https://doi.org/10.0000/tmsce.2026.001

A Review of Code-Based Identification and Signature Schemes

Sapna Jyoti Patel

Department of Mathematics, National Institute of Technology Jamshedpur, India, 831014
Corresponding author: 2023rsma001@nitjsr.ac.in

Abstract. Code-based cryptography has emerged as a strong candidate for post-quantum security, owing to its solid mathematical foundations and resilience against quantum attacks. This paper presents a systematic review of identification and signature schemes developed within the code-based framework. Beginning with the seminal constructions of Stern and Véron, we trace the evolution of zero-knowledge identification protocols through their key refinements, including the CVE, AGS, and LESS schemes. We further examine hash-and-decrypt-based signature constructions, such as CFS and Wave, which extend code-based techniques toward practical, efficient digital signatures. By reviewing these developments chronologically, the survey reveals the principal trends, methodological innovations, and enduring challenges that shape the field. The aim is to provide a cohesive understanding of the progress achieved so far and to highlight why code-based methods remain strong contenders in the landscape of post-quantum digital signatures.

Key words: Code-based cryptography; Zero-knowledge proofs; Identification schemes; Digital signatures; Post-quantum cryptography
Received: 25 December 2025 **Revised:** 24 January 2026 **Accepted:** 27 January 2026

1. Introduction

Digital signatures [23, 25, 8, 10] are cryptographic mechanisms that enable recipients to verify the authenticity of a message using straightforward algorithms. They ensure the authenticity of the signer, the integrity of the message, and non-repudiation. Typically, a digital signature scheme requires two keys: a *private key* (also known as the *signing key* or *secret key*) belonging to the signer, and a *verification key* (or *public key*). The design of a digital signature scheme ensures that the signature for any given message is intricately linked to both the message and the signer through their respective keys.

The emergence of digital signatures parallels the rise of computers and digital communication. Initially, these signatures were created using classical public-key cryptosystems, such as RSA [30]. These digital signatures relied heavily on the difficulty of solving number-theoretic problems, including the prime factorisation and discrete logarithm problems. However, over time, Peter Shor [33, 34] developed an algorithm that efficiently solves any classical number-theoretic problem on quantum computers with sufficient computational resources. This development presented a substantial threat to existing cryptographic protocols.

Given the continually increasing computational capabilities of quantum computers, there is a pressing need to develop quantum-resistant cryptographic protocols. Fortunately, NIST has identified five primary candidates for post-quantum cryptography, which are listed below.

- **Hash-based cryptography** depends on the hardness of inverting hash functions, and signatures like Sphincs [3] have been designed using this underlying problem.
- **Lattice-based cryptography** depends on the hardness of solving some problems based on a lattice structure, like *shortest integer solution (SIS)* and *shortest vector problem (SVP)*. Signatures like Crystals-dilithium [19] have been developed using these.
- **Multivariate** signatures like Rainbow [18] have been developed considering the difficulty in solving systems of multivariate polynomials.
- **Isogeny-based** signatures like SQIsign [16] use maps between two elliptic curves, and the difficulty in finding such maps to design signatures.
- **Code-based cryptography** relies on some hard problems based on codes like *syndrome decoding problem (SDP)* as their foundations.

Among the candidates for post-quantum cryptography, code-based cryptography has a history dating back to the 1970s (see [29, 31, 35]). The McEliece cryptosystem [27], developed around the same time, has uniquely withstood quantum attacks since its inception in 1978, unlike RSA. Additionally, most code-based protocols involve introductory matrix algebra, making the underlying structure and mathematics relatively easy to understand. This quality positions code-based cryptography as a promising candidate for post-quantum systems.

Code-based cryptography offers flexibility to change the underlying code if a structural attack is found against the code used in a cryptosystem or a signature. A notable example is the McEliece cryptosystem and its equivalent, the Niederreiter cryptosystem [28], which were both designed using syndrome decoding (SDP) on any code as their foundation. However, these systems have faced attacks based on underlying structures like Reed-Solomon or Reed-Muller codes. By switching to Goppa codes, developers have made them resilient to quantum attacks, ensuring their security for over 45 years. Despite this, designing code-based signatures has remained a complex yet compelling challenge for cryptographers worldwide. This complexity arises because the standard hash-and-decrypt paradigm is not straightforward when creating a signature scheme in a code-based environment. Nevertheless, numerous attempts have been made to develop secure and efficient signature solutions.

This paper presents a condensed study of existing code-based signatures, highlighting their strengths and weaknesses. We begin with a detailed timeline in Section 2. Next, in Section 3, we describe commonly used terms, definitions, concepts, and hard problems to aid understanding of the signature schemes. The subsequent sections discuss the code-based signatures and related protocols developed to date. Finally, we conclude the paper with a comparative analysis.

2. History and Motivation

The development of code-based identification and signature schemes dates back to the 1990s, with zero-knowledge protocols like those developed by Stern [36] and Véron [37] being the pioneers. Another approach, the hash-and-decrypt kind of signatures, came in 2003 after Courtois et al. [14] developed a signature scheme. However, because these do not meet the modern-day thresholds for security, there were some developments in the area of code-based identification and signature schemes.

In 2018, the Wave signature scheme [17, 5] was developed using a new concept called generalized $(U, U + V)$ codes. This scheme also falls under the category of hash-and-decrypt signature schemes in code-based cryptography. Unlike the CFS scheme, the authors of [5] proposed that the decoded vector should have a sufficiently large weight to facilitate efficient

decoding. This feature made the Wave scheme suitable for practical implementation, enabling it to enter the NIST PQC Round 1 (Additional Digital Signatures) standardization process.

The Linear Equivalence Signature Scheme (LESS) developed in 2020 by Biasse et al. [9] is a cryptographic scheme based on the linear equivalence problem (LEP). Over time, this scheme has undergone refinements and performance improvements, leading to its inclusion in Round 2 (Additional Digital Signatures) of the NIST PQC standardization process [4, 15].

Recent efforts to design code-based identification protocols with increasingly lower soundness errors reached new milestones when Gueron et al. [24] introduced a protocol that uses codes over large finite fields (with $q > 100$) while achieving the desired soundness error. However, due to the substantial computational resources required for implementation, this protocol was not submitted to the NIST PQC team for standardization.

Despite significant advancements in code-based cryptography that have marked important milestones, code-based signature schemes still face challenges due to their large key sizes. The substantial memory and storage requirements hinder the practical application of these schemes in the current era, where computational power is often limited. This situation has motivated researchers worldwide to develop code-based signature schemes with enhanced performance or smaller signature and key sizes. A chronological examination of previous research reveals trends and methodological innovations that, in turn, inspired the systematic literature review presented in this manuscript.

3. Preliminaries

The following notations are adopted throughout this paper.

Notation	What it represents
$\mathbb{F}_q$	field of order q
$\mathbf{A}$	a matrix $\mathbf{A}$
$\mathbf{a}$	a vector $\mathbf{a}$
λ	security parameter
$\leftarrow$	assignment operator (value on the right is assigned to the variable on the left)

Additionally, the following concepts are introduced for clarity.

Definition 1 (Negligible Function [22])**.** *A function $f(x) : \mathbb{N} \to \mathbb{R}$ is said to be negligible in terms of x, if for every value a, there exist some $N_a \in \mathbb{N}$ such that for every $x \geq N_a$, $|f(x)| \leq \frac{1}{x^a}$.*

Definition 2 (Computational Difficulty or Hardness [22])**.** *An algorithm or process $\mathcal{A}$ is said to be computationally hard if the probability of success of the algorithm $\mathcal{A}$ is negligible in terms of the number of attempts or runs of the algorithm $\mathcal{A}$.*

This means that, for a computationally hard problem, finding an output requires many attempts and re-runs of the algorithm to obtain one solution of the problem. In this paper, we have described some computationally hard problems used to design code-based digital signature schemes.

3.1. Hash Functions

A hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^l$ is a mathematical function that takes an arbitrary-length bit string as input and gives a fixed-length bit string as output (called the *digest*) such that:

1. For any two bit strings $w_1 \in \{0,1\}^*$ and $w_2 \in \{0,1\}^*$, it only occurs negligible number of times that $\mathcal{H}(w_1)$ has the same value as $\mathcal{H}(w_2)$.
2. For any given bit string x, it is easy to find a $\mathbf{x} = \mathcal{H}(x)$ such that $\mathbf{x} \in \{0,1\}^l, l \in \mathbb{N}$.
3. For any random $\mathbf{y} \in \{0,1\}^l$, it is computationally hard to find a preimage $y \in \{0,1\}^*$ such that $\mathcal{H}(y) = \mathbf{y}$.
4. Given a digest $\mathbf{z}$ and one of its preimage $z_1 \in \{0,1\}^*$, it is computationally hard to find another bit string $z_2 \in \{0,1\}^*$ such that $\mathcal{H}(z_2) = \mathbf{z}$.

3.2. Background of Coding Theory

A linear $[n,k,d]_q$-code $\mathcal{C}$ is defined as a subset of $(\mathbb{F}_q)^n$ that has dimension k. For this code, there exists a full rank matrix known as the *parity-check matrix*, denoted as $\mathbf{H}$, of size $(n-k) \times n$. This matrix satisfies the condition that for any codeword $\mathbf{c} \in \mathcal{C}$, the equation $\mathbf{cH}^T = 0$ holds. The elements of $\mathcal{C}$ are referred to as *codewords*. These codewords are generated by applying a k to n linear transformation to a full-rank vector space $M \subseteq (\mathbb{F}_q)^k$ of dimension k, utilizing a matrix called the *generator matrix*, denoted as $\mathbf{G}$. Specifically, for any vector $\mathbf{m} \in M$, the codeword is obtained by the operation $\mathbf{mG} = \mathbf{c} \in \mathcal{C}$.

Throughout this paper, $\mathbf{H}$ will represent a parity-check matrix, and $\mathbf{G}$ will denote a generator matrix for a code over a field $\mathbb{F}_q$, with the value of q specified as needed.

A codeword $\mathbf{c} = (c_1, c_2, ..., c_n) \in \mathcal{C}$ is said to have a *Hamming weight* w if the number of non-zero components c_i (where $1 \leq i \leq n$) in $\mathbf{c}$ is exactly w.

Given a parity-check matrix $\mathbf{H}$ of a linear code, and considering any vector $\mathbf{y} \in (\mathbb{F}_q)^n$, the vector $\mathbf{s} = \mathbf{yH}^T$ is termed the *syndrome* of $\mathbf{y}$ with respect to $\mathbf{H}$.

An isometry is a mapping $\rho : \mathcal{C}_1 \to \mathcal{C}_2$ that consists of a permutation π of n objects and a vector $\mathbf{v} \in (\mathbb{F}_q)^n$. The mapping is defined such that $\rho(\mathbf{c}) = \pi(\mathbf{c}) \otimes \mathbf{v}$, where $\mathbf{a} \otimes \mathbf{b}$ results in a vector whose components are the modular products of the corresponding components of $\mathbf{a}$ and $\mathbf{b}$. Two codes are considered *isometric* if such an isometry exists between them.

3.3. Hard Problems in Coding Theory

Definition 3 (Syndrome Decoding Problem (SDP))**.** *Given a q-ary parity-check matrix $\mathbf{H}$, a q-ary vector $\mathbf{s} \in (\mathbb{F}_q)^{(n-k)}$, and an integer w, the goal is to find a vector $\mathbf{v} \in (\mathbb{F}_q)^n$ of weight w such that $\mathbf{vH}^T = \mathbf{s}$.*

Setting $q = 2$ here makes this problem a *binary SDP*. This problem is known to be computationally challenging, as established in [2] and [23].

Definition 4 (General Decoding Problem (GDP))**.** *Given a q-ary generator matrix $\mathbf{G}$, a vector $\mathbf{y} \in (\mathbb{F}_q)^n$, and an integer w, the goal is to find two vectors, $\mathbf{m} \in (\mathbb{F}_q)^k$ and $\mathbf{e} \in (\mathbb{F}_q)^n$, such that the weight of $\mathbf{e}$ is w (denoted $wt(\mathbf{e}) = w$) and $\mathbf{xG} + \mathbf{e} = \mathbf{y}$.*

As shown in [37], this problem can be reduced to the Syndrome Decoding Problem (SDP), indicating that it is equally hard.

Definition 5 (Linear Equivalence Problem (LEP))**.** *Given two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with their respective generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$, the task is to find a $k \times k$ non-singular matrix $\mathbf{S} \in GL_k(q)$ and an $n \times n$* monomial *matrix $\mathbf{Q} = \mathbf{DP}$ (where $\mathbf{D}$ is a diagonal matrix and $\mathbf{P}$ is a permutation matrix) such that $\mathbf{G}_2 = \mathbf{SG}_1\mathbf{Q}$.*

This problem is discussed in detail in [9]. These problems form the foundation of code-based cryptography. All encryption and digital signature schemes in code-based cryptography are constructed using one or more of these problems.

Definition 6 (Decoding One Out of Many (DOOM) Problem)**.** *Given a set of syndromes* $S = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{|S|}\}$ *with* $\mathbf{s}_i \in (\mathbb{F}_q)^{n-k}$*, it is intended to find a vector* $\mathbf{e} \in (\mathbb{F}_q)^n$ *such that* $\mathbf{eH}^T = \mathbf{s}_i$ *for some* $\mathbf{s}_i \in S$*.*

This problem is discussed in depth and used in [17, 5]. It is noteworthy that this problem is believed to be hard, and there is no concrete evidence of its hardness.

3.4. Zero-Knowledge-Proof (ZKP)-based Identification Schemes

The concept of zero-knowledge proof schemes originates from the fundamental question of how a prover, denoted as $\mathcal{P}$, can convincingly demonstrate to a verifier, represented as $\mathcal{V}$, that they possess a secret $\mathbf{x}$ without revealing any information about the secret itself. The underlying idea is straightforward: $\mathcal{V}$ challenges $\mathcal{P}$ to provide responses that convince $\mathcal{V}$ of $\mathcal{P}$'s knowledge of $\mathbf{x}$. A zero-knowledge proof scheme generally follows this structure:

- In the **Setup** phase, public parameters are established, which both parties agree upon.
- The **Key Generation** algorithm assists the prover, $\mathcal{P}$, in generating public and private keys. The private key typically consists of the secret that $\mathcal{P}$ aims to prove possession of, while the public key serves as auxiliary information that helps $\mathcal{V}$ verify $\mathcal{P}$'s claim.
- At this stage, $\mathcal{P}$ creates a commitment, denoted as `cmt`, using the **Commitment** algorithm, which is then sent to $\mathcal{V}$.
- The verifier, $\mathcal{V}$, challenges the claim of the prover, $\mathcal{P}$, during the **Challenge** stage. The corresponding challenge, `ch`, is sent to the prover.
- In response, $\mathcal{P}$ provides a value, `rsp`, which depends on both `cmt` and `ch`.
- Using `cmt`, `ch`, and `rsp`, along with the public key, the verifier outputs either 1 (indicating "accept") or 0 (indicating "reject").

In some instances, the verifier may pose multiple challenges that the prover needs to respond to, resulting in variations in the number of interactions (known as passes) in the identification scheme. A three-pass zero-knowledge proof scheme is commonly referred to as a *Sigma protocol*. Additionally, there is a possibility that a dishonest prover, denoted as $\mathcal{F}$, who does not know $\mathbf{x}$, could successfully convince the verifier, $\mathcal{V}$, of their possession of the secret. The probability of such an event is termed the *cheating probability* or *soundness error* of the zero-knowledge identification scheme.

A ZKP protocol possesses two additional properties:

- *Completeness*: An honest prover will always succeed in proving their authenticity by convincing the verifier that they possess $\mathbf{x}$.
- *Soundness*: A false prover can be misidentified as an authentic prover with only a small probability.

These properties make ZKP schemes an ideal choice for designing identification protocols and digital signatures. An interactive ZKP protocol can be transformed into a non-interactive signature scheme for a message, `msg`, using the *Fiat-Shamir transformation* (see [21] and [32]), which operates as follows:

1. Run the **Commitment** algorithm $r(\in \mathbb{N})$ times to generate the commitments $\texttt{cmt}_1$, $\texttt{cmt}_2$, ..., $\texttt{cmt}_r$. Let $\texttt{CMT} = (\texttt{cmt}_1 \,||\texttt{cmt}_2||\ldots||\texttt{cmt}_r)$.
2. Using a hash function $\mathcal{HASH}$, calculate the digest d of $(\texttt{msg}||\texttt{CMT})$.

3. Using d as the value of `ch`, generate the corresponding response `RSP`.

4. The signature on `msg` will be (`CMT`||`RSP`).

The value of r depends on the desired level of security.

3.5. The Hash-and-Decrypt Approach to Obtain Digital Signature Schemes

An effective methodology for designing a secure and efficient digital signature scheme is the Hash-and-Decrypt approach. This method requires a hash function, denoted as $\mathcal{H}ash$, which produces a digest of an appropriate length. The signing process for a message, referred to as `msg`, is outlined below:

1. The message `msg` is hashed to obtain a value α.

2. The digest α is treated as the encrypted form of a plaintext message within the context of a public-key cryptosystem. Here, the public key, K_{pub}, is published as the verification key, while the private key, K_{pr}, is kept secret by the signer and serves as the signing key.

3. Using the private key, and under the assumption that α is the encrypted form of some plaintext, the signer finds a value β such that $\beta = Dec_{K_{pr}}(\alpha)$. In this case, α is the encrypted form of β. Encryption is performed with the publicly known key K_{pub}, which is straightforward, whereas decryption is computationally challenging and is efficiently performed using the trapdoor provided by K_{pr}.

4. The value of β is then published as the signature for the message `msg`.

This approach was initially introduced in [14] and later refined in [17]. When a verifier receives the pair $(\texttt{msg}, \beta)$, they need to verify that $Enc_{K_{pub}}(\beta)$ matches $\mathcal{H}ash(\texttt{msg})$. It's important to note that hash functions are one-way functions: the digest (the output of the hash) can be computed efficiently, but it is computationally difficult to find the original input given the digest.

4. Code-Based Identification Schemes

As mentioned in the previous section, zero-knowledge protocols are an effective approach for researchers developing signature schemes. In the realm of code-based cryptography, several such protocols have been proposed. The following subsections present them in chronological order.

4.1. Véron's Zero-Knowledge Identification Scheme

In 1997, Véron [37] proposed a zero-knowledge identification scheme analogous to Stern's protocol, leveraging the difficulty of the general decoding problem. Unlike Stern's scheme, which employs a parity-check matrix, Véron's protocol utilizes a generator matrix. The details of Véron's three-pass zero-knowledge identification scheme are illustrated in Figure 1.

Since the general decoding problem and the Syndrome Decoding Problem (SDP) can be polynomially reduced to each other, the security of Véron's protocol remains comparable to that of Stern's protocol. Additionally, in this protocol, a dishonest prover $\mathcal{F}$ can deceive a verifier into producing an "accept" or 1 in the verification phase approximately (2/3) of the time using the following strategies:

- $\mathcal{F}$ can employ $\mathbf{m}' \in (\mathbb{F}_2)^k$ and $\mathbf{e}' \in (\mathbb{F}_2)^n$ with a weight $wt(\mathbf{e}') = w$ instead of $\mathbf{e}$. In this case, $\mathbf{m}'\mathbf{G} \oplus \mathbf{e}'$ does not need to equal $\mathbf{x}$, and the prover hopes that the challenge $\mathtt{ch}$ is either 1 or 2.
- $\mathcal{F}$ can also use $(\mathbf{m}'', \mathbf{e}'')$ such that $\mathbf{m}''\mathbf{G} \oplus \mathbf{e}'' = \mathbf{x}$ while the weight of $\mathbf{e}''$ does not necessarily equal w. In this scenario, the prover succeeds if $\mathtt{ch}$ is either 0 or 2.

Thus, similar to Stern's protocol, Véron's protocol exhibits a soundness error of (2/3), necessitating the same number of rounds as in Stern's case to achieve a desired level of security, approximately 137 rounds for 80-bit security and 219 rounds for 128-bit security.

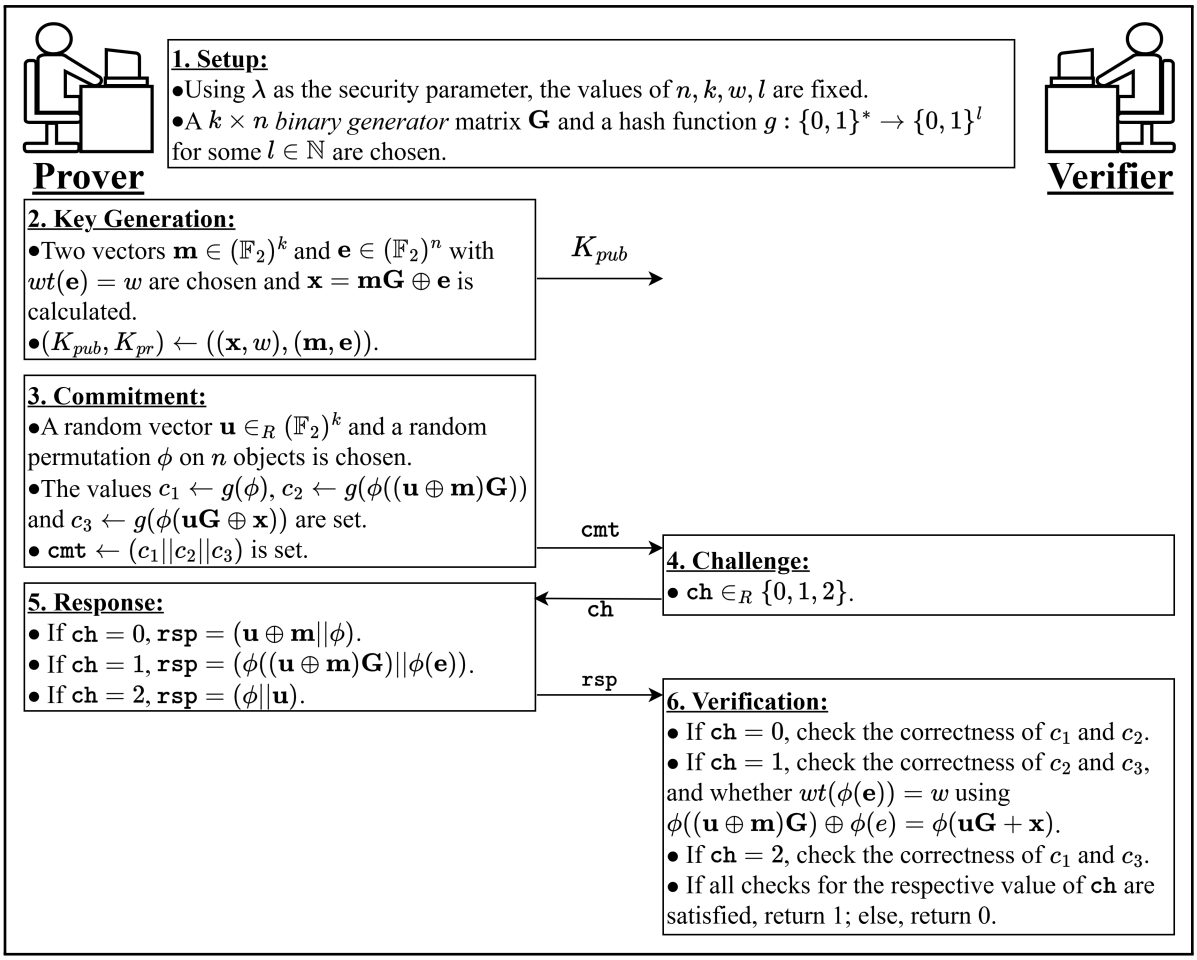

Figure 1: Véron's [37] Zero-Knowledge Identification Scheme

4.2. CVE Scheme

In 2010, Cayrel, Véron, and El Yousfi Alaoui [12] proposed an identification scheme based on the q-ary syndrome-decoding problem. Unlike previous schemes that had a soundness error of $\frac{2}{3}$, the CVE scheme offers a soundness error slightly greater than $\frac{1}{2}$. Notably, the zero-knowledge protocol they introduced is a 5-pass protocol, as illustrated in Figure 2.

Cayrel et al. recommended using $q = 2^m$ where $m \in \mathbb{N}$ and $m > 1$. In their paper, they demonstrated that the cheating probability, which corresponds to the soundness error, of their proposed scheme is $\frac{q}{2(q-1)}$. As m increases, this value approaches $\frac{1}{2}$, allowing the protocol to achieve a desired level of security with fewer rounds compared to existing schemes.

However, designing q-ary codes for such values of q presents a challenge. Efficient decoding often requires algebraic structures that attackers can exploit, such as polynomial-based constructions. If an attacker successfully implements a structural attack on the underlying code, the entire signature scheme may become insecure and unsuitable for practical use. A structural attack on the underlying code may render the signature scheme insecure and unfit for practical use.

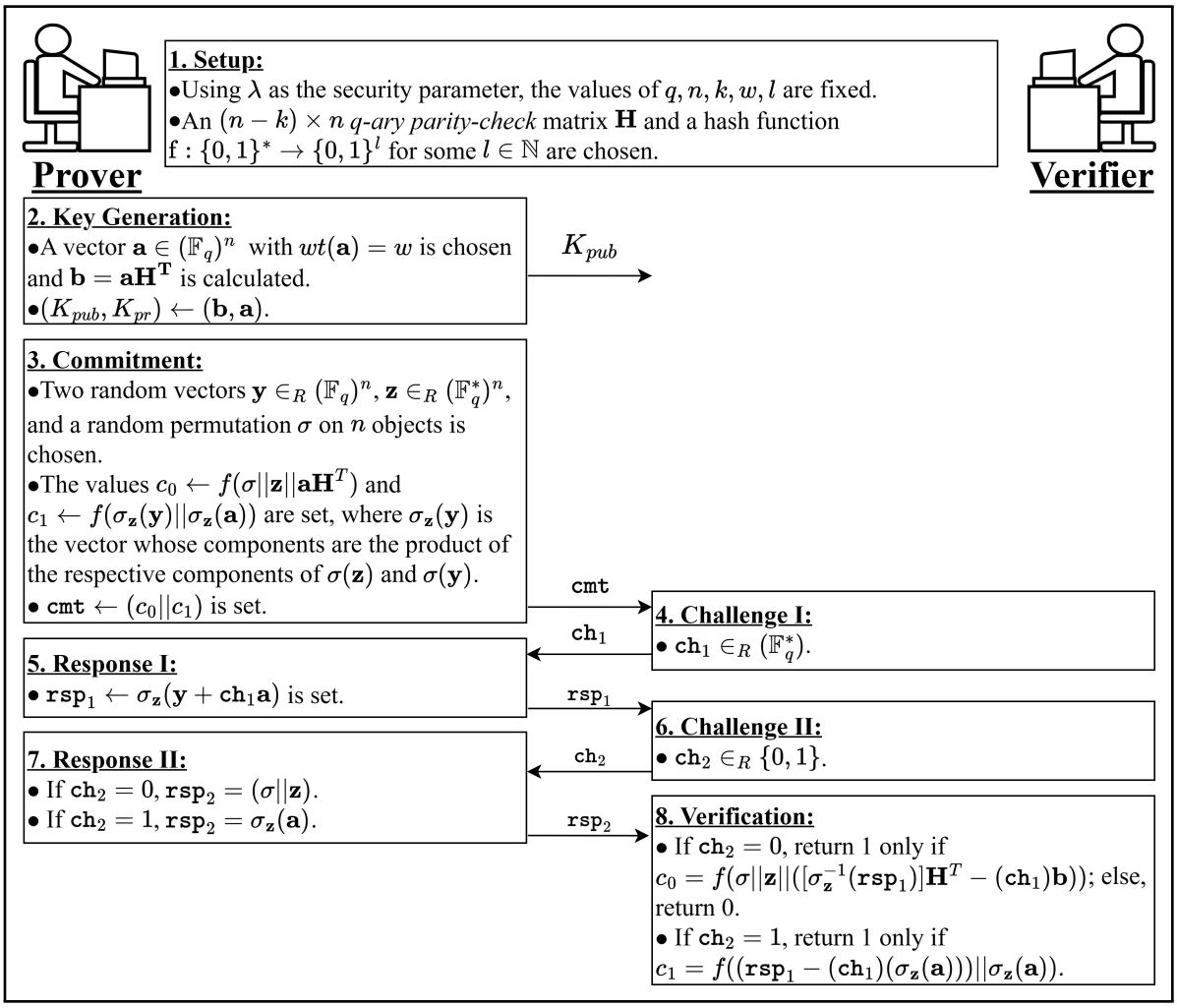

Figure 2: The CVE Identification Scheme [12]

4.3. AGS Scheme

In 2011, Aguilar, Gaborit, and Schrek [1] introduced a 5-pass identification scheme that is a variant of Stern's authentication protocol. They proposed using *double circulant codes* and *cyclic shifts* in the protocol. Additionally, by employing a hash function, they asymptotically reduced the soundness error to 1/2. This advancement led to a decrease in the number of rounds needed to achieve λ bits of security compared to Stern's or Véron's schemes. A comprehensive description of the AGS scheme, along with all the related algorithms, is provided in Figure 3.

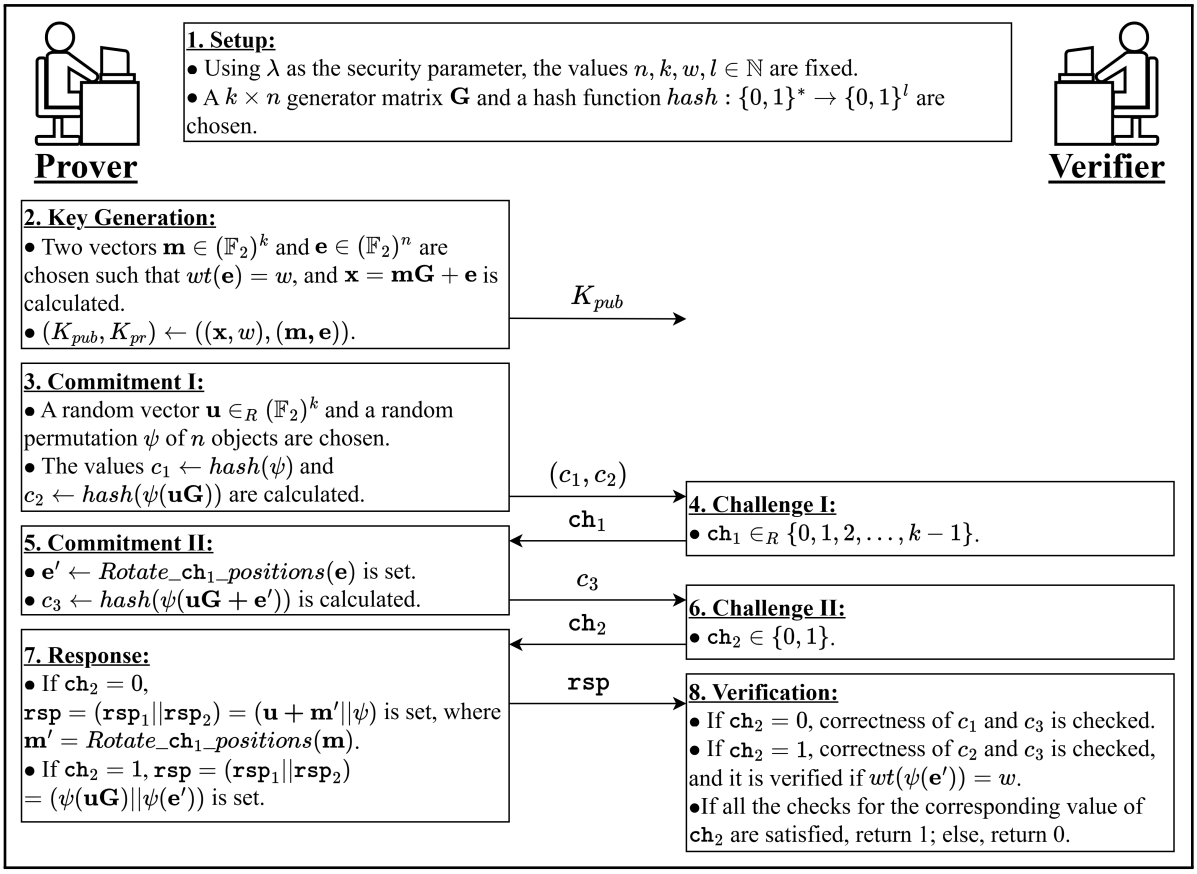

Figure 3: The AGS Identification Scheme [1]

4.4. LESS Scheme

Unlike existing schemes that rely heavily on the Syndrome Decoding Problem (SDP) or related problems for security, Biasse et al. [9] used the hardness of the linear code equivalence problem to design a zero-knowledge identification scheme. This scheme has a soundness error of exactly $\frac{1}{2}$, which means that for λ bits of security, exactly λ rounds of the protocol must be followed. However, the signature size remained too large for practical applications. In 2025, Chou et al. [13] introduced a variant of the LESS scheme that reduced the signature size. Later, Beckwith et al. [7] developed an optimized scheme that achieved even smaller signature sizes and faster signing and verification times. Additionally, Baldi et al. [4] positioned LESS as a candidate for NIST's Round 2 in Post-Quantum Cryptography [15]. The original scheme, as developed by Biasse et al., is illustrated in Figure 4.

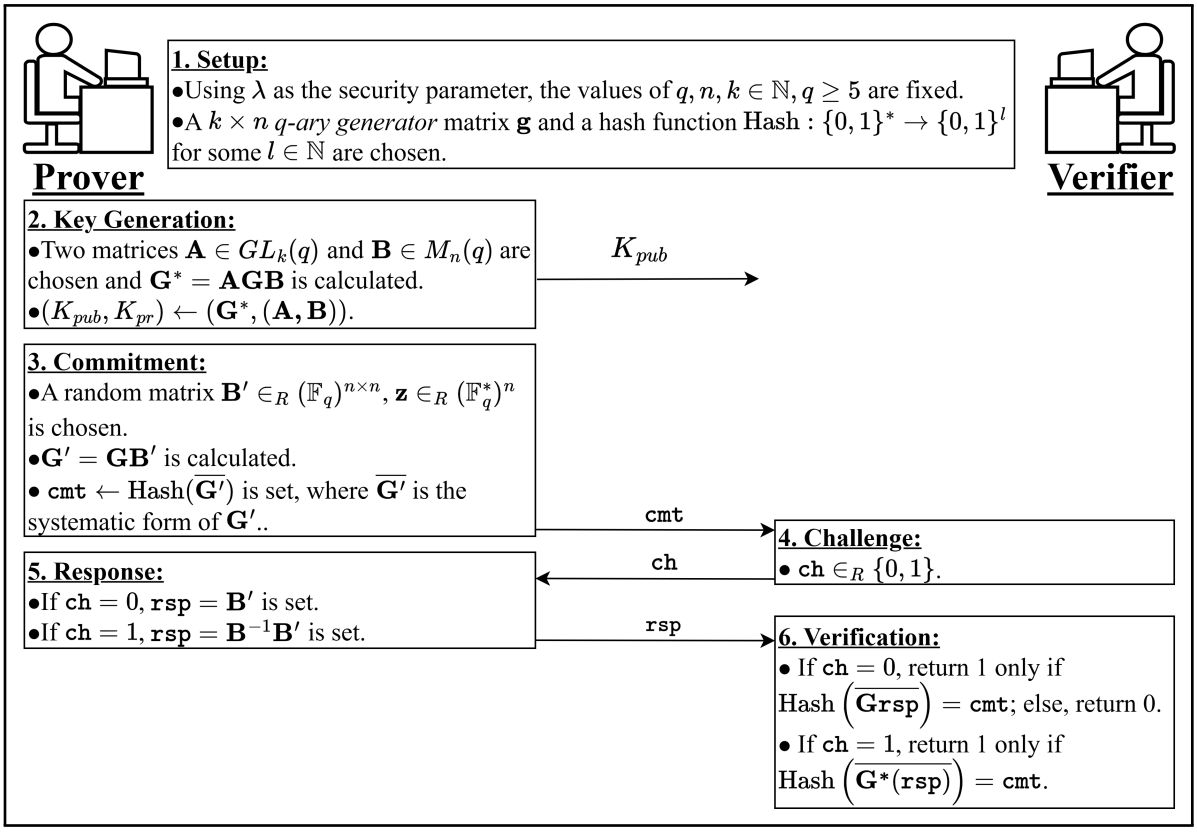

Figure 4: LESS [9, 7] Identification Scheme

4.5. GPS Zero-Knowledge Identification Scheme with a Helper

Existing schemes typically achieve a soundness error of only 1/2 asymptotically. To address this limitation, Gueron, Persichetti, and Santini [24] developed an identification scheme capable of achieving a significantly lower soundness error. The key innovation was designing the signature scheme over fields of larger order (with a higher value of q). Additionally, a helper is required to distribute a seed to the prover and an auxiliary value aux corresponding to the seed to the verifier. Figure 5 illustrates this version with a helper.

In [24], modifications were proposed that eliminate the need for a helper. In this revised scheme, the prover selects N different seeds, denoted as $\mathtt{seed}^{(i)}$, and performs the **Setup** and **Commitment** steps for each seed:

- $\mathtt{aux} = (\mathtt{aux}^{(1)}||\mathtt{aux}^{(2)}||...||\mathtt{aux}^{(N)})$, where $\mathtt{aux}^{(i)}$ is the respective auxiliary value for $\mathtt{seed}^{(i)}$; and
- $\mathtt{cmt} = (\mathtt{cmt}^{(1)}||\mathtt{cmt}^{(2)}||...||\mathtt{cmt}^{(N)})$, where $\mathtt{cmt}^{(i)}$ is the output of the **Commitment** phase corresponding to $\mathtt{seed}^{(i)}$.
- Also, in this case, the challenge algorithm produces $\mathtt{ch} = (I||\mathtt{ch}^{(I)})$, where $I \in_R \{0, 1, 2, ..., N\}$ and $\mathtt{ch}^{(I)} \in \mathbb{F}_q$ is the respective output of the **Challenge** phase of Figure 5 corresponding to the index I.

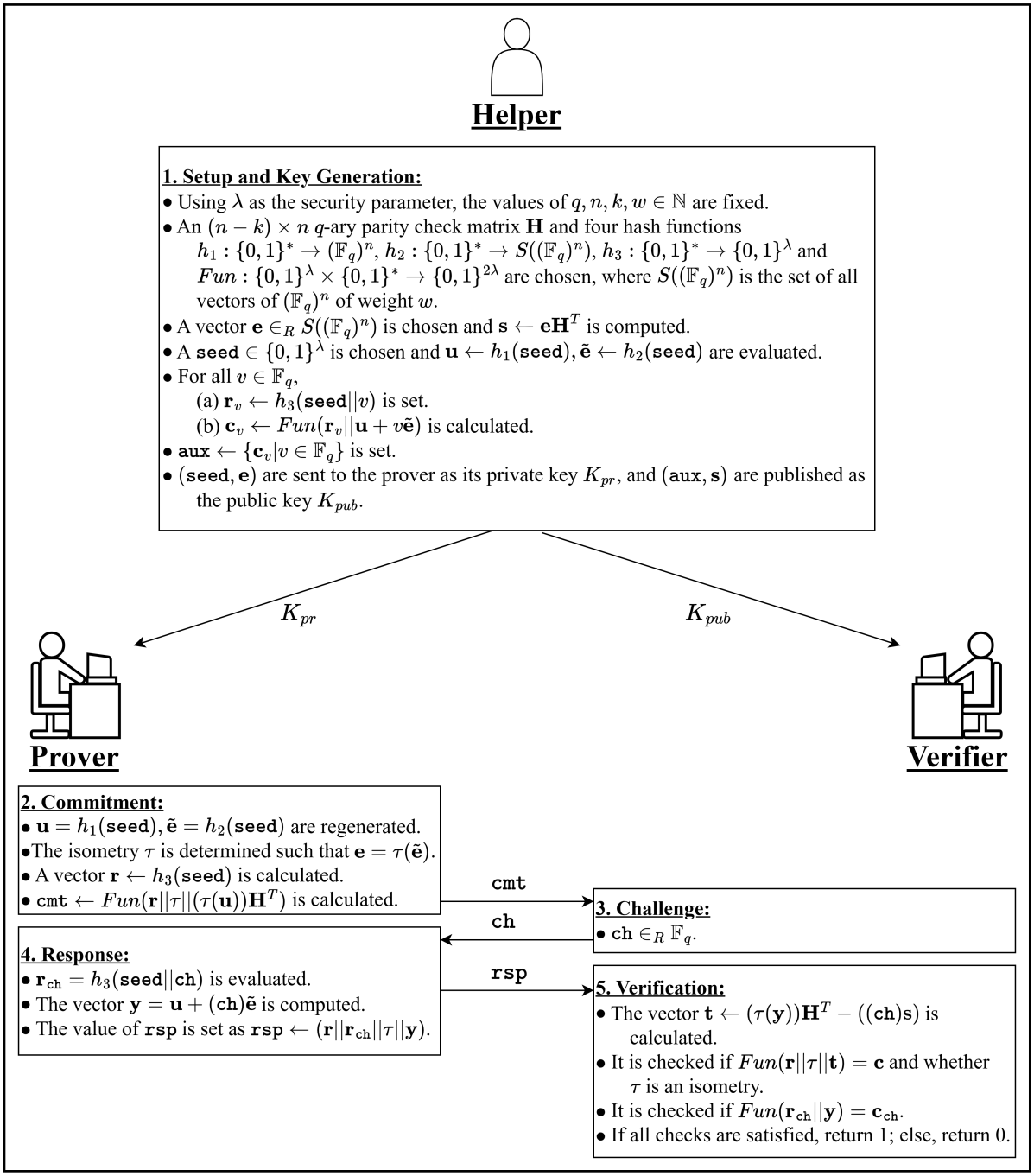

Figure 5: Identification scheme with helper as in [24]

The process must be repeated a specific number of times using the Fiat-Shamir transformation to create a non-interactive signature scheme. A dishonest prover might attempt to find a collision by utilizing N different indices I or q different values $\mathtt{ch}^{(i)} \in \mathbb{F}_q$, as noted in [24]. Consequently, the soundness error in this scheme is given by $\max\left(\frac{1}{q}, \frac{1}{N}\right)$. By increasing the values of q and N, this error can be made arbitrarily small, resulting in a scheme that surpasses existing alternatives. However, a notable limitation arises from the complexity of designing efficient codes over large finite fields; operations like matrix multiplication in this context lead to increased computational complexity and higher memory requirements. The authors mitigate this issue by carefully selecting the parameters to make the implementation

more feasible (see [24]).

5. Code-Based Hash-and-Decrypt Signature Algorithms

An alternative approach to authenticating a signer while ensuring message integrity is the hash-and-decrypt method. In this approach, a message $\texttt{msg}$ is first hashed using any publicly known hash function, denoted as $Hash$, to produce a digest $\mathbf{m}$. This hash output $\mathbf{m}$ is then treated as an encrypted message in a public-key cryptosystem, where decryption yields the signature σ. In these signatures, the encryption key is made publicly available, while the decryption key is kept secret by the signer. Consequently, any verifier can encrypt the signature σ to obtain $\mathbf{m}$ and check if it matches the value of $Hash(\texttt{msg})$. The RSA signature scheme is an example of a digital signature scheme based on the RSA cryptosystem. However, both the McEliece and Niederreiter cryptosystems, while public-key, do not provide signature schemes of this nature. The reason for this limitation is that not all hashed values can be decoded within code-based frameworks. Despite these challenges, two notable signature constructions using this method have been proposed, which are explained below.

5.1. CFS Signature Protocol

In 2003, Courtois, Finiasz, and Sendrier [14] pioneered the development of the first code-based hash-and-decrypt digital signature scheme. They proposed using t-error-correcting Goppa codes as the foundation for this scheme. With some modifications, they achieved a signature size of approximately 144 bits, marking a significant advancement in code-based cryptography.

To begin, the involved parties must agree on the values of n, k, and t, along with an $(n-k) \times n$ parity-check matrix $\mathbf{H}$ of a binary Goppa code. Additionally, a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{(n-k)}$ needs to be established between the signing party and the verifying party. The signing process and verification algorithm are outlined in Algorithms 1 and 2, respectively.

Algorithm 1 CFS signing algorithm

Input: $\mathbf{H}$, $\mathcal{H} : \{0,1\}^* \to \{0,1\}^{(n-k)}$, $\texttt{msg}$
Output: Signature σ on $\texttt{msg}$

$\mathbf{m} \leftarrow \mathcal{H}(\texttt{msg})$
$i \leftarrow 0$
$\mathbf{s}_0 \leftarrow \mathbf{m}$
while $\mathbf{s}_i$ is not decodable **do**
    $i \leftarrow i + 1$
    $\mathbf{s}_i \leftarrow \mathcal{H}(\mathbf{m}||i)$
end while
$\mathbf{z} \leftarrow Dec_{\mathbf{H}}(\mathbf{s}_i)$ [where $Dec_{\mathbf{H}}(\mathbf{s}_i)$ decodes $\mathbf{s}_i$ w.r.t. $\mathbf{H}$ to give $\mathbf{z}$ with $wt(\mathbf{z}) = t$ and whose non-zero positions of $\mathbf{z}$ are $p_1, p_2, ..., p_t$]
$I(\mathbf{z}) \leftarrow 1 +^{p_1} C_1 +^{p_2} C_2 + ... +^{p_t} C_t$
return $\sigma = (I(\mathbf{z})||i)$

Algorithm 2 CFS verification algorithm

Input: $\mathbf{H}$, h, $(\texttt{msg}, \sigma)$
Output: $b \in \{0,1\}$

$\left(\overline{I(\mathbf{z})},\ \bar{i}\right) \leftarrow Parse(\sigma)$ [where $Parse(\sigma)$ parses σ into its components]
$\overline{\mathbf{z}} \leftarrow Retrieve\left(\overline{I(\mathbf{z})}\right)$ [where $Retrieve\left(\overline{I(\mathbf{z})}\right)$ gives back the vector whose non-zero positions are as per the index $\overline{I(\mathbf{z})}$]
$\overline{\mathbf{s}} \leftarrow \mathbf{z}\mathbf{H}^T$
$\overline{\mathbf{m}} \leftarrow \mathcal{H}\left(\mathcal{H}(\texttt{msg})||i\right)$
if $\overline{\mathbf{m}} = \overline{\mathbf{s}}$ **then** put $b = 1$
else put $b = 0$
end if
return b

In [14], it was noted that $t!$ hashing and decoding attempts are required to obtain a decodable $\mathbf{s}_j$. This requirement highlighted a significant flaw in the CFS signature scheme. To address this issue, the parameters of the Goppa code were selected to ensure at least 80-bit security against the Canteaut-Chabaud ISD attack [11], with t set as low as possible. This

adjustment leads to an increased code rate, defined as the ratio k/n. However, this change represented an additional limitation of the CFS signature scheme, especially following the distinguishing attack outlined in [20].

5.2. Wave Signature Protocol

Addressing the limitations of the CFS scheme presents a significant challenge. However, in 2018, a new signature scheme, Wave [17], was introduced. This scheme proposes using *ternary generalised* $(U, U+V)$ *codes* and the *decoding one out of many (DOOM)* problem as the underlying hard problem, rather than relying merely on the SDP.

The public parameters are set as follows:

- A parity-check matrix $\mathbf{H}$ of a ternary generalized $(U, U+V)$ code is selected uniformly at random from the set of all $(n-k) \times n$ parity-check matrices of generalized admissible $(U, U+V)$ codes.
- An $n \times n$ permutation matrix $\mathbf{P}$ and an $(n-k) \times (n-k)$ non-singular matrix $\mathbf{S}$ are also chosen uniformly at random.
- The public (verification) key is calculated as $\overline{\mathbf{H}} = \mathbf{SHP}$, while $(\mathbf{S}, \mathbf{H}, \mathbf{P})$ are kept as the secret (signing) key.
- A hash function $\mathcal{G} : \{0,1\}^* \to \{0,1\}^{(n-k)}$ is selected, along with an agreed-upon value for the *security parameter* λ.

Based on all these parameters, the signature and verification algorithms are as given in Algorithms 3 and 4, respectively.

Algorithm 3 Wave signature algorithm

Input: $\mathbf{H}$, $\mathcal{G}$, `msg`

Output: Signature `sig` on `msg`

1: $\mathbf{w} \leftarrow \mathcal{G}(\texttt{msg})$
2: $\mathbf{r} \in_R \{0,1\}^\lambda$
3: $\mathbf{y} \leftarrow \mathcal{G}(\mathbf{w}||\mathbf{r})$
4: **if** $\mathbf{y}$ is not decodable **then** go to 2
5: **else** $\mathbf{z} \leftarrow Dec_{\mathbf{H}}(\mathbf{y}(\mathbf{S}^{-1})^T)$ [where $Dec_{\mathbf{H}}(\mathbf{y}(\mathbf{S}^{-1})^T)$ decodes $\mathbf{y}(\mathbf{S}^{-1})^T$ w.r.t. $\mathbf{H}$ and gives an n-bit vector $\mathbf{z}$ of weight w]
6: **end if**
7: return $\texttt{sig} \leftarrow (\mathbf{zP}||\mathbf{r})$

Algorithm 4 Verification in Wave

Input: $\overline{\mathbf{H}}$, $\mathcal{G}$, $(\texttt{msg}, \texttt{sig})$

Output: $v \in \{0,1\}$

1: $(\overline{\mathbf{z}}, \mathbf{r}) \leftarrow Parse(\texttt{sig})$ [where $Parse(\texttt{sig}$ gives back the components of `sig`]
2: $\overline{\mathbf{w}} \leftarrow \mathcal{G}(\texttt{msg})$
3: $\overline{\mathbf{y}} \leftarrow \mathcal{G}(\overline{\mathbf{w}}||\mathbf{r})$
4: **if** $\overline{\mathbf{z}}\overline{\mathbf{H}}^T = \overline{\mathbf{y}}$ **then** set $v = 1$
5: **else** set $v = 0$
6: **end if**
7: return v

Unlike existing schemes that require the weight of the decoded vector to be below the Gilbert-Varshamov (GV) bound, the Wave approach focuses on requiring the weight to be above this bound, but still low enough to maintain the difficulty of the decoding problem. This innovation enables more efficient decoding of a hashed value than the CFS scheme, which often requires multiple attempts before obtaining a decodable syndrome. Additionally, Wave does not require a structured code, meaning that U and V can be any random codes of dimensions $(n-k) \times (n/2)$. The Wave method's efficiency made it suitable for submission in Round 1 (Additional Digital Signatures) of the NIST PQC process [5].

6. Comparisons and Analysis

For each of the schemes mentioned in this paper, all the authors asserted that their respective constructions offer a certain level of security. It is notable that, in each scheme, the underlying security assumption was shaped by the complexity of the best-known attacks available at the time the scheme was proposed. For instance, some of the earliest code-based identification schemes- such as those of Stern's and Véron's, were developed under the assumption that 2^{20} operations constituted a large security margin in a classical setup. The concepts of quantum computers and quantum/post-quantum security emerged much later, long after some of these schemes had already been developed. Consequently, the original parameter values were revised in subsequent years and decades to ensure that the key sizes and signature sizes are not too large, while still meeting the minimum security requirement.

A similar case arose with respect to birthday attacks on hash functions and various structural attacks on the underlying code families. In today's era, 128-bit security is regarded as the minimum threshold value for the security of any scheme. However, to date, for many of the schemes mentioned in this paper, there is no widely accepted and standardized set of parameter values that meet this level of security.

Nevertheless, Table 1, compiles the parameter sets recommended in the original publications of the respective schemes, along with the corresponding key and signature sizes as computed in our analysis. Additionally, we have attempted to present the key generation, signing, and verification times in an asymptotic form. The table also summarizes the principal advantages or limitations associated with each scheme.

6.1. Security Discussion in the Context of Modern Attacks

Many of the constructions were proposed when significantly lower computational bounds were considered secure, and therefore, their original parameter choices no longer reflect contemporary threat models. For zero-knowledge identification schemes such as Stern's and Véron's protocols, along with their refinements (CVE, AGS, GPS), modern information-set decoding (ISD) techniques—such as BJMM [6] and May–Ozerov [26] have substantially reduced the effective security margins.

The CFS hash-and-decrypt signature scheme similarly relies on syndrome decoding below the Gilbert–Varshamov (GV) bound. However, the low-weight requirement makes the scheme very prone to ISD attacks. With modern advancements and ISD variants, the CFS scheme is unusable in today's era.

The LESS scheme avoids explicit syndrome decoding and offers better efficiency. Although no polynomial-time classical or quantum attacks are known, recent advances in equivalence testing and canonical form computation reduce security margins, making careful parameter selection essential.

The Wave signature scheme currently offers the strongest security foundation among code-based signatures. By enabling decoding above the GV bound with many valid solutions, it avoids uniqueness-based weaknesses and withstands known classical and quantum-assisted ISD attacks. Despite large key sizes, no effective attacks compromising its unforgeability are known to date.

7. Conclusion and Future Directions

This paper has provided an integrated overview of the principal identification and signature schemes developed within code-based cryptography. While early constructions such as those of Stern, Véron, and their descendants achieved meaningful reductions in soundness error, their

Table 1: Summarization of the different code-based identification and signature schemes

Scheme, Hard Problem	S.E., ZK-type	Claimed Security Level and no. of rounds	Parameters' Values Suggested	K_{pub} Size	K_{pr} Size	Sig Size	$KeyGen$ Time	Signing Time	Verifying Time	Notable Points
Véron [37], GDP	$\frac{2}{3}$, 3-pass	70-bits, 35 rounds, cheating probability: 2^{-20}	n=512, k=256, w=56, q=2, hash digest: 128 bits	≈0.06 KB	≈0.09 KB	≈6 KB	$\mathcal{O}(nk)$	$\mathcal{O}(\lambda nk)$	$\mathcal{O}(\lambda nk)$	•2^{20} operations needed for forgery. •Birthday attack complexity: 2^{64}. •Needs too many rounds and computations to achieve quantum security.
CVE [12], SDP	$\frac{q}{2(q-1)}$, 5-pass	80-bits, 16 rounds, cheating probability: 2^{-16}	n=128, k=64, t=21, w=49, q=256, hash digest: 160 bits	0.062 KB	0.125 KB	6.4 KB	$\mathcal{O}(n(n-k))$	$\mathcal{O}(\lambda n(n-k))$	$\mathcal{O}(\lambda n(n-k))$	•2^{16} operations needed for forgery. •Birthday attack complexity: 2^{80}. •Needs too many rounds and computations to achieve quantum security.
AGS [1], GDP	≈ $\frac{1}{2}$, 5-pass	128-bits, 18 rounds, cheating probability: 2^{-18}	n=1094, k=547, w=70, q=2, hash digest: 256 bits	≈0.13 KB	≈0.2 KB	≈6.5 KB	$\mathcal{O}(nk)$	$\mathcal{O}(\lambda nk)$	$\mathcal{O}(\lambda nk)$	•2^{18} operations needed for forgery. •Birthday attack complexity: 2^{128}. •Paper does not specify the exact digest size of the hash function used.
LESS [4], LEP	$\frac{1}{2}$, 3-pass	≥128-bits against best-known attacks, 907 rounds	n=548, k=274, q=127	129 KB	≈67.9 KB	≈26 KB	$\mathcal{O}(nk(n-k))$	$\mathcal{O}(\lambda(n^2+k^2))$	$\mathcal{O}(\lambda k^2 n)$	•Signature and key sizes reduced using proper optimizations. •Listed in NIST round 1 and 2 (digital signatures). •q is too high for implementations.
GPS [24], SDP	$max(\frac{1}{N},\frac{1}{q})$, 3-pass	>128-bits, 9 rounds	n=196, k=92, w=84, N=2048, s=16, q=512	≈0.12 KB	≈0.2 KB	21.2 KB	$\mathcal{O}(n(n-k)+\lambda+q)$	$\mathcal{O}(\lambda+n+n(n-k))$	$\mathcal{O}(\lambda n(n-k))$	•Signature and key sizes reduced using proper optimizations. •q is too high for implementations.
CFS [14], SDP	−−	≥80-bits	n=2^m=2^{16}, k=n-mt, t=9	1.125 MB	–	144 bits	$\mathcal{O}(1)$	$\mathcal{O}(t!t^2m^3)$	$\mathcal{O}(t^2m)$	•High code rate (the ratio k/n) makes it susceptible to attacks. •Needs $t!$ attempts to obtain one valid signature.
Wave [17, 5], SDP, DOOM	−−	128-bits	n=8576, k=4288, w=7668	3.5 MB	≈27 KB	≈0.8 KB	$\mathcal{O}(1)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n(n-k))$	•Large key size, and the DOOM problem can be exploited by attackers. •Submitted to NIST Round 1.

practical deployment remains limited by large key sizes, substantial communication costs, and nontrivial computational overhead. More recent proposals, most notably the LESS and Wave schemes, represent significant milestones that advance the efficiency and applicability of code-based signatures, offering more promising foundations for post-quantum secure authentication. The survey also highlights a distinguishing strength of the code-based paradigm: its flexibility in leveraging a wide variety of linear codes and its reliance on conceptually simple algebraic structures. These features make the area not only mathematically attractive but also fertile for continued exploration. Despite the notable progress made over the past decades, several challenges remain before code-based signatures can be regarded as efficient alternatives in real-world post-quantum ecosystems. Looking ahead, multiple research avenues merit deeper investigation. A primary objective is to design more compact schemes with reduced key sizes and signature lengths, suitable for constrained devices and large-scale applications. Improvements in decoding algorithms, including their quantum-resistant analysis, may further enhance both efficiency and security. Another promising direction is to identify new families of codes whose structure resists both algebraic and statistical distinguishers while supporting efficient trapdoors. Additionally, protocols employing multi-challenge or aggregated rounds may yield lower communication costs without sacrificing soundness. Finally, practical considerations such as side-channel resistance, constant-time implementations, hardware acceleration, and the feasibility of schemes over large finite fields remain critical for transitioning theoretical constructions into robust, deployable systems. Overall, the developments reviewed in this study reinforce the strong potential of code-based approaches to serve as reliable candidates for post-quantum digital signatures. At the same time, sustained research efforts are essential to refine these constructions, ensuring that they meet the stringent performance, security, and implementability requirements of the post-quantum era.

References

[1] Carlos Aguilar, Philippe Gaborit, and Julien Schrek. A new zero-knowledge code based identification scheme with reduced communication. In *2011 IEEE Information Theory Workshop*, pages 648–652. IEEE, 2011.

[2] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003.

[3] Jean-Philippe Aumasson, Daniel J Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, et al. Sphincs. Technical report, Technical report, Stanford Univ., Tech. Rep, 2019.

[4] Marco Baldi, Alessandro Barenghi, Luke Beckwith, Jean-Fran¸cois Biasse, Tung Chou, Andre Esser, Kris Gaj, Patrick Karl, Kamyar Mohajerani, Gerardo Pelosi, Edoardo Persichetti, Markku-Juhani O Saarinen, Paolo Santini, Robert Wallace, and Floyd Zweydinger. Less specification document-round 2. 2022.

[5] Gustavo Banegas, Kévin Carriet, André Chailloux, Alain Couvreur, Thomas Debris-Alazard, Philippe Gaborit, Pierre Karpman, Johanna Loyer, Ruben Niederhagen, Nicolas Sendrier, Benjamin Smith, and Jean-Pierre Tillich. Wave round 1 submission. 2023.

[6] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1+ 1= 0 improves information set decoding. In *Annual*

international conference on the theory and applications of cryptographic techniques, pages 520–536. Springer, 2012.

[7] Luke Beckwith, Andre Esser, Edoardo Persichetti, Paolo Santini, and Floyd Zweydinger. Less is even more: Optimizing digital signatures from code equivalence. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(4):386–408, 2025.

[8] Mihir Bellare and Shafi Goldwasser. Chapter 10: Digital signatures. *Lecture Notes on Cryptography (PDF)*, page 168, 2008.

[9] Jean-François Biasse, Giacomo Micheli, Edoardo Persichetti, and Paolo Santini. Less is more: code-based signatures without syndromes. In *International Conference on Cryptology in Africa*, pages 45–65. Springer, 2020.

[10] Dan Boneh and Victor Shoup. A graduate course in applied cryptography. *Draft 0.5*, page 14, 2020.

[11] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.

[12] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In *International Workshop on Selected Areas in Cryptography*, pages 171–186. Springer, 2010.

[13] Tung Chou, Edoardo Persichetti, and Paolo Santini. On linear equivalence, canonical forms, and digital signatures. *Designs, Codes and Cryptography*, pages 1–43, 2025.

[14] Nicolas T Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174. Springer, 2001.

[15] NIST CSRC. Post-quantum cryptography: Additional digital signature schemes. `https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures`.

[16] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. Sqisign: compact post-quantum signatures from quaternions and isogenies. In *International conference on the theory and application of cryptology and information security*, pages 64–93. Springer, 2020.

[17] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new code-based signature scheme. 2018.

[18] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *International conference on applied cryptography and network security*, pages 164–175. Springer, 2005.

[19] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.

[20] Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate mceliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

[21] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer, 1986.

[22] Oded Goldreich. *Foundations of Cryptography, Volume 1.* Cambridge University Press, Cambridge, 2001.

[23] Oded Goldreich. *Foundations of Cryptography, Volume 2.* Cambridge University Press, Cambridge, 2004.

[24] Shay Gueron, Edoardo Persichetti, and Paolo Santini. Designing a practical code-based signature scheme from zero-knowledge proofs with trusted setup. *Cryptography*, 6(1):5, 2022.

[25] Katz Jonathan and Lindell Yehuda. Chapter 12: Digital signature schemes. *Introduction to Modern Cryptography*, page 399, 2007.

[26] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 203–228. Springer, 2015.

[27] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. In *Jet Propulsion Laboratory: Deep Space Network Progress Report*, volume 42–44, pages 114–116. NASA, 1978.

[28] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.

[29] Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. In *Post-quantum cryptography*, pages 95–145. Springer, 2009.

[30] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[31] Nicolas Sendrier. Code-based cryptography: State of the art and perspectives. *IEEE Security & Privacy*, 15(4):44–50, 2017.

[32] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4*, pages 47–53. Springer, 1985.

[33] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[34] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[35] Harshdeep Singh. Code based cryptography: Classic mceliece. *arXiv preprint arXiv: 1907.12754*, 2019.

[36] Jacques Stern. A new identification scheme based on syndrome decoding. In *Annual International Cryptology Conference*, pages 13–21. Springer, 1993.

[37] Pascal Véron. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1997.