contain a homogeneous quadratic component, representable in the form $\hat{p}_i = x^T P^i x$ for $i = 1,\ \ldots,\ o$. Owing to the structured nature of p_i, the associated coefficient matrices M^i take the block form:

$$\begin{bmatrix} A_{v\times v} & B_{v\times o} \\ C_{o\times v} & 0_{o\times o} \end{bmatrix}.$$

This configuration enables the efficient inversion of the quadratic transformation Γ, which forms an underdetermined system comprising o equations over $o + v$ unknowns. By assigning values to v of these variables (e.g., $x_1,\ \ldots,\ x_v$), the system reduces to o linear constraints involving the remaining o variables $x_{v+1},\ \ldots,\ x_n$, and then efficiently resolved using row echelon form transformation.

Remark 3.2 To solve the equation $\Gamma(y) = z$, one typically assigns values to the subset $y_1,\ \ldots,\ y_v$, and substitutes them into the polynomials $p_1,\ \ldots,\ p_o$. This transformation yields a system of o linear expressions in o unknowns $y_{v+1},\ \ldots,\ y_n$, which can subsequently be solved via matrix reduction process. If the system admits no solution, alternative values for $x_1,\ \ldots,\ x_v$ may be selected and the procedure repeated.

Definition 3.5 Given inputs $(x,\ c) \in \mathbb{F}_q^k \times L_d$, we construct the mapping

$$f : \mathbb{F}_{q^m}^k \times L_d \longrightarrow \mathbb{F}_{q^m}^{N-K},$$

$$f(ppk,\ x,\ c) = Q \cdot (S_1 \star x P_1)^T + H \cdot \Gamma(T(c^T))$$

Here, parameters $ppk = (H,\ Q)$ are taken from KKS construction introduced in Section 3.1. The component Γ is defined in Definition 3.4, with $P_1 \in GL(k,\ \mathbb{F}_{q^m})$, $S_1 \in GL(m,\ \mathbb{F}_q)$, and $T \in GL(N,\ \mathbb{F}_{q^m})$.

Solution 4 We propose a procedure for generating trapdoor-induced collisions in the mapping f described in Definition 3.8.

Consider two input pairs $(x_1,\ c_1) \in \mathbb{F}_{q^m}^k \times L_d$ and $x_2 \neq x_1$. Define a collision constructor $\mathsf{Coll}(psk,\ (x_1,\ c_1),\ x_2)$ as follows:

1. Compute $v^T = (S_1 \star x_1 P_1 - S_1 \star x_2 P_1)^T \in \mathbb{F}_{q^m}^k$ using the trapdoor elements S_1 and P_1.

2. Solve $Q \cdot v^T = H \cdot c^T$ for $c \in L_{[l_1,\ l_2]}$ using the KKS signing mechanism as described in Section 3.1, and associate c as the signature of v.

3. Determine c_2 by solving $\Gamma \circ T(c_2^T) = c^T + \Gamma \circ T(c_1^T)$, and set $c_2 := \mathsf{Coll}(psk,\ (x_1,\ c_1),\ x_2)$.

The constructed collision involves two distinct tuples $(x_1,\ c_1) \neq (x_2,\ c_2)$ such that

$$\{f(ppk,\ x_1,\ c_1) \mid c_1 \leftarrow L_d\} \equiv \left\{f(ppk,\ x_2,\ c_2) \mid c_2 \leftarrow L_{[l_1,\ l_2]}\right\},$$

thereby satisfying the uniformity condition characteristic of Chameleon hash functions.

Based on this construction, we can derive the equivalence:

$$\begin{aligned} f(ppk,\ x_2,\ c_2) &= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot \Gamma \circ T(c_2^T) \\ &= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot \Gamma \circ T(c_1^T) + H \cdot c^T \\ &= Q \cdot (S_1 \star x_1 P_1 - v)^T + H \cdot \Gamma \circ T(c_1^T) + H \cdot c^T \\ &= Q \cdot (S_1 \star x_1 P_1)^T - Q \cdot v^T + H \cdot \Gamma \circ T(c_1^T) + H \cdot c^T \\ &= Q \cdot (S_1 \star x_1 P_1)^T + H \cdot \Gamma \circ T(c_1^T) \\ &= f(ppk,\ x_1,\ c_1). \end{aligned}$$

Theorem 3.5 Let $N,\ K,\ k,\ l_1,\ l_2,\ l$ be positive integers with $l_1 < l_2 < l$, and let h denote the entropy function. Assume the inequality

$$N - K - N \cdot h\left(\frac{4l + 2l_2}{N}\right) > \lambda$$

holds. Further suppose that one of the following problems is computationally hard:

$$\mathscr{P} \in \{DSD(N,\ K,\ l),\ DRSD(N,\ K, l),\ extDSD(N,\ K,\ l{-}l_2,\ l{+}l_2),\ extDRSD(N,\ K,\ l{-}l_2,\ l{+}l_2)\},$$

and that the KKS signature scheme is one-time strongly unforgeable under parameters $l_1,\ l_2,\ Q,\ H$. Then, the function f in Definition 3.5 meets the properties of a Chameleon hash function.

Proof. The conclusion follows from the trapdoor collision procedure detailed in Solution 4, combined with the analysis in Theorem 3.8.

Next, we extend the construction in Definition 3.4 to a hierarchical multilayer framework. This enables a more compact design, reducing both key and signature sizes while improving the scheme's overall efficiency.

Let $V = \{1,\ 2,\ \ldots,\ n\}$, and suppose there exist integers $v_1 < v_2 < \cdots < v_u = n$, forming nested index subsets $V_l = \{1,\ 2,\ \ldots,\ v_l\}$ for $l = 1,\ \ldots,\ u$, with the inclusion chain:

$$V_1 \subset V_2 \subset \cdots \subset V_u = V.$$

The incremental size between successive layers is denoted $o_i = v_{i+1} - v_i$, and the corresponding difference sets are defined as $O_i = V_{i+1} \setminus V_i$, for $i = 1, \ldots, u-1$.

Definition 3.6 For each layer l, define the polynomial space f_l consisting of quadratic forms of the structure:

$$\sum_{i \in O_l,\ j \in V_l} \alpha_{i,\ j} x_i x_j + \sum_{i,\ j \in V_l} \beta_{i,\ j} x_i x_j + \sum_{i \in V_{l+1}} \gamma_i x_i + \eta,$$

where coefficients $\alpha_{i,\ j}$, $\beta_{i,\ j}$, $\gamma_i \in \mathbb{F}$, and $\eta \in \mathbb{F}$ are randomly selected.

Here, variables in O_l are referred to as O-variables of layer l, while those in V_l serve as its V-variables. The corresponding polynomials are termed l-layer OV-polynomials. Clearly, the inclusion $f_i \subset f_j$ holds for $i < j$, indicating nested structure across layers. This relationship is reflected by the recursive form:

$$V_{i+1} = V_i \cup O_i.$$

Definition 3.7 Let Γ be a mapping from k^n to k^{n-v_1}, defined as:

$$\Gamma(x_1, \ldots, x_n) = (\bar{\Gamma}_1(x_1, \ldots, x_n), \ldots, \bar{\Gamma}_{u-1}(x_1, \ldots, x_n)) = (\Gamma_1(x_1, \ldots, x_n), \ldots, \Gamma_{n-v_1}(x_1, \ldots, x_n)).$$

Here, function $\bar{\Gamma}_i$ comprises o_i independently chosen quadratic forms sampled from the polynomial family f_i, where the coefficients are randomly selected over a finite field.

The structure of Γ involves $u-1$ hierarchical layers. In the initial layer, the set $\{\Gamma_1, \ldots, \Gamma_{o_1}\}$ corresponds to o_1 polynomials, where $x_j \in O_1$ act as O-type variables and $x_j \in V_1$ serve as V-type variables. At layer i, the set $\{\Gamma_{v_i+1}, \ldots, \Gamma_{v_{i+1}}\}$ contains o_i polynomials, with O_i and V_i defining the respective O-and V-variable sets.

This layered formulation gives rise to the following incremental variable compositions:

$$[x_1, \ldots, x_{v_1}];\ \{x_{v_1+1}, \ldots, x_{v_2}\}$$

$$[x_1, \ldots, x_{v_2}];\ \{x_{v_2+1}, \ldots, x_{v_3}\}$$

$$[x_1, \ldots, x_{v_3}];\ \{x_{v_3+1}, \ldots, x_{v_4}\}$$

$$\vdots$$

$$[x_1, \ldots, x_{v_{u-1}}];\ \{x_{v_{u-1}+1}, \ldots, x_n\}$$

In this scheme, each row represents one hierarchical layer, with variables in brackets interpreted as V-type variables (used in the polynomial expressions), and those in braces indicating newly introduced O-type variables. The V-set at each level recursively accumulates all previous variables. Consequently, Γ establishes a polynomial transformation constructed over $u-1$ nested layers.

Solution 5 Let L_1 and L_2 be two independently chosen invertible affine linear transformations, where L_1 acts on k^{n-v_1} and L_2 on k^n. Given a target vector $Y' = (y'_1, \dots, y'_{n-v_1}) \in k^{n-v_1}$, our objective is to find a solution to the following system:

$$L_1 \circ \Gamma \circ L_2(x_1, \dots, x_n) = Y'.$$

We begin by composing both sides with L_1^{-1}, which results in:

$$\Gamma \circ L_2(x_1, \dots, x_n) = \bar{Y}',$$

where $\bar{Y}' = L_1^{-1} Y'$ is the transformed output.

To proceed, we must invert the map Γ, which reduces the task to solving:

$$\Gamma(x_1, \dots, x_n) = \bar{Y}' = (\bar{y}'_1, \dots, \bar{y}'_{n-v_1}).$$

The inversion procedure begins by assigning random values to the variables $x_1, \dots, x_{v_1}$, and evaluating the first component layer $\bar{\Gamma}_1$, which corresponds to the initial o_1 entries of $\bar{Y}'$, that is,

$$\bar{\Gamma}_1 = (\bar{y}'_1, \dots, \bar{y}'_{o_1}).$$

It leads to a a system of o_1 linear equations in o_1 unknowns $x_{o_1+1}, \dots, x_{v_2}$, which is then solved to determine corresponding variable assignments. As a result, all variables in the set V_2 are resolved.

Subsequently, these known values are plugged into the second layer of equations, yielding another linear system of size o_2, from which the values in V_3 can be derived. This process is repeated iteratively across layers until all coordinates $x_1, \dots, x_n$ have been computed.

If at any stage the resulting linear system lacks a solution, the procedure is restarted by sampling a new random initialization for $x_1, \dots, x_{v_1}$. According to the analysis in [23], this iterative decoding strategy succeeds with high probability, provided the number of layers remains within a tractable bound.

Finally, the inverse transformation L_2^{-1} is applied to the solution to recover the preimage $X' = (x'_1, \dots, x'_n)$, which constitutes a valid signature for the given output Y'.

If the dimension of Y' is large, the same strategy applies to Flash-type schemes [24], where one first hashes Y' and then signs the resulting digest. The following definition and theorem are derived from this framework.

Definition 3.8 Let $(x, c) \in \mathbb{F}_q^k \times L_d$ be a given input. The function f is defined by:

$$f : \mathbb{F}_{q^m}^k \times L_d \to \mathbb{F}_{q^m}^{N-K}, \quad f(ppk, x, c) = Q \cdot (V_1 \star x U_1)^T + H \cdot \bar{\Gamma} \circ S(c^T),$$

where parameter $ppk = (H, Q)$ is derived from the KKS signature system introduced in Section 3.1. The map Γ is defined in Definition 3.7, and the matrices satisfy $U_1 \in GL(k, \mathbb{F}_{q^m}),\ V_1 \in GL(m, \mathbb{F}_q)$, and $S \in GL(N, \mathbb{F}_{q^m})$.

Theorem 3.6 Let N, K, k, l_1, l_2, l be positive integers such that $l_1 < l_2 < l$, and let h represent the entropy function. Suppose the inequality

$$N - K - N \cdot h\left(\frac{4l + 2l_2}{N}\right) > \lambda$$

is satisfied. Assume further that the following computational problems are hard: $DSD(N,\ K,\ l)$, or both $DSD(N,\ K,\ l)$ and $extDSD(N,\ K,\ l-l_2,\ l+l_2)$, or alternatively, $extDRSD(N,\ K,\ l-l_2,\ l+l_2)$. Also suppose that the KKS signature scheme provides one-time strong unforgeability for the parameters $l_1,\ l_2,\ Q,\ H$. Then, under these conditions, the function f defined in Definition 3.8 behaves as a Chameleon hash function.

Definition 3.9 Let $\mathbb{E}$ be a finite field that extends $\mathbb{F}_q$ with degree n, and suppose there exists an isomorphism $\phi : \mathbb{F}_q^n \to \mathbb{E}$. We define a univariate polynomial transformation $\Psi : \mathbb{E} \to \mathbb{E}$ given by

$$\Psi(X) = \sum_{i,\ j=0}^{q_i+q_j \leq D} \alpha_{ij} X^{q_i+q_j} + \sum_{i=0}^{q_i \leq D} \beta_i X^{q_i} + \gamma,$$

where the coefficients $\alpha_{ij},\ \beta_i,\ \gamma$ are independently sampled from $\mathbb{E}$.

Given this structure, we define the induced map $\bar{\Psi} := \phi^{-1} \circ \Psi \circ \phi$, which acts as a quadratic mapping on the vector space $\mathbb{F}_q^n$. To conceal its internal algebraic structure, set $\Phi = S \circ \bar{\Psi} \circ T$, where S and T are invertible linear maps over $\mathbb{F}_q^n$. We define:

$$\Gamma := S \circ \bar{\Psi} \circ T.$$

This results in Γ being a quadratic transformation from $\mathbb{F}_q^n$ to $\mathbb{F}_q^n$.

Remark The structure of $\Gamma(x)$ admits several generalizations, including incorporating higher-order terms into Ψ or modifying the underlying field. The degree parameter D significantly impacts the invertibility of $\Psi(X)$, which is critical in the context of signature generation.

Specifically, inverting Ψ requires solving a univariate polynomial equation of degree D, and the computational burden is closely tied to the size of D. A large D increases complexity, so it must be selected judiciously to balance security and efficiency.

Since Ψ is not a bijective map, not every $w \in \mathbb{F}_q^n$ will necessarily admit a preimage z such that $\Gamma(z) = w$. To resolve this, we employ a retry mechanism by introducing a counter $r \in \mathbb{N}$.

If the inversion fails for a given w, we increment r and compute a new target $w \leftarrow h(w \,\|\, r)$ using a hash function h. The procedure is iterated until a valid preimage z is found such that $\Gamma(z) = w$.

Solution 6 We now outline the process for solving the equation $\Gamma(z) = w$:

1. First, invert the affine map S to derive:

$$x = S^{-1}(w) \in \mathbb{F}_q^n,$$

then transform the result into the extension field $\mathbb{E}$ through the isomorphism ϕ, i.e.,

$$X = \phi(x).$$

2. Resolve the univariate polynomial equation $\Psi(Y) = X$ over $\mathbb{E}$ by means of algorithms like Berlekamp's method [25] or the Cantor-Zassenhaus algorithm [26]. If no solution $Y \in \mathbb{E}$ is identified, increment a counter $r = 0, 1, \ldots,$ compute a fresh hash value

$$w = h(w \parallel r),$$

and restart from step 1.

3. Once a valid Y is identified, use the inverse of ϕ to revert back to the vector space:

$$y = \phi^{-1}(Y) \in \mathbb{F}_q^n,$$

then utilize the inverse of T to derive the signature:

$$z = T^{-1}(y).$$

Definition 3.10 Let $(x, c) \in \mathbb{F}_q^k \times L_d$ be a given input. We define the function as:

$$f : \mathbb{F}_{q^m}^k \times L_d \to \mathbb{F}_{q^m}^{N-K}, \quad f(ppk, x, c) = Q \cdot (S_1 \star xP_1)^T + H \cdot \Gamma(c^T),$$

where the public parameters $ppk = (H, Q)$ are obtained from the KKS framework described in Section 3.1, Γ is the nonlinear mapping from Definition 3.9, and the matrices satisfy $P_1 \in GL(k, \mathbb{F}_{q^m}),\ S_1 \in GL(m, \mathbb{F}_q)$.

Definition 3.11 We extend the univariate map $\Psi : E \times \mathbb{F}^v \to E$ to the following structured form:

$$\Psi(X, x_1, \ldots, x_v) = \sum_{\substack{q_i + q_j \leq D \\ i, j \geq 0}} \alpha_{i, j} X^{q_i + q_j} + \sum_{\substack{q_i \leq D \\ i \geq 0}} \beta_i(x_1, \ldots, x_v) X^{q_i} + \gamma(x_1, \ldots, x_v),$$

where the coefficients $\alpha_{i, j}$ are drawn uniformly at random from E, and the functions $\beta_i : \mathbb{F}^v \to E$ and $\gamma : \mathbb{F}^v \to E$ are linear and quadratic.

We define

$$\bar{\Psi} = \phi^{-1} \circ \Psi \circ (\phi \times \mathrm{id}_v),$$

resulting in a quadratic transformation from $\mathbb{F}^{n+v}$ to $\mathbb{F}^n$, where id_v is the identity on $\mathbb{F}^v$.

This transformed map is composed with two linear or affine mappings:

$$S : \mathbb{F}^n \to \mathbb{F}^{n-a} \quad \text{and} \quad T : \mathbb{F}^{n+v} \to \mathbb{F}^{n+v},$$

which yields the final mapping:

$$\Gamma = S \circ \phi^{-1} \circ \Psi \circ (\phi \times \mathrm{id}_v) \circ T : \mathbb{F}^{n+v} \to \mathbb{F}^{n-a}.$$

Solution 7 We now illustrate how to solve the equation $\Gamma(z) = w$, given $w \in \mathbb{F}_q^{n-a}$, using the structured map Γ defined previously.

1. Compute the preimage $x \in \mathbb{F}_q^n$ such that $S(x) = w$, where S is the affine transformation. Lift this vector to the extension field $\mathbb{E}$, and denote it by X.

2. Sample random values for $V = \{x_1, \ldots, x_v\}$ and plug them into the multivariate map Ψ to generate a univariate polynomial $\Psi_V : \mathbb{E} \to \mathbb{E}$ with parameters fixed by the chosen x_i.

3. Determine the solution of the equation $\Psi_V(\hat{Y}) = X$ over $\mathbb{E}$, e.g., via Berlekamp's algorithm [25]. If a root $Y \in \mathbb{E}$ is found, proceed. If no solution exists, return to step 2 and choose a new set of V.

4. Map Y back into the vector space via $y' = \phi^{-1}(Y) \in \mathbb{F}_q^n$, then concatenate the sampled variables $x_1, \ldots, x_v$ to construct

$$y = (y' \parallel x_1 \parallel \cdots \parallel x_v) \in \mathbb{F}_q^{n+v}.$$

5. Finally, reverse the linear transformation T to recover the signature:

$$z = T^{-1}(y).$$

Definition 3.12 Let $(x, c) \in \mathbb{F}_q^k \times L_d$. We define the function

$$f : \mathbb{F}_{q^m}^k \times L_d \to \mathbb{F}_{q^m}^{N-K}$$

as

$$f(ppk, x, c) = Q \cdot (S_1 \star x P_1)^T + H \cdot \Gamma(c^T),$$

where $ppk = (H, Q)$ is the public parameter generated by the KKS scheme introduced in Section 3.1. The map Γ is specified in Definition 3.11. Here, $P_1 \in GL(k, \mathbb{F}_{q^m})$ and $S_1 \in GL(m, \mathbb{F}_q)$ are randomly chosen invertible matrices.

Solution 8 To identify trapdoor collisions for the f in Definitions 3.10 and 3.12, we proceed as follows:

Assume we are given $(x_1, c_1) \in \mathbb{F}_{q^m}^k \times L_d$, a distinct input $x_2 \neq x_1$. Introduce a trapdoor-based collision mapping

$$\mathrm{Coll}(\mathrm{psk}, (x_1, c_1), x_2)$$

1. Compute the difference vector:

$$v^T = (S_1 \star x_1 P_1 - S_1 \star x_2 P_1)^T \in \mathbb{F}_{q^m}^k$$

using the known trapdoor components S_1 and P_1.

2. Solve the following linear system:

$$Q \cdot v^T = H \cdot c^T$$

to obtain $c \in L_{[l_1,\, l_2]}$. This step utilizes the signing procedure of the KKS scheme (see Section 3.1) by treating v as the message and computing its corresponding signature c.

3. Solve the equation:

$$\Gamma(c_2^T) = c^T + \Gamma(c_1^T)$$

to determine c_2. The computation follows the inversion methods detailed in Solutions 6 and 7, where the goal is to invert $\Gamma(z) = w$ for $w = c^T + \Gamma(c_1^T)$. Finally, we assign $c_2 = \text{Coll}(\text{psk},\, (x_1,\, c_1),\, x_2)$.

This results in a collision $(x_1,\, c_1) \neq (x_2,\, c_2)$ such that both values map to the same hash output. For inputs x_1, x_2 and $c_1, c_2 \in L_{[l_1,\, l_2]}$, the indistinguishability requirement implies:

$$\{f(\text{ppk},\, x_1,\, c_1) \mid c_1 \leftarrow L_d\} \equiv \left\{f(\text{ppk},\, x_2,\, c_2) \mid c_2 \leftarrow L_{[l_1,\, l_2]}\right\}$$

ensuring computational indistinguishability between the two distributions.

Hence, f satisfies the criteria of a Chameleon hash function.

Moreover, for a pair $(x_1,\, c_1)$ and another x_2, the generated value c_2 via the trapdoor collision process satisfies:

$$l - l_2 < \|c_2\| < l + l_2$$

by the triangle inequality. Additionally, we can verify:

$$\begin{aligned}
f(\text{ppk},\, x_2,\, c_2) &= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot \Gamma(c_2^T) \\
&= Q \cdot (S_1 \star x_2 P_1)^T + H \cdot (\Gamma(c_1^T) + c^T) \\
&= Q \cdot (S_1 \star x_1 P_1 - v)^T + H \cdot \Gamma(c_1^T) + H \cdot c^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T - Q \cdot v^T + H \cdot \Gamma(c_1^T) + H \cdot c^T \\
&= Q \cdot (S_1 \star x_1 P_1)^T + H \cdot \Gamma(c_1^T) \\
&= f(\text{ppk},\, x_1,\, c_1)
\end{aligned}$$

Thus confirming that the function values collide.

Theorem 3.7 Let $N,\ K,\ k,\ l_1,\ l_2,\ l$ be positive integers satisfying $l_1 < l_2 < l$ and h denote the entropy function. Suppose the inequality holds: $N - K - N \cdot h\left(\frac{4l+2l_2}{N}\right) > \lambda$. Additionally, assuming $DSD(N,\ K,\ l)$ or $DSD(N,\ K,\ l)$ and $extDSD(N,\ K,\ l-l_2,\ l+l_2)$ or $extDRSD(N,\ K,\ l-l_2,\ l+l_2)$ problems are intractable and the KKS scheme is one-time secure for parameters $l_1,\ l_2,\ Q,\ H$. Under these conditions, it can be concluded that the function f described in Definition 3.8 operates as a Chameleon hash function.

Proof. Following the procedure for solving trapdoor collisions detailed in Solution 8 and the proof given in Theorem 3.8, the property can be readily established.

3.9 *A novel two-tier signature framework*

Chameleon hash functions provide the necessary structure to enable the construction of a practical one-time two-tier signature scheme. While prior work has demonstrated that Chameleon hashing supports standard one-time signatures [6], this work presents a new two-tier signature framework, aimed at enhancing security and mitigating vulnerabilities in such schemes. As established in Section 3.2, we construct several instances of Chameleon hash functions, forming the basis for implementing one-time or d-time two-tier signature schemes.

We define a non-adaptively strongly secure one-time two-tier signature system denoted as $TTSig_f = (PriGen,\ SecGen,\ TTSign,\ TTVerify)$, where the construction is built upon a Chameleon hash function f. The relevant definitions for f, L_d, and $L_{[l_1,\ l_2]}$ can see Section 3.2. The input message x lies in the space $\mathbb{F}_{q^m}^k$.

1. $PriGen(1^\lambda)$: This step follows the key generation routine described in the KKS scheme from Section 3.1, where the primary public key is $ppk = (Q,\ H,\ A_{SIS})$ and the associated private key is denoted $psk = (G,\ J)$.

2. $SecGen(ppk,\ psk)$: A random value $\hat{c} \leftarrow L_d$ is selected, and the quantity $h = CHamHash(ppk;\ \hat{x};\ \hat{c})$ is evaluated for a representative public message $\hat{x} \in \mathbb{F}_{q^m}^k$. This output h becomes the secondary public key, i.e., $spk = h$, and the secondary secret key is given by $ssk = (\hat{c},\ P_1,\ P_2,\ S_1,\ S_2)$. (The function from Definition 3.1 is invoked here; similar constructions are valid.)

3. $TTSign(ppk,\ spk,\ x)$: The signer uses the trapdoor information $(G,\ J)$ and the private key $(P_1,\ P_2,\ S_1,\ S_2)$ to compute a collision value $c = Coll(psk,\ \hat{x},\ \hat{c},\ x)$. The resulting signature on the message x is c, where $c \in L_{[l-l_2,\ l+l_2]}$.

4. $TTVerify(ppk,\ spk,\ x,\ c)$: The verifier approves the signature if and only if the verification condition $f(ppk,\ x,\ c) = h$ is satisfied.

This one-time construction naturally extends to a d-time two-tier signature framework, see Definition 2.6.

4. Security analysis

In this section, we provide a rigorous security evaluation for the proposed schemes.

Theorem 4.1 If the new KKS scheme relies on computational intractability associated with SIS, and the public key matrix H and generator matrix G are constructed to meet security requirements (such as sparsity and noise matrix configurations), then the signature scheme satisfies the following security properties

- Resistance to forgery attacks,
- Resistance to Chosen Message Attacks (CMA),
- Quantum resistance.

Proof. Assume there exists a polynomial-time attacker $\mathscr{A}$ who can forge a signature $\sigma = (J,\ s_J,\ x)$ or perform a chosen message attack. We will show that these attacks are infeasible in the new KKS signature scheme by analyzing each attack model in detail.

Resistance to Forgery: Signature generation depends on solving the SIS problem, which involves solving the following equation:

$$A_{SIS} \cdot x_{SIS} = x \pmod{q},$$

where A_{SIS} is a matrix from the SIS problem, and x is the message digest. In the context of the KKS scheme, the SIS problem is formulated as finding short vectors that satisfy this equation, which is computationally hard. Specifically, finding a nontrivial solution to this equation is assumed to be intractable for classical computers and is believed to be equally hard for quantum computers (with some possible polynomial speedup for quantum algorithms).

The hardness of SIS is a key cryptographic assumption in lattice-based cryptography, with widely accepted evidence that the problem remains hard even for quantum adversaries. The quantum computer's power, while significant, does not provide an efficient way to solve lattice problems like SIS in high dimensions, making SIS a robust foundation for post-quantum security.

Therefore, any attacker attempting to forge a signature must solve a problem that is assumed to be computationally infeasible. Thus, the scheme is resistant to forgery attacks.

Resistance to Chosen Message Attacks (CMA): In a chosen message attack, an attacker selects a message m^* and attempts to generate a valid signature. The signature generation process depends on the message digest $x = H(m)$, where H is a cryptographically secure hash function (e.g., SHA-3 or BLAKE3). The security of the scheme depends on the collision resistance of the hash function, meaning that an attacker cannot generate two different messages m_1 and m_2 such that $H(m_1) = H(m_2)$.

Since $x = H(m)$, and given that H is collision-resistant, the attacker cannot generate a digest for any arbitrary message without the private key. This property ensures that the attacker cannot forge a signature for a message of their choice.

Additionally, the introduction of the noise matrix E during signature generation adds an additional layer of unpredictability. The noise matrix ensures that even if the attacker knows the digest x, they cannot reproduce the signature for any other message. This randomness makes it computationally infeasible for the attacker to predict or control the signature generation process, thus preventing the creation of valid signatures for chosen messages.

Quantum Resistance: The robustness of the new KKS signature scheme relies on the assumed intractability of the SIS problem. Quantum computers, while offering speedups for certain problems (e.g., through Grover's algorithm providing quadratic speedup for unstructured search), do not provide exponential speedup for the SIS problem. The best-known quantum algorithms for solving lattice-based task, including the Quantum Shortest Vector Problem (QSVP), do not exhibit polynomial or exponential speedup in the context of the SIS problem.

While there is ongoing research into quantum algorithms for lattice problems, no algorithm has yet been proven to solve the SIS problem efficiently in a quantum environment. Thus, the SIS problem remains computationally difficult for both classical and quantum platforms, ensuring robust post-quantum security guarantees.

Hence, leveraging the assumed post-quantum intractability of the SIS problem, the new KKS signature scheme is quantum-resistant.

Theorem 4.2 If $f(ppk,\ x,\ c)$ satisfies the properties of a $(t,\ \varepsilon)$-collision-resistant Chameleon hash function, then for every positive integer $q > 0$, the signature scheme $TTSig_f$ achieves $(t',\ q,\ 1,\ \varepsilon')$-TT-EUF-NCMA security, where $t' = t - O(q)$ and $\varepsilon' = \varepsilon - O(q)$.

Proof. Suppose there exists a PPT adversary $\mathscr{F}$ capable of compromising the TT-EUF-NCMA security of $TTSig_f$ under parameters $(t',\ q,\ 1,\ \varepsilon')$. Then we can construct another adversary $\mathscr{B}$ that violates the collision-resistance of the hash function f with complexity $(t,\ \varepsilon)$.

In particular, adversary $\mathscr{B}$ must generate two different input tuples $(x,\ c)$ and $(x',\ c')$ satisfying $x \neq x'$, satisfying

$$\mathsf{CHamHash}(chhk;\ x;\ c) = \mathsf{CHamHash}(chhk;\ x';\ c').$$

Initialization of $\mathscr{B}$: The challenger provides $\mathscr{B}$ with the public key $chhk$ for the Chameleon hash. $\mathscr{B}$ sets $ppk = chhk$ to simulate the primary public key in the signature protocol and forwards it to $\mathscr{F}$. Since the trapdoor psk is not known to $\mathscr{B}$, it cannot generate signatures directly and must rely on a signing oracle to emulate the signing environment.

Simulation of Signing Oracle: In order to answer the signature requests from $\mathscr{F}$, $\mathscr{B}$ defines a simulated signing oracle, denoted as NTTSign. Specifically, the oracle works as follows:

$$\text{NTTSign}(x_1,\ \dots,\ x_g):$$

- Increment the counter i by one.
- Generate secondary keys:

$$(spk_i,\ ssk_i) \overset{\$}{\longleftarrow} SecGen(ppk,\ psk).$$

- For each queried message x_j, where $j = 1,\ \dots,\ g$, generate signatures:

$$\sigma_j \overset{\$}{\longleftarrow} TTSign(psk,\ ssk_i,\ x_j).$$

- Store the queries $(x_1,\ \dots,\ x_g)$ in the list Q_i.
- Return $(spk_i,\ \sigma_1,\ \dots,\ \sigma_g)$.

Since $\mathscr{B}$ lacks the Chameleon hash trapdoor, it must simulate the signing process. For each queried message x_i:

- Randomly select x_i from the interval $L_{[l-l_2,\ l+l_2]}$.
- Compute:

$$h_i = CHamHash(ppk;\ x_i;\ c_i),$$

where c_i is randomly chosen from the valid range.

- Define the secondary public key as $spk_i = h_i$ and return both spk_i and the signature c_i to $\mathscr{F}$.

Indistinguishability of Simulation: In the real scheme, secondary public keys are computed as $spk_i = CHamHash(ppk;\ 0;\ r_i)$, whereas in the simulation, they are computed as $spk_i = CHamHash(ppk;\ \ x_i;\ c_i)$. Due to the uniformity of the Chameleon hash and the randomness of its parameters, these distributions are statistically indistinguishable. Thus, $\mathscr{F}$ cannot distinguish between the simulated and real environments.

Extracting the Collision: When $\mathscr{F}$ produces a forged tuple $(x^*,\ c^*,\ i^*)$, it must satisfy:

$$CHamHash(ppk;\ x_{i^*};\ c_{i^*}) = spk_{i^*} = CHamHash(ppk;\ x^*;\ c^*).$$

If $(x^*,\ c^*) \neq (x_{i^*},\ c_{i^*})$, then $\mathscr{B}$ outputs the collision:

$$\big((x_{i^*},\ c_{i^*}),\ (x^*,\ c^*)\big),$$

which successfully breaks the collision resistance of f.

Success Probability and Complexity Analysis:

- Success Probability: Since $\mathscr{F}$ is assumed to forge signatures with probability ε', $\mathscr{B}$ finds a collision in the Chameleon hash function with at least the same probability ε'. Therefore, we can conclude that $\varepsilon^{'} \leq \varepsilon - O(q)$.
- Time Complexity: The time complexity of $\mathscr{B}$ includes the overhead of simulating the signing oracle, which incurs an additional cost of $O(q)$ for q queries. Thus, $\mathscr{B}$'s total runtime is $t' = t - O(q)$.

Conclusion: If $\mathscr{F}$ successfully forges signatures with non-negligible probability ε', then $\mathscr{B}$ finds a collision in the Chameleon hash function f with the same non-negligible probability. This contradicts the $(t,\ \varepsilon)$-collision resistance of f. Therefore, $TTSig_f$ is $(t^{'},\ q,\ 1,\ \varepsilon^{'})$-TT-EUF-NCMA secure, where $\varepsilon' \leq \varepsilon - O(q)$ and $t' = t - O(q)$. This completes the proof.

Remark [Resistance to Practical Attacks] While the current manuscript primarily focuses on provable security under formal models (e.g., EUF-CMA, IND-CCA2), the scheme also exhibits inherent structural resilience against common practical attacks, as discussed below:

• **Resistance to Man-in-the-Middle (MitM) Attacks:** The non-interactive design of the two-tier signature scheme inherently mitigates MitM attacks. Signature generation and verification are unidirectional processes requiring no interactive key exchange, challenge-response protocol, or session negotiation. The signer computes the signature solely using the private key and message, while the verifier independently validates it using the public key and the message-signature pair. Since no communication channel is established between the parties, the adversary lacks an intermediate "relay point" to exploit. This architectural feature eliminates the classical vectors targeted in MitM attacks.

• **Resistance to Denial-of-Service (DoS)/Resource Depletion Attacks:** The scheme employs lightweight verification based on sparse matrix-vector multiplications over finite fields, with polynomial-time complexity. This guarantees that verification costs remain bounded and predictable. In practical implementations, further mitigation techniques such as rate-limiting, client whitelisting, and pre-validation of signature structure (e.g., format and length) can effectively reduce DoS threats.

• **Resilience Against Side-Channel Attacks (SCA):** Although the current manuscript does not provide a formal leakage-resilience proof, the structure of the scheme supports standard SCA countermeasures. Sensitive components in TTSign include:

- Trapdoor-based evaluation of the chameleon hash function $f(ppk,\ x,\ c)$,
- Randomized decoding of rank syndromes,
- Generation of the sparse noise matrix E and permutation seeds.

These operations can be implemented in constant time and benefit from regular linear algebra structures, allowing masking and blinding. In particular:

- Sparse matrix routines reduce conditional branching and promote uniform memory access patterns, limiting timing and power leakage.
- Hardware-based PRNGs further obscure secret-dependent behavior.

While a full empirical SCA evaluation is beyond the current scope, we acknowledge its importance and plan to investigate it in future hardware-oriented implementations.

5. Comparison of the proposed signature scheme with prior works

5.1 *Comparative assessment with established signature schemes*

To highlight the strengths of the developed two-level signature architecture, we perform a comparative analysis with representative post-quantum schemes, as illustrated in Table 2. In this evaluation, we select parameters such that the complexity of practical information set decoding remains no less than 2^{λ}, which is essential for preserving the soundness of the KKS assumption through the hardness of $DSD(N,\ K,\ \ell)$ and $extDSD(N,\ K,\ \ell-\ell_2,\ \ell+\ell_2)$ problems. Table 3 outlines the exact parameter configurations used.

According to Proposition 3.2 and Theorem 3.3, the value P provides the likelihood the signature weight deviates from the target interval $[\ell_1,\ \ell_2]$, and this probability should be sufficiently small to guarantee validity.

These post-quantum signature candidates exhibit varying levels of resistance to quantum threats. In general, their robustness grows exponentially with the bit-security parameter λ, aligning with the complexity of the most effective known decoding or algebraic attacks. Our assessment considers the underlying difficulty of a range of cryptographic problems, including the KKS assumption, SD, RSD, DRSD or SIS.

Table 2. Comparison of two-tier signature framework with existing schemes

The signature schemes	Field	Two-tier signature	Computational efficiency	Hardness of problems
KKS variants [21, 27]	$\mathbb{F}_q$ (commonly $q = 2$)	no	efficient	NP-hard
CFS methods [15]	$\mathbb{F}_q$ (mostly $q = 2$)	no	low practicality (decoding and hashing cost $\approx 2t!$)	NP-complete
HORS [28]	$\mathbb{F}_q$ (typically $q = 2$)	no	acceptable	NP-hard
Stern ID protocol [29]	$\mathbb{F}_q$ (often $q = 2$)	no	efficient	NP-complete
Cayrel-Véron-El Yousif [30]	$\mathbb{F}_q$ (where $q \neq 2$)	no	efficient	NP-complete
AGS [31]	$\mathbb{F}_2$	no	efficient	NP-complete
Proposed two-tier design	$\mathbb{F}_{q^m}$	yes	efficient	NP-complete

Table 3. Parameter settings in the designed signature

λ	N	K	n	k	l_1	l_2	l	P
128	48,948	24,474	892	43	396	496	893	$< 2^{-401}$
192	14,700	6,765	2,850	387	1,281	1,569	2,069	$< 2^{-856}$
256	19,600	9,020	3,800	518	1,706	2,094	2,894	$< 2^{-1,476}$

5.2 *Theoretical complexity analysis and comparison*

We present a detailed theoretical analysis of the computational and storage complexity of our scheme and compare it with several mainstream post-quantum schemes, including lattice-based and hash-based approaches.

5.2.1 *Computational complexity analysis*

The proposed two-tier signature scheme involves the following key computational procedures, each analyzed with asymptotic complexity based on parameter dimension n and matrix sparsity rates.

1. **Key Generation:**

• Constructing a sparse parity-check matrix $H \in \mathbb{F}_{q^m}^{(N-K)\times N}$ involves randomly sampling a sparse binary matrix with a given sparsity rate ρ_H, requiring approximately $\mathscr{O}(\rho_H \cdot N^2)$ operations.

• The generator matrix is formed as $G = G_1 + E$, where G_1 is a sparse structured matrix and E is sampled from a Gaussian distribution over $\mathbb{F}_{q^m}$. The overall complexity of this matrix generation and addition is $\mathscr{O}(\rho_{G_1} \cdot n^2)$.

• The verification matrix $Q = H_J G^T$ is computed via matrix multiplication. If both H_J and G are sparse, the complexity can be optimized to $\mathscr{O}(\rho_H \cdot \rho_G \cdot n^2)$.

2. **Signature Generation:**

• Hashing the message to obtain the digest $x = H(m)$ is assumed to take linear time $\mathscr{O}(n)$, assuming fixed-output cryptographic hash functions (e.g., SHAKE256).

• The encoding step $u = x \cdot G$ is a standard matrix-vector multiplication in $\mathbb{F}_{q^m}$, requiring $\mathscr{O}(n^2)$ operations.

• Solving the short integer solution (SIS) problem $A_{\text{SIS}} \cdot x_{\text{SIS}} = x \mod q$ is the core of the second-tier signature. While the worst-case complexity is exponential in the lattice dimension d, practical implementations using lattice basis reduction algorithms (e.g., BKZ, LLL) typically run in polynomial or subexponential time, depending on parameterization and target norm bounds.

• The partial signature vector s_J is constructed by mapping or extracting a subset of the global signature vector, which incurs at most $\mathscr{O}(n)$ operations.

3. **Signature Verification:**

• The full signature vector s is reconstructed, and the verification equation $H \cdot s^T = Q \cdot x^T$ is checked via matrix-vector multiplication, which has a computational cost of $\mathscr{O}(n^2)$.

• Additionally, the verifier checks whether $A_{\text{SIS}} \cdot x_{\text{SIS}} \equiv x (\text{mod } q)$, which is also a matrix-vector operation of similar complexity $\mathcal{O}(n^2)$.

All steps of the scheme, including key generation, signing, and verification, are bounded by polynomial-time complexity under standard parameter regimes. Moreover, the use of sparse matrices enables further optimization in real-world implementations, particularly in memory-constrained environments.

5.2.2 *Storage overhead analysis*

• **Public Key:** Includes the sparse matrix H, the verification matrix Q, and the SIS matrix A_{SIS}. Under parameters such as $n = 256$, the total size can reach approximately 6.8 MB. Note: The public key size corresponds to the uncompressed representation. In practical implementations, optimizations such as sparse matrix compression or seed-based key generation can significantly reduce the size.

• **Private Key:** Composed of the generator matrix G and subkey elements such as the index set J, typically of size $\mathcal{O}(n)$, much smaller than the public key.

• **Signature:** Includes the index set J, signature vector s_J, and digest x. For $n = 256$, the estimated signature size is around 5.8 KB.

5.3 *Comparison with mainstream PQC schemes*

Table 4. Comparison with mainstream post-quantum schemes

Scheme	Public key size	Signature size	Main operations and complexity
Proposed (KKS + Chameleon)	∼6.8 MB ($n = 256$)	∼5.8 KB	Matrix operations; SIS-based trapdoor with complexity $\mathcal{O}(n^3)$.
`Dilithium-III`	∼1.5 KB	∼2.7 KB	Lattice-based; NTT-accelerated polynomial multiplication $\mathcal{O}(n \log n)$.
`SPHINCS+-SHAKE`	∼1 KB	∼41 KB	Stateless hash-based scheme using Merkle trees and one-time signatures $\mathcal{O}(2^h)$.
`Rainbow-I`	∼0.2 MB	∼0.3 MB	Multivariate scheme; solving MQ systems via linearization $\mathcal{O}(n^3)$.

Observations in Table 4:

• The proposed scheme features a relatively small signature size, beneficial for low-bandwidth scenarios, at the cost of a large public key-typical in rank metric/code-based schemes [32].

• `Dilithium` achieves high efficiency due to NTT-based polynomial multiplication, and is currently a NIST standardized lattice-based signature scheme [33].

• `SPHINCS+` minimizes key size but incurs a significantly larger signature due to Merkle tree traversal and Winternitz one-time signatures, with overall time complexity roughly $\mathcal{O}(2^h)$, where h is the tree height [33].

• `Rainbow` offers compact signatures, but recent cryptanalysis has demonstrated full key-recovery attacks (notably in 2022-2023), leading to its exclusion from the NIST Round 4 standardization process [34].

5.4 *Security assumptions and extensibility*

The security of our scheme relies on the following well-studied hard problems:

• **Extended Rank Syndrome Decoding (extRSD):** Given a parity-check matrix H and syndrome y, it is hard to determine if y has a preimage of bounded rank weight. The best known attacks require time $\mathcal{O}(q^{(m-r)(n-r)})$, and parameters are chosen such that $(m-r)(n-r) \geq \lambda$ for a λ-bit security level.

- **Short Integer Solution (SIS):** Finding a short vector x such that $A_{\text{SIS}} \cdot x = 0 \mod q$, which remains hard even for quantum algorithms.

Furthermore, the two-tier design-separating master and ephemeral keys-improves flexibility and supports message reuse without compromising security. The use of chameleon hashing enhances non-transferability and enables designated-verifier functionality.

The proposed scheme offers the following theoretical advantages:

- **Compact signatures:** The signature size is considerably smaller than that of hash-based schemes such as SPHINCS+, which helps reduce communication and storage overhead.
- **Strong structure:** The two-tier key separation supports multi-message signing with enhanced non-repudiation guarantees.
- **Post-quantum security:** Based on extRSD and SIS, which are widely assumed to withstand attacks from quantum adversaries.

While the public key size is relatively large, this is a well-known trade-off in rank-metric cryptography. Techniques such as sparse matrix compression (e.g., CSR format), structured codes, and parameter tuning can reduce the overhead in future optimized implementations.

This theoretical analysis substantiates the efficiency and security claims of the proposed signature scheme and provides a sound response to the reviewer's concerns regarding the absence of empirical benchmarks.

5.4.1 *Performance comparison*

Table 5 compares the performance of the proposed two-tier signature scheme with the original KKS scheme [21], its improved version [27], and other representative post-quantum signature algorithms. Data for standard schemes such as Dilithium, Falcon, SPHINCS+, Rainbow, and McEliece is derived from the NIST PQC Round 3 submissions and reference implementations [33, 35].

Table 5. Comparison of signature schemes: efficiency and security

Scheme	Efficiency	Security
Original KKS scheme	Large key/signature sizes reduce efficiency	Potentially quantum-resistant
Improved KKS scheme	More efficient than original KKS	Post-quantum secure
Proposed two-tier scheme	Optimized for practical use with matrix sparsity	Post-quantum secure (based on extRSD and SIS)
Dilithium	Balanced performance with moderate key/signature sizes	Post-quantum secure
Falcon	High efficiency with small signatures	Post-quantum secure
SPHINCS +	Higher resource demand due to large signatures	Post-quantum secure
Rainbow	Fast verification but requires a very large public key	Broken by recent cryptanalysis [34]

Notes: Data for the original and improved KKS schemes is based on [21, 27]. Parameters for Dilithium, Falcon, SPHINCS+, Rainbow, and McEliece are from official NIST PQC submissions [33] and reference implementations [35].

5.4.2 *Performance optimization mechanisms and analysis*

Sparse Matrices and Efficient Representation

The proposed scheme exploits sparsity in both the parity-check matrix H and the generator matrix $G = G_1 + E$, where G_1 is a sparse binary matrix and E is a Gaussian noise matrix generated via a cryptographically secure pseudorandom number generator (e.g., SHAKE or AES-CTR). The sparsity rate $\rho_H \in \{0.1, 0.3\}$ enables reduction of the matrix operation complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(\rho_H \cdot N^2)$. Further optimization is achieved by storing matrices in Compressed Sparse Row (CSR) format, potentially reducing memory usage.

Post-Quantum Security Foundations

This scheme's security assurance is based on two fundamental constructs.

• **Full-rank sparse matrices:** The parity-check matrix H is iteratively verified to be full-rank, ensuring the robustness of the underlying rank-metric code even under sparsity constraints.

• **Noise-augmented generator construction:** The generator matrix $G = G_1 + E$ introduces sufficient randomness while enabling reduction to standard post-quantum hardness assumptions including hard computational problems like Short Integer Solution (SIS) and Learning With Errors (LWE), which are widely regarded as resistant to attacks by quantum-capable adversaries.

Potential Applications

Our two-tier signature scheme is designed to produce compact signatures and achieve non-transferability through chameleon hashing, making it particularly suitable for the following applications:

• **Blockchain and Smart Contract Signature Verification:**

The scheme's compact signature size is beneficial for low-bandwidth on-chain transactions and efficient signature verification in smart contracts. Its non-transferability property also aids in preventing unauthorized signature reuse.

• **Secure Digital Document and Messaging Systems:**

The compact and robust signature structure can be effectively employed in secure email, document signing, and digital rights management (DRM) applications to ensure data integrity and authenticity.

• **IoT/Edge Device Authentication and Identity Signing (Optional):**

Although the scheme features a relatively large public key, it can be selectively applied in IoT/edge scenarios where public key distribution is managed by a centralized or resource-rich server. In such cases, the compact signature size and efficient verification process are beneficial for device authentication and secure identity signing.

6. Conclusions

This paper presents a novel and tightly-secure two-tier signature scheme that integrates Chameleon hash functions with rank metric code-based cryptography. Through rigorous structural design and security reductions, the scheme ensures non-adaptive strong unforgeability under chosen message attacks. The dual-layer structure, reinforced by hard problems such as DSD and extDRSD, enhances resistance against algebraic and decoding-based attacks. Moreover, the formally established link between signature security and the collision-resistance property of the Chameleon hashing mechanism ensures sustained robustness.

Overall, this work contributes a quantum-resistant digital signature solution that balances theoretical rigor with implementation feasibility, making it well-suited for future cryptographic infrastructures in the post-quantum era.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 1978; 21(2): 120-126. Available from: https://doi.org/10.1145/359340.359342.

[2] Chaum D, Van Heijst E, Pfitzmann B. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: Feigenbaum J. (ed.) *Advances in Cryptology-CRYPTO'91*. Heidelberg: Springer; 1991. p.470-484. Available from: https://doi.org/10.1007/3-540-46766-1_38.

[3] Krawczyk H, Rabin T. *Chameleon Hashing and Signatures*. USA: Cryptology ePrint Archive; 1998.

[4] Ateniese G, De Medeiros B. Identity-based chameleon hash and applications. In: Juels A. (ed.) *Financial Cryptography*. Heidelberg: Springer; 2004. p.164-180. Available from: https://doi.org/10.1007/978-3-540-27809-2_19.

[5] Blazy O, Kakvi SA, Kiltz E, Pan J. Tightly-secure signatures from Chameleon hash functions. In: *Public-Key CryptographyPKC-2015*. Heidelberg: Springer; 2015. p.256-279. Available from: https://doi.org/10.1007/978-3-662-46447-2_12.

[6] Mohassel P. One-time signatures and Chameleon hash functions. In: Biryukov A, Gong G, Stinson DR. (eds.) *Selected Areas in Cryptography*. Heidelberg: Springer; 2011. Available from: https://doi.org/10.1007/978-3-642-19574-7_21.

[7] Debris-Alazard T, Sendrier N, Tillich JP. *Wave: A New Code-Based Signature Scheme*. France: HAL science; 2018.

[8] Gaborit P, Girault M. Lightweight code-based identification and signature. In: *2007 IEEE International Symposium on Information Theory*. Nice, France: IEEE; 2007. p.191-195. Available from: https://doi.org/10.1109/ISIT.2007.4557225.

[9] Hu J, Cheung RC. Toward practical code-based signature: Implementing fast and compact QC-LDGM signature scheme on embedded hardware. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2017; 64(8): 2086-2097. Available from: https://doi.org/10.1109/TCSI.2017.2684828.

[10] Lin Y, Xie Z, Chen T, Cheng X, Wen H. Image privacy protection scheme based on high-quality reconstruction DCT compression and nonlinear dynamics. *Expert Systems with Applications*. 2024; 257: 124891. Available from: https://doi.org/10.1016/j.eswa.2024.124891.

[11] Xie Z, Lin Y, Liu T, Wen H. Face privacy protection scheme by security-enhanced encryption structure and nonlinear dynamics. *iScience*. 2024; 27(9): 110768. Available from: https://doi.org/10.1016/j.isci.2024.110768.

[12] Mao G, Chen D, Li G, Dai W, Sanka AI, Koç ÇK, et al. High-performance and configurable SW/HW co-design of post-quantum signature CRYSTALS-Dilithium. *ACM Transactions on Reconfigurable Technology and Systems*. 2023; 16(3): 1-28. Available from: https://doi.org/10.1145/3569456.

[13] Bellare M, Shoup V. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In: Okamoto T, Wang X. (eds.) *Public Key Cryptography-PKC 2007*. Heidelberg: Springer; 2007. p.201-216. Available from: https://doi.org/10.1007/978-3-540-71677-8_14.

[14] Wang Y, Ismail ES. Tightly-secure two-tier signatures on code-based digital signatures with chameleon hash functions. *Mathematics*. 2024; 12(15): 2375. Available from: https://doi.org/10.3390/math12152375.

[15] Courtois NT, Finiasz M, Sendrier N. How to achieve a McEliece-based digital signature scheme. In: Boyd C. (ed.) *Advances in Cryptology-ASIACRYPT 2001*. Heidelberg: Springer; 2001. p.157-174. Available from: https://doi.org/10.1007/3-540-45682-1_10.

[16] Gaborit P, Zémor G. On the hardness of the decoding and the minimum distance problems for rank codes. *IEEE Transactions on Information Theory*. 2016; 62(12): 7245-7252. Available from: https://doi.org/10.1109/TIT.2016.2616127.

[17] Otmani A, Tillich JP. An efficient attack on all concrete KKS proposals. In: Yang BY. (ed.) *Post-Quantum Cryptography*. Heidelberg: Springer; 2011. p.98-116. Available from: https://doi.org/10.1007/978-3-642-25405-5_7.

[18] Hofheinz D, Jager T. Tightly secure signatures and public-key encryption. In: Safavi-Naini R, Canetti R. (eds.) *Advances in Cryptology-CRYPTO 2012*. Heidelberg: Springer; 2012. p.590-607. Available from: https://doi.org/10.1007/978-3-642-32009-5_35.

[19] Lenstra AK, Lenstra HW, Lovász L. Factoring polynomials with rational coefficients. *Mathematische Annalen*. 1982; 261: 515-534. Available from: https://doi.org/10.1007/BF01457454.

[20] Schnorr CP, Euchner M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*. 1994; 66: 181-199. Available from: https://doi.org/10.1007/BF01581144.

[21] Kabatianskii G, Krouk E, Smeets B. A digital signature scheme based on random error-correcting codes. In: *Crytography and Coding*. Heidelberg: Springer; 1997. p.161-167. Available from: https://doi.org/10.1007/BFb0024461.

[22] Kipnis A, Patarin J, Goubin L. Unbalanced oil and vinegar signature schemes. In: Stern J. (ed.) *Advances in Cryptology-EUROCRYPT '99*. Heidelberg: Springer; 1999. p.206-222. Available from: https://doi.org/10.1007/3-540-48910-X_15.

[23] Patarin J. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: *Advances in Cryptology-EUROCRYPT '96*. Heidelberg: Springer; 1996. p.33-48. Available from: https://doi.org/10.1007/3-540-68339-9_4.

[24] Patarin J, Courtois N, Goubin L. FLASH, a fast multivariate signature algorithm. In: Naccache D. (ed.) *Topics in Cryptology-CT-RSA 2001*. Heidelberg: Springer; 2001. p.298-307. Available from: https://doi.org/10.1007/3-540-45353-9_22.

[25] Berlekamp ER. Factoring polynomials over finite fields. *Bell System Technical Journal*. 1967; 46(8): 1853-1859. Available from: https://doi.org/10.1002/j.1538-7305.1967.tb03174.x.

[26] Cantor DG, Zassenhaus H. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*. 1981; 36(154): 587-592. Available from: https://doi.org/10.2307/2007663.

[27] Cayrel PL, Otmani A, Vergnaud D. On Kabatianskii-Krouk-Smeets signatures. In: Carlet C, Sunar B. (eds.) *Arithmetic of Finite Fields*. Heidelberg: Springer; 2007. p.237-251. Available from: https://doi.org/10.1007/978-3-540-73074-3_18.

[28] Reyzin L, Reyzin N. Better than BiBa: Short one-time signatures with fast signing and verifying. In: Batten L, Seberry J. (eds.) *Information Security and Privacy* . Heidelberg: Springer; 2002. p.144-153. Available from: https://doi.org/10.1007/3-540-45450-0_11.

[29] Stern J. A new identification scheme based on syndrome decoding. In: Stinson DR. (ed.) *Advances in Cryptology-CRYPTO' 93*. Heidelberg: Springer; 1994. p.13-21. Available from: https://doi.org/10.1007/3-540-48329-2_2.

[30] Cayrel PL, Véron P, El Yousfi Alaoui SM. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: Biryukov A, Gong G, Stinson DR. (eds.) *Selected Areas in Cryptography*. Heidelberg: Springer; 2010. p.171-186. Available from: https://doi.org/10.1007/978-3-642-19574-7_12.

[31] Aguilar C, Gaborit P, Schrek J. A new zero-knowledge code based identification scheme with reduced communication. In: *2011 IEEE Information Theory Workshop*. Paraty, Brazil: IEEE; 2011. p.648-652. Available from: https://doi.org/10.1109/ITW.2011.6089577.

[32] Debris-Alazard T, Sendrier N, Tillich JP. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In: Galbraith S, Moriai S. (eds.) *Advances in Cryptology-ASIACRYPT 2019*. Heidelberg: Springer; 2019. p. 21-51. Available from: https://doi.org/10.1007/978-3-030-34578-5_2.

[33] National Institute of Standards and Technology. *Post-Quantum Cryptography Standardization*. 2023. Available from: https://csrc.nist.gov/projects/post-quantum-cryptography [Accessed 25th December 2024].

[34] Beullens W. Breaking rainbow takes a weekend on a laptop. In: Dodis Y, Shrimpton T. (eds.) *Advances in Cryptology-CRYPTO 2022*. Heidelberg: Springer; 2022. p.464-479. Available from: https://doi.org/10.1007/978-3-031-15979-4_16.

[35] Alagic G, Apon D, Cooper D, Dang Q, Dang T, Kelsey J, et al. *Status Report on the Third Round of the Nist Post-Quantum Cryptography Standardization Process (NIST IR 8413)*. USA: National Institute of Standards and Technology; 2022. Available from: https://doi.org/10.6028/NIST.IR.8413.

Appendix A

Description of the trapdoor collision algorithm

```
Algorithm 2: Trapdoor Collision Algorithm (Coll)
    Input: Private key psk with trapdoor matrices $(S_1, P_1)$, original pair $(x_1, c_1)$, target message $x_2$
    Output: Collision value $c_2$ such that $f(\mathsf{ppk}, x_1, c_1) = f(\mathsf{ppk}, x_2, c_2)$
    Step 1:    Compute the difference vector

                   $$v \leftarrow S_1 \star x_1 P_1 - S_1 \star x_2 P_1.$$

    Step 2:    Solve the linear system

                   $$Q \cdot v^\top = H \cdot c^\top,$$

    with solution $c$ satisfying $\|c\| \in [l_1, l_2]$. This is typically solved via the KKS signature process.
    Step 3:    Set $c_2 \leftarrow c$ and return it.
    Step 4:    Return $c_2$
```

Appendix B

Theorem 3.1 security reductions

We let λ be the security parameter. We say that an algorithm (or advantage) is *polynomial-time* (resp. *negligible*) in λ if it scales as $\mathrm{poly}(\lambda)$ (resp. $1/\mathrm{poly}(\lambda)$), where $\mathrm{poly}(\lambda)$ denotes a polynomial function in λ, such as λ^c for some constant c.

Reduction to RSD/extDRSD

We recall that our improved KKS-based scheme involves a parity-check matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$, which induces a rank-metric code with large minimum distance. We show that any successful forgery or chameleon-hash collision can be turned into an RSD/extDRSD solver.

Theorem B.1 [Reduction to RSD/extDRSD] Suppose there is a PPT adversary $\mathscr{A}$ that forges a valid signature (or produces a valid chameleon-hash collision) in our scheme with non-negligible probability ε. Then there is a PPT algorithm $\mathscr{B}$ solving an RSD/extDRSD instance with probability at least $\varepsilon/\mathrm{poly}(\lambda)$, contradicting the hardness of RSD/extDRSD.

Proof. Step 1 (*RSD Instance Input*) Algorithm $\mathscr{B}$ is given $(H,\, y)$, with $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and a target $y \in \mathbb{F}_{q^m}^{(n-k)}$. By assumption, there exists a low-rank x^* such that $H \cdot x^{*T} = y^T$.

Step 2 (*Public Key Embedding*) $\mathscr{B}$ constructs the public key $\mathsf{ppk} = (H,\, Q,\, A_{\mathrm{SIS}},\, \ldots)$ by embedding H directly. The other components $(Q,\, P_1,\, P_2,\, S_1,\, S_2)$ are generated according to the scheme. Thus, any real signature/collision must satisfy an equation involving H.

Step 3 (*Simulating Signatures*) When $\mathscr{A}$ queries a (one-time) signature on a message m, $\mathscr{B}$ simulates the signing via standard KKS steps:

- Compute $x = H(m)$ (the hash of m).
- Generate codeword $u = x \cdot G$ and solve the SIS part for x_{SIS} if needed.
- Output a signature $\sigma = (J,\, s_J,\, x)$ with $s_J = u + x_{\mathrm{SIS}}$.

The simulation is statistically or computationally indistinguishable from a real environment.

Step 4 (*Forgery/Collision Extraction*) Eventually, $\mathscr{A}$ outputs either a forged signature $\sigma^* = (J^*,\, s_J^*,\, x^*)$ for a new message or a chameleon-hash collision $(x_1,\, c_1) \neq (x_2,\, c_2)$. In both scenarios, we obtain an equation of the form

$$H \cdot s^{*T} = Q \cdot x^{*T}, \quad \text{or} \quad f(\mathsf{ppk},\, x_1,\, c_1) = f(\mathsf{ppk},\, x_2,\, c_2).$$

Due to the trapdoor structure, $\mathscr{B}$ can extract an auxiliary vector c^* satisfying

$$H \cdot c^{*T} = y^T$$

with $\mathrm{rank}(c^*) \leq w$ (or $\mathrm{rank}(c^*)$ in $[r_{\min},\, r_{\max}]$ for the extDRSD case). Hence, c^* solves the given instance.

Step 5 (*Advantage Analysis*) If $\mathscr{A}$ succeeds with probability ε, $\mathscr{B}$ finds c^* with probability at least $\varepsilon/\mathrm{poly}(\lambda)$. This contradicts the assumed hardness of RSD/extDRSD, completing the Proof. □

Reduction to SIS

Next, we show that the SIS-based component of our scheme (namely the short-vector solution required in the signing procedure) yields a reduction to the SIS problem.

Theorem B.2 [Reduction to SIS] If a PPT adversary $\mathscr{A}$ forges a signature in our scheme (without knowledge of the SIS trapdoor) with non-negligible probability ε, then there is a PPT algorithm $\mathscr{B}$ solving the SIS instance $(A_{\mathrm{SIS}},\, b)$ with probability at least $\varepsilon/\mathrm{poly}(\lambda)$, contradicting SIS hardness.

Proof. **Step 1** (*SIS Instance*) $\mathscr{B}$ is given $(A_{\text{SIS}},\ b)$, where $A_{\text{SIS}} \in \mathbb{Z}_q^{n\times m}$ and the goal is to find z with $\|z\| \leq \beta$ s.t. $A_{\text{SIS}} \cdot z \equiv b \pmod q$.

Step 2 (*Public Key*) $\mathscr{B}$ embeds A_{SIS} into the scheme's public key. Other matrices $(H,\ Q,\ \ldots)$ are generated as in the real system.

Step 3 (*Oracle Simulation*) For each message m that $\mathscr{A}$ requests a signature on, $\mathscr{B}$ solves the SIS part using the real trapdoor (or direct knowledge) to produce a valid short vector x_{SIS}, ensuring $\|x_{\text{SIS}}\| \leq l_2$. Thus, σ is valid and indistinguishable from a real one.

Step 4 (*Forgery Extraction*) If $\mathscr{A}$ forges a new signature σ^*, it necessarily contains a short vector $\tilde{x}_{\text{SIS}}$ that satisfies

$$A_{\text{SIS}} \cdot \tilde{x}_{\text{SIS}} \equiv x^* \pmod q, \quad \|\tilde{x}_{\text{SIS}}\| \leq l_2.$$

With some additional manipulation (possibly re-centering $\tilde{x}_{\text{SIS}}$), $\mathscr{B}$ converts $\tilde{x}_{\text{SIS}}$ into a solution z for $(A_{\text{SIS}},\ b)$.

Step 5 (*Probability*) Hence, if $\mathscr{A}$ wins with probability ε, so does $\mathscr{B}$ up to a polynomial factor in λ, contradicting SIS hardness. □

Overall security implications

Our improved KKS-based signature and associated chameleon hash scheme rely on both the rank-metric code structure (tied to RSD/extDRSD) and the short integer solution approach (tied to SIS). If either reduction holds, we derive a contradiction to the presumed hardness of those underlying assumptions. Consequently, to break the scheme, an adversary would need to solve either:

1. A rank-based decoding problem (RSD/extDRSD), or
2. A short integer solution problem (SIS),

both of which are conjectured to be infeasible for PPT adversaries at appropriate parameter sizes.

As a result, our construction achieves security under the combined assumptions of rank-based and lattice-based hardness, offering resilience in a post-quantum setting.

Conclusion

We have demonstrated how any successful forgery or hash collision against our scheme can be transformed into an algorithm solving either RSD/extDRSD or SIS. The reductions involve constructing a simulated public key, responding to signing queries via either rank-metric trapdoors or SIS trapdoors, and then extracting a valid solution from the adversary's final forgery. In each case, the probability of finding a solution to the underlying hard problem is at least $\varepsilon/\text{poly}(\lambda)$. This shows that both the signature scheme and the chameleon hash function maintain *collision-resistance*, *trapdoor collision*, and *uniformity* under standard assumptions in rank-based and lattice-based cryptography.

Appendix C
Hybrid reduction for Theorem 3.8

In this section, we present a refined and more rigorous version of the hybrid reduction proof for Theorem 3.4, showing how any collision attack against our chameleon hash function

$$f(ppk,\, x,\, c) \;=\; Q \cdot (S_1 \star x P_1)^T \;+\; H \cdot (S_2 \star c P_2)^T$$

implies an efficient algorithm to solve either the SIS or RSD (resp. extDRSD) problem, thus contradicting their presumed hardness.

Preliminaries and setup

Notation

• Let λ be the security parameter, and poly(λ) represent an unspecified polynomial in λ.

• We write $\|\cdot\|$ for a norm bound (in the SIS context, typically the ℓ_2 norm or ℓ_∞ norm, depending on the scheme), and rank$(\cdot)$ for the rank of a vector over an extension field in the rank-metric context.

• $A_{\mathrm{SIS}} \in \mathbb{Z}_q^{n\times m}$ is the matrix used in the SIS problem. The goal there is to find a short non-zero vector z such that $A_{\mathrm{SIS}} \cdot z \equiv 0 \pmod q$ (or equals some specific $b \pmod q$) if required).

• $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ is the parity-check matrix used in the rank-based code, involved in an RSD instance: find x with rank$(x) \le w$ s.t. $H \cdot x^T = y^T$.

Statement of the hybrid reduction

Theorem C.1 [Hybrid Reduction to SIS and RSD] Let

$$f(ppk,\, x,\, c) \;=\; Q \cdot (S_1 \star x P_1)^T \;+\; H \cdot (S_2 \star c P_2)^T$$

be the chameleon hash function constructed in Section 3.1 (and extended in Definition 3.2-3.8). Suppose there exists a PPT adversary $\mathscr{A}$ that, with non-negligible probability ε, outputs two distinct pairs $(x_1,\, c_1)$ and $(x_2,\, c_2)$ with $x_1 \neq x_2$ such that

$$f(ppk,\, x_1,\, c_1) = f(ppk,\, x_2,\, c_2).$$

Then there exists a PPT algorithm $\mathscr{B}$ that, with probability at least $\varepsilon/\mathrm{poly}(\lambda)$, solves either:

• An instance of the SIS problem (*short integer solution*), or

• An instance of the RSD (or extDRSD) problem (*rank syndrome decoding*),

contradicting the hardness assumptions of SIS and RSD.

Proof of Theorem 3.2

Proof. We describe how to construct $\mathscr{B}$ in a hybrid manner, depending on properties of the *difference vector* $\Delta c :=$ $(c_2 - c_1)$.

Step 1 Collision Equation By the collision condition, we have:

$$Q\cdot(S_1\star x_1P_1)^T+H\cdot(S_2\star c_1P_2)^T \;=\; Q\cdot(S_1\star x_2P_1)^T+H\cdot(S_2\star c_2P_2)^T.$$

Rearrange to isolate differences:

$$Q\cdot\big(S_1\star(x_1-x_2)P_1\big)^T \;=\; H\cdot\big(S_2\star(c_2-c_1)P_2\big)^T.$$

Define $v:=(x_1-x_2)$ and $\Delta c:=(c_2-c_1)$. Thus,

$$Q\cdot(S_1\star vP_1)^T \;=\; H\cdot(S_2\star\Delta cP_2)^T.$$

Note that $v\neq 0$ since $x_1\neq x_2$.

Step 2 Embedding SIS We assume the scheme sets H in a form that can be *decomposed* as

$$H \;=\; A_{\text{SIS}}\,R,$$

where R is an invertible matrix over the relevant field/ring, and A_{SIS} is a SIS matrix. (This approach is common in lattice-based or mixed code-lattice-based constructions, where H can be embedded into an SIS context.)

We rewrite the collision equation:

$$Q\cdot(S_1\star vP_1)^T \;=\; A_{\text{SIS}}\cdot\big[R\cdot(S_2\star\Delta cP_2)^T\big].$$

If

$$z \;:=\; R\cdot(S_2\star\Delta cP_2)^T$$

has sufficiently small norm (e.g., $||z||\leq\beta$ under some ℓ_2 or ℓ_∞ metric), then we obtain a short, nonzero z satisfying

$$A_{\text{SIS}}\cdot z \;=\; Q\cdot(S_1\star vP_1)^T.$$

If $Q\cdot(S_1\star vP_1)^T$ is either 0 or some known vector b mod q, one can transform it to the homogeneous form $A_{\text{SIS}}\cdot z\equiv 0$ (mod q) or b accordingly. Thus z solves the SIS instance.

Hence, whenever $||\Delta c||$ is *small enough* such that $||z||\leq\beta$, we derive an SIS solution.

Step 3 RSD Path if Δc is Not Short Now, if Δc (and thus z) is *not* guaranteed to be short, the SIS path fails. In that scenario, $\mathscr{B}$ switches to a rank-decoding approach:

$$Q\cdot(S_1\star vP_1)^T \;=\; H\cdot(S_2\star\Delta cP_2)^T.$$

One can interpret $(S_2 \star \Delta c P_2)$ as a vector in $\mathbb{F}^n_{q^m}$; let us denote this vector by $\widetilde{c} \in \mathbb{F}^n_{q^m}$. Then we have

$$H \cdot \widetilde{c}^T \; = \; Q \cdot (S_1 \star v P_1)^T.$$

Using the trapdoor from the rank-metric KKS scheme, we can extract an $\widetilde{x}$ with bounded rank such that $H \cdot \widetilde{x}^T = Q \cdot (S_1 \star v P_1)^T$. This effectively yields an RSD (or extDRSD) solution, because $\widetilde{x}$ must have rank in a limited range (e.g. $[l_1,\, l_2]$ or $\leq w$), consistent with how the KKS signature scheme handles codewords.

In other words, if Δc is large in norm (so SIS-based bounding does not hold), it typically implies $\widetilde{c}$ remains within a rank-limited set (due to the scheme's parameter constraints and the structure of $c_1,\, c_2$ in L_d or $L_{[l_1,\, l_2]}$). Hence, $\mathscr{B}$ obtains an $\widetilde{x}$ that solves $H \cdot \widetilde{x}^T = y^T$ for $y := Q \cdot (S_1 \star v P_1)^T$, which is precisely an RSD instance.

Step 4 Combined Hybrid Conclusion Algorithm $\mathscr{B}$ runs $\mathscr{A}$:

- On input $(A_{\text{SIS}},\, H)$, it embeds H as $H = A_{\text{SIS}} \cdot R$ if SIS is the main target, or keeps H as is (with rank constraints) if RSD is the main target.
- $\mathscr{A}$ returns $(x_1,\, c_1)$, $(x_2,\, c_2)$ forming a collision.
- $\mathscr{B}$ checks the norm of $\Delta c = c_2 - c_1$. If $\|\Delta c\| \leq$ (threshold), $\mathscr{B}$ forms the short vector z for SIS. Otherwise, it proceeds to extract an RSD solution from the rank structure.

In either case, $\mathscr{B}$ solves SIS or RSD with probability at least $\varepsilon/\text{poly}(\lambda)$, where the polynomial factor accounts for repeated attempts, negligible distinguishing errors, or other simulation overhead.

Concluding Remarks Thus, the existence of a PPT collision-finder $\mathscr{A}$ for f directly contradicts the hardness of SIS and RSD (or extDRSD). Consequently, under these standard assumptions, no PPT adversary can produce a valid collision in our chameleon hash function with non-negligible advantage. □

Remark C.1 [On Parameter Choices and Norm/Rank Constraints] In practice, the scheme enforces a norm bound (e.g. $\|\Delta c\| \leq l_2$) or a rank bound ($\text{rank}(\Delta c) \leq w$) within certain intervals (e.g. $l_1 < \|\Delta c\| < l_2$) to ensure that *one* of the two cases (SIS or RSD) always applies. This is typically achieved by carefully choosing $(n,\, m,\, k)$ for the rank-metric and $(q,\, \beta)$ for the SIS dimension and modulus, ensuring an either-or scenario arises, thus guaranteeing the soundness of this hybrid reduction.

Remark C.2 [Advantage Loss and $\text{poly}(\lambda)$ Factor] As is standard in cryptographic reductions, whenever we pass from an adversary's success probability ε to the solver's success probability, we may lose a polynomial factor. This can arise from repeating the simulation multiple times or from minor distinguishing errors in the embedding steps. Hence the solver's success is at least $\varepsilon/\text{poly}(\lambda)$, preserving non-negligibility if ε was non-negligible.