Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 18** : 1–20.
ISSN-0037-8712
doi:10.5269/bspm.82997

Constash: A Post-Quantum Hash Function from Constacyclic Codes over $R = \mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q$

Asmaa Cherkaoui and Seddik Abdelalim

ABSTRACT: In this work, we introduce *Constash*, a new syndrome-based hash construction built from λ-constacyclic codes over a finite Frobenius ring. The proposed design maps each input to a sparse vector over the ring and computes its syndrome with respect to a public parity-check matrix. In this setting, preimage resistance is related to bounded-weight syndrome decoding, while collision resistance is related to low-weight kernel relations. The ring structure also supports efficient evaluation in the negacyclic case. We discuss concrete parameter choices and practical attack baselines, giving a ring-based perspective on post-quantum syndrome hashing.

Keywords: Post-quantum cryptography, code-based hashing, constacyclic codes, frobenius rings, gray map, NTT.

Contents

2020 *Mathematics Subject Classification:* 94A60, 11T71.
Submitted March 25, 2026. Published June 22, 2026.

1. Introduction

The advent of large-scale quantum computers poses a serious threat to widely deployed cryptographic primitives based on factoring or discrete logarithms. In response, the National Institute of Standards and Technology (NIST) initiated a standardization process for post-quantum cryptography in 2016, spurring significant research into alternatives rooted in mathematical problems believed to resist quantum attacks. Among these, code-based cryptography stands out for its long history of scrutiny and foundational hardness results dating back to the NP-completeness of syndrome decoding [15].

Within code-based primitives, hash functions have received less attention than encryption or signatures, yet they play a critical role in constructing secure protocols such as commitment schemes, Merkle trees, and hash-and-sign signatures. The most influential framework for code-based hashing is the syndrome-based paradigm introduced by Augot, Finiasz, and Sendrier [12], which defines the hash output as the syndrome of a low-weight error vector derived from the message. This design reduces preimage and collision resistance to well-studied decoding problems, specifically, bounded-weight syndrome decoding and the search for low-weight codewords. Early instantiations like Fast Syndrome-Based (FSB) [12] demonstrated feasibility, and subsequent optimizations such as Really Fast Syndrome-Based (RFSB) [10] improved efficiency using quasi-cyclic structure and fast arithmetic. However, the use of highly structured codes has also enabled cryptanalytic advances: Fouque and Leurent [27] showed that certain quasi-cyclic syndrome-hash designs are vulnerable to algebraic and statistical attacks when the underlying code exhibits hidden linearity or low-density patterns.

Motivated by the need for structurally diverse post-quantum hash constructions, we propose *Constash*, a new syndrome-based hash family built from λ-constacyclic codes over the finite Frobenius ring

$$R = \mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q.$$

The construction uses the algebraic structure of R, in particular its idempotent decomposition and the associated componentwise description of constacyclic codes over $\mathbb{F}_q$ [22]. This framework allows the compression map to be defined through sparse ring-valued syndromes, while the negacyclic case admits efficient evaluation by polynomial arithmetic and, when $2n \mid (q-1)$, by the number-theoretic transform (NTT) based methods applied componentwise over $\mathbb{F}_q$ [13,34]. In contrast with earlier syndrome-based hash constructions defined purely over fields, *Constash* relies on a deterministic sparse encoder over R with coefficients in a ring alphabet induced from a base set $S \subseteq \mathbb{F}_q$. Its security is analyzed through bounded Lee-weight syndrome decoding and low-weight kernel relations associated with the public parity-check matrix, together with practical baselines from decoding-style and generalized birthday attacks [12,11,10].

The remainder of this paper is organized as follows. Section 2 introduces the necessary algebraic and coding-theoretic background over R. Section 3 presents the construction of *Constash*, including its Merkle–Damgård iteration. Section 4 discusses the efficient evaluation of the compression function. Section 5 provides a detailed security analysis. Section 6 addresses parameter selection and concrete instantiations. Finally, Section 7 illustrates the construction with a concrete example.

2. Preliminaries

Let $\mathbb{F}_q$ be a finite field with q elements, where $q = p^m$ and p is a prime. Throughout the paper, we consider the commutative ring

$$R = \mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q, \qquad u^2 = u, \quad v^2 = v, \quad uv = vu.$$

Every element $r \in R$ admits a unique representation

$$r = a + bu + cv + duv, \qquad a, b, c, d \in \mathbb{F}_q.$$

We recall that R is a finite Frobenius ring and that it decomposes through the orthogonal idempotents

$$e_1 = (1-u)(1-v), \qquad e_2 = u(1-v), \qquad e_3 = (1-u)v, \qquad e_4 = uv,$$

which satisfy

$$e_i^2 = e_i, \qquad e_i e_j = 0 \ (i \neq j), \qquad e_1 + e_2 + e_3 + e_4 = 1.$$

Accordingly,

$$R = e_1\mathbb{F}_q \oplus e_2\mathbb{F}_q \oplus e_3\mathbb{F}_q \oplus e_4\mathbb{F}_q,$$

and each element $r \in R$ can be written uniquely in the form

$$r = e_1a_1 + e_2a_2 + e_3a_3 + e_4a_4, \qquad a_i \in \mathbb{F}_q,$$

see [22].

We also use the Gray map

$$\Phi : R \longrightarrow \mathbb{F}_q^4, \qquad \Phi(a + bu + cv + duv) = (d,\ c + d,\ b + d,\ a + b + c + d),$$

extended componentwise to R^n. The Lee weight on R^n is defined by

$$wt_L(x) := wt_H(\Phi(x)),$$

where wt_H denotes the Hamming weight over $\mathbb{F}_q$. This point of view allows one to measure vectors over R through their q-ary images; see [22].

Let $\lambda \in R^*$. A linear code $C \subseteq R^n$ is called λ-constacyclic if it is invariant under the λ-constacyclic shift

$$T_\lambda(c_0, c_1, \ldots, c_{n-1}) = (\lambda c_{n-1}, c_0, \ldots, c_{n-2}).$$

Equivalently, C may be identified with an ideal of the quotient ring

$$R[x]/\langle x^n - \lambda\rangle.$$

The structure of constacyclic codes over R is governed by the above idempotent decomposition. More precisely, if

$$C = e_1C_1 \oplus e_2C_2 \oplus e_3C_3 \oplus e_4C_4,$$

then C is a λ-constacyclic code over R if and only if each C_i is a λ_i-constacyclic code of length n over $\mathbb{F}_q$, where $\lambda = \sum_{i=1}^4 e_i\lambda_i$; see [22]. In particular, if $g_i(x)$ denotes the generator polynomial of C_i, then C is generated by the corresponding idempotent combination of these component generators [22].

This decomposition also gives a convenient matrix description. If G_i is a generator matrix of C_i over $\mathbb{F}_q$ for $1 \le i \le 4$, then a generator matrix of C over R is obtained from the idempotent-weighted blocks

$$G = \begin{pmatrix} e_1G_1 \\ e_2G_2 \\ e_3G_3 \\ e_4G_4 \end{pmatrix}.$$

Thus the code over R is assembled from four component codes over $\mathbb{F}_q$, while remaining intrinsically defined over the ring R.

Finally, if $C \subseteq R^n$ is a linear code of rank k, a parity-check matrix of C is a matrix

$$H \in R^{(n-k)\times n}$$

such that

$$C = \{x \in R^n :\ Hx^{\mathsf{T}} = 0\}.$$

Equivalently, H may be taken as any generator matrix of the dual code $C^\perp$. This parity-check viewpoint will be used in the definition of the proposed syndrome-based hash construction.

3. The *Constash* Hash Function

This section presents *Constash*, a code-based hash construction in the syndrome-hashing spirit of Augot, Finiasz, and Sendrier [12]. As in our previous hash proposal [4], the full hash function is placed in a Merkle–Damgård-style framework. However, the underlying compression mechanism considered here is different: it is defined through sparse syndromes over λ-constacyclic codes on the finite Frobenius ring $R = \mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q$, where $u^2 = u,\ v^2 = v,\ uv = vu$. Let $C = e_1C_1 \oplus e_2C_2 \oplus e_3C_3 \oplus e_4C_4$ be a λ-constacyclic code over R of length n and rank k, and let $H \in R^{(n-k)\times n}$ be a parity-check matrix of C.

The compression mechanism is based on a deterministic map

$$t \in \{0,1\}^* \longmapsto z(t) \in R^n,$$

where $z(t)$ is a sparse vector with prescribed support size and coefficients in a fixed subset of R. The compression output is then defined as the syndrome of $z(t)$ with respect to the public matrix H.

To define $z(t)$, fix a finite subset $S \subseteq \mathbb{F}_q$, and set

$$B = \left\{ \sum_{j=1}^{4} e_j s_j \;\middle|\; s_j \in S \right\} \subseteq R, \tag{3.1}$$

where

$$e_1 = (1-u)(1-v), \qquad e_2 = u(1-v), \qquad e_3 = (1-u)v, \qquad e_4 = uv.$$

For a prescribed sparsity parameter w, define

$$\mathcal{E}_{n,w,B} = \{ z \in R^n :\ |\operatorname{supp}(z)| = w,\ z_i \in B \setminus \{0\} \text{ for } i \in \operatorname{supp}(z) \}.$$

Proposition 3.1 *Let $S \subseteq \mathbb{F}_q$ be a finite coefficient set, and define*

$$B = \left\{ \sum_{j=1}^{4} e_j s_j :\ s_j \in S \right\} \subseteq R.$$

Then

$$|B| = |S|^4.$$

Proof: Since

$$R = e_1\mathbb{F}_q \oplus e_2\mathbb{F}_q \oplus e_3\mathbb{F}_q \oplus e_4\mathbb{F}_q,$$

each element of B is uniquely determined by a 4-tuple $(s_1, s_2, s_3, s_4) \in S^4$. Hence the map

$$S^4 \longrightarrow B, \qquad (s_1, s_2, s_3, s_4) \longmapsto \sum_{j=1}^{4} e_j s_j$$

is bijective, and therefore $|B| = |S|^4$. □

Remark 3.1 *Proposition 3.1 is a purely combinatorial statement. It shows that the ring-based coefficient alphabet B is larger than the underlying field alphabet S, which may be useful when balancing sparsity and encoding-domain size.*

Thus, the coefficient alphabet over R grows naturally from the base set $S \subseteq \mathbb{F}_q$, providing additional flexibility in the design of sparse encodings. We now describe how the vector $z(t)$ is sampled deterministically from the input.

The vector $z(t)$ is obtained from a deterministic expansion stream

$$\sigma = \mathrm{XOF}\Big(\text{“Constash”} \,\|\, \mathrm{enc}(q, n, k, w, \lambda) \,\|\, t \Big), \tag{3.2}$$

where enc$(\cdot)$ denotes a canonical encoding of the public parameters. The stream σ is then parsed as described in Algorithm 1.

Algorithm 1 Deterministic sampling of the sparse vector $z(t)$

Require: Expansion stream σ, parameters n, w, S, idempotents e_1, e_2, e_3, e_4
Ensure: A vector $z(t) \in \mathcal{E}_{n,w,B}$

Initialize $z \leftarrow (0, \ldots, 0) \in R^n$, $I \leftarrow \emptyset$, and $\ell \leftarrow 0$
while $\ell < w$ **do**
    Parse bits from σ and use rejection sampling to obtain

$$i \in \{0, \ldots, n-1\} \setminus I$$

    Parse bits from σ and use rejection sampling to obtain

$$(s_1, s_2, s_3, s_4) \in S^4$$

    Set

$$b \leftarrow \sum_{j=1}^{4} e_j s_j \in B$$

    if $b = 0$ **then**
        continue
    end if
    Set $z_i \leftarrow b$
    Update $I \leftarrow I \cup \{i\}$
    Set $\ell \leftarrow \ell + 1$
end while
return $z(t) \leftarrow z$

Proposition 3.2 *Algorithm 1 defines a deterministic map*

$$t \longmapsto z(t) \in \mathcal{E}_{n,w,B}.$$

In particular, $z(t)$ has exactly w distinct nonzero coordinates, all belonging to $B \setminus \{0\}$.

Proof: The stream σ is deterministically derived from t and the public parameters through (3.2). The algorithm starts from the zero vector in R^n and inserts one nonzero entry at each accepted step. Since each sampled index lies in $\{0, \ldots, n-1\} \setminus I$, no position is selected twice. Moreover, each accepted coefficient is of the form $b = \sum_{j=1}^{4} e_j s_j$, hence belongs to B, and the rejection step excludes the case $b = 0$. Therefore, after exactly w accepted iterations, the output belongs to $\mathcal{E}_{n,w,B}$. □

Proposition 3.3 *Let*

$$M_B := \max_{b \in B \setminus \{0\}} wt_L(b).$$

Then every output $z(t)$ of Algorithm 1 satisfies

$$wt_L\big(z(t)\big) \leq w\, M_B.$$

Proof: By Proposition 3.2, the vector $z(t)$ has exactly w nonzero coordinates, each belonging to $B \setminus \{0\}$. Hence each nonzero coordinate contributes at most M_B to the Lee weight, and therefore

$$wt_L\big(z(t)\big) \leq w\, M_B.$$

□

Corollary 3.1 *For every output $z(t)$ of Algorithm 1,*

$$wt_L\big(z(t)\big) \leq 4w.$$

Proof: Since $\Phi(b) \in \mathbb{F}_q^4$ for every $b \in R$, one has

$$wt_L(b) = wt_H(\Phi(b)) \leq 4.$$

In particular, $M_B \leq 4$. The conclusion then follows from Proposition 3.3. □

Remark 3.2 *Corollary 3.1 is used only as a conservative estimate. Via the Gray map, it yields an explicit Hamming-weight upper bound for the associated q-ary representation of the sampled vectors. This is useful for parameter discussion and comparison, although the bound is not expected to be tight in general.*

The compression function associated with the proposed construction is the public syndrome map

$$\mathsf{Constash}(t) = H\, z(t)^{\mathsf{T}} \in R^{n-k}, \tag{3.3}$$

where $H \in R^{(n-k)\times n}$ is a parity-check matrix and $z(t) \in R^n$ is produced by Algorithm 1.

Remark 3.3 *The construction is defined natively over the ring R. When a comparison with field-valued syndrome hashes is desired, one may apply the Gray map componentwise to the output $Hz(t)^{\mathsf{T}}$, thereby obtaining a vector in $\mathbb{F}_q^{4(n-k)}$. This representation is used only for comparison or serialization purposes, and is not part of the core definition of the hash.*

3.1. The Merkle–Damgård construction of *Constash*

In order to process inputs of arbitrary length, the proposed compression mechanism is placed in a Merkle–Damgård-style framework. The input bitstream t is first padded and partitioned into blocks

$$t^{(1)}, \ldots, t^{(i)}.$$

For each block $t^{(j)}$, the deterministic encoder produces a sparse vector $z(t^{(j)}) \in R^n$, and the corresponding compression output is computed as

$$y^{(j)} = H\, z(t^{(j)})^{\mathsf{T}} \in R^{n-k},$$

where $H \in R^{(n-k)\times n}$ is the parity-check matrix associated with the chosen constacyclic code over R. The resulting outputs are chained from one round to the next, starting from an initialization value $IV \in R^{n-k}$, until the last block is processed. The final value $y^{(i)}$ is then taken as the output of the iterated construction. Figure 1 summarizes this Merkle–Damgård-style organization of the proposed hash.

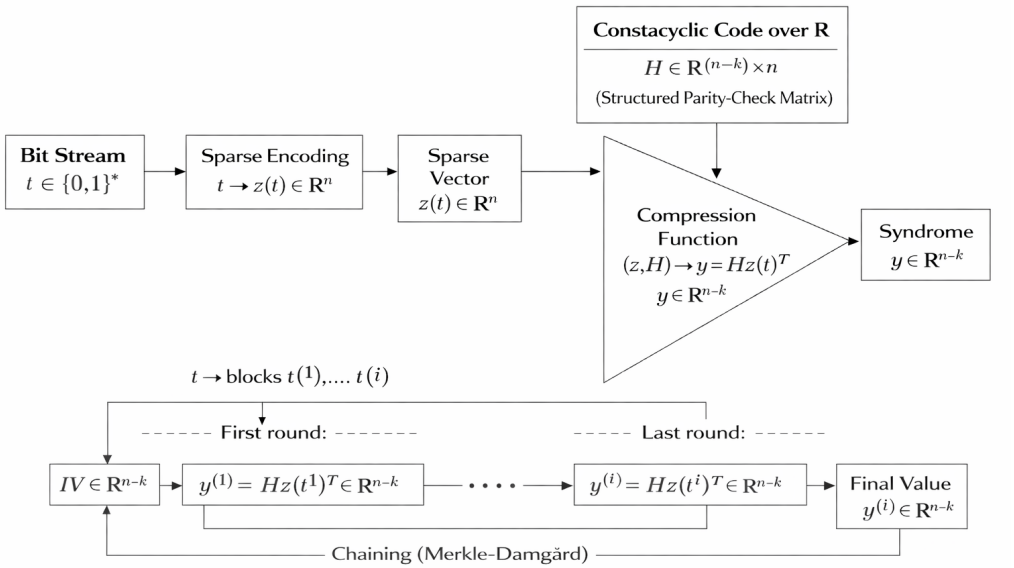

Figure 1: Merkle–Damgård-style organization of the proposed *Constash* hash.

4. Efficient Evaluation of the Compression Function

The compression map

$$\mathsf{Constash}(t) = H\, z(t)^{\mathsf{T}}$$

can be evaluated efficiently by exploiting both the constacyclic structure of the code and the sparsity of the vector $z(t)$.

Indeed, vectors in R^n are naturally identified with polynomials in the quotient ring

$$S_\lambda = R[x]/\langle x^n - \lambda\rangle.$$

Under this identification, the vector $z(t) \in R^n$ corresponds to a polynomial $z(x) \in S_\lambda$, and each row of the parity-check matrix H may be represented by a polynomial $h_j(x) \in S_\lambda$. The j-th coordinate of the syndrome is then obtained as the product

$$s_j = h_j(x)\, z(x) \pmod{x^n - \lambda}.$$

Thus, the computation of $Hz(t)^{\mathsf{T}}$ reduces to a collection of polynomial multiplications in S_λ.

In the negacyclic case $\lambda = -1$, one works in the ring

$$R[x]/\langle x^n + 1\rangle.$$

Using the decomposition

$$R = e_1\mathbb{F}_q \oplus e_2\mathbb{F}_q \oplus e_3\mathbb{F}_q \oplus e_4\mathbb{F}_q,$$

each polynomial over R may be treated componentwise over $\mathbb{F}_q$; see [22]. Consequently, multiplication in $R[x]/\langle x^n + 1\rangle$ reduces to four negacyclic multiplications in

$$\mathbb{F}_q[x]/\langle x^n + 1\rangle,$$

followed by recombination through the idempotents. When

$$2n \mid (q-1),$$

the field $\mathbb{F}_q$ contains a primitive $2n$-th root of unity, so these componentwise products can be carried out efficiently by NTT [13]. This yields quasi-linear complexity $O(n \log n)$ for each negacyclic multiplication, and therefore an overall complexity of order

$$O(r\, n \log n)$$

for the evaluation of the full syndrome, where $r = n - k$; see also [34].

On the other hand, the encoder produces a w-sparse vector

$$z(x) = \sum_{\ell=1}^{w} b_\ell x^{i_\ell},$$

with coefficients $b_\ell \in B \subset R$. In this regime, one may also evaluate the syndrome directly by sparse multiplication:

$$h_j(x)\, z(x) = \sum_{\ell=1}^{w} b_\ell\, x^{i_\ell} h_j(x) \pmod{x^n - \lambda}.$$

This requires $O(wn)$ operations per row, hence $O(rwn)$ operations for the full syndrome. Such a strategy may be preferable when $w \ll n$ or when only a single evaluation is required, whereas the NTT-based approach becomes more attractive for larger dimensions or repeated computations.

Therefore, the proposed construction admits two natural evaluation modes: a sparse direct method adapted to low-weight inputs, and a fast polynomial method based on negacyclic arithmetic and NTT in the case $\lambda = -1$. This flexibility is one of the practical advantages of combining sparse encodings with constacyclic structure over the ring R.

5. Security Analysis

In this section, we relate the security of the proposed compression map

$$\mathsf{Constash}(t) = H\, z(t)^{\mathsf{T}} \in R^{n-k}$$

to sparse algebraic search problems over the ring R. As in syndrome-based hash constructions such as FSB and its variants, the basic security intuition is that preimages correspond to solving bounded-weight syndrome equations, while collisions correspond to finding nonzero low-weight vectors in the kernel of the public parity-check matrix [12,10]. However, the present setting differs from the original FSB encoding: our encoder samples support positions globally without replacement and uses coefficients in a ring alphabet $B \subset R$, so the classical blockwise "regular word" formulation does not transfer verbatim.

We therefore analyze the scheme directly in terms of the deterministic sparse encoder

$$t \longmapsto z(t) \in \mathcal{E}_{n,w,B},$$

5.1. Underlying search problems

The following two search problems capture the preimage and collision tasks associated with the proposed compression map.

Definition 5.1 (Bounded Lee-weight syndrome decoding over R) *Let $H \in R^{(n-k)\times n}$, let $y \in R^{n-k}$, and let $W \geq 1$. The bounded Lee-weight syndrome decoding problem consists in finding*

$$e \in R^n$$

such that

$$He^{\mathsf{T}} = y \qquad \text{and} \qquad wt_L(e) \leq W.$$

Definition 5.2 (Low Lee-weight codeword search over R) *Let $H \in R^{(n-k)\times n}$ and let $W \geq 1$. The low Lee-weight codeword search problem consists in finding a nonzero vector*

$$d \in R^n$$

such that

$$Hd^{\mathsf{T}} = 0 \qquad \text{and} \qquad wt_L(d) \leq W.$$

These problems are the natural analogues, in the present ring setting, of the sparse syndrome search problems underlying code-based hashing over finite fields.

5.2. Size of the encoded sparse domain

Since the encoder outputs vectors in $\mathcal{E}_{n,w,B}$, the effective domain of the compression map is not all of R^n, but a structured sparse subset.

Proposition 5.1 *The cardinality of $\mathcal{E}_{n,w,B}$ is*

$$|\mathcal{E}_{n,w,B}| = \binom{n}{w}(|B|-1)^w.$$

Proof: A vector in $\mathcal{E}_{n,w,B}$ is obtained by first choosing its support, which can be done in $\binom{n}{w}$ ways, and then assigning to each support position a nonzero coefficient in $B \setminus \{0\}$, which yields $(|B|-1)^w$ possibilities. Multiplying these two counts gives the formula. □

In view of Proposition 3.1, one has $|B| = |S|^4$. Hence

$$|\mathcal{E}_{n,w,B}| = \binom{n}{w}(|S|^4-1)^w,$$

which shows that the ring-valued coefficient alphabet substantially enlarges the sparse encoding domain relative to a field-based alphabet of size $|S|$.

5.3. Theoretical security relations

We now relate inversion, second-preimage search, and collision search for Constash to the above sparse algebraic problems.

Proposition 5.2 (Preimage relation) *Let $y \in R^{n-k}$. If there exists $t \in \{0,1\}^*$ such that*

$$\mathsf{Constash}(t) = y,$$

then $z(t)$ is a solution to a bounded Lee-weight syndrome decoding instance:

$$Hz(t)^{\mathsf{T}} = y \qquad \text{with} \qquad wt_L(z(t)) \le wM_B,$$

where

$$M_B := \max_{b \in B\setminus\{0\}} wt_L(b).$$

Proof: By definition,

$$\mathsf{Constash}(t) = Hz(t)^{\mathsf{T}}.$$

If $\mathsf{Constash}(t) = y$, then $z(t)$ satisfies

$$Hz(t)^{\mathsf{T}} = y.$$

Moreover, Proposition 3.3 gives

$$wt_L(z(t)) \le wM_B.$$

Hence $z(t)$ is a bounded Lee-weight solution of the syndrome equation. □

Proposition 5.3 (Second-preimage relation) *Let $t \in \{0,1\}^*$ and set*

$$y = \mathsf{Constash}(t) = Hz(t)^{\mathsf{T}}.$$

Any second preimage $t' \neq t$ satisfying

$$\mathsf{Constash}(t') = \mathsf{Constash}(t)$$

yields another vector $z(t') \in \mathcal{E}_{n,w,B}$ such that

$$Hz(t')^{\mathsf{T}} = y \qquad \text{and} \qquad wt_L(z(t')) \le wM_B.$$

Proof: This is the same syndrome equation as in Proposition 5.2, but with the target value fixed to

$$y = Hz(t)^{\mathsf{T}}.$$

Since Algorithm 1 always outputs a vector in $\mathcal{E}_{n,w,B}$, the bound

$$wt_L(z(t')) \leq wM_B$$

again follows from Proposition 3.3. □

Proposition 5.4 (Collision relation) *Let $t, t' \in \{0,1\}^*$ with $t \neq t'$. If*

$$\mathsf{Constash}(t) = \mathsf{Constash}(t'),$$

then the difference

$$d := z(t) - z(t') \in R^n$$

is a nonzero vector satisfying

$$Hd^{\mathsf{T}} = 0$$

and

$$wt_L(d) \leq 2wM_B.$$

In particular, collision search for Constash *yields a low Lee-weight codeword in the kernel of H.*

Proof: If $\mathsf{Constash}(t) = \mathsf{Constash}(t')$, then

$$Hz(t)^{\mathsf{T}} = Hz(t')^{\mathsf{T}}.$$

Subtracting gives

$$H\big(z(t) - z(t')\big)^{\mathsf{T}} = 0.$$

Thus $d := z(t) - z(t')$ lies in the kernel of H. If $d = 0$, then $z(t) = z(t')$. In that case, the collision comes from two distinct inputs mapping to the same sparse vector; otherwise $d \neq 0$ is a nonzero kernel vector. Since

$$d = z(t) - z(t') = z(t) + (-z(t')),$$

the triangle inequality for the Lee weight gives

$$wt_L(d) \leq wt_L(z(t)) + wt_L(-z(t')).$$

Since $wt_L(-x) = wt_L(x)$ for every $x \in R^n$, it follows that

$$wt_L(d) \leq wt_L(z(t)) + wt_L(z(t')) \leq 2wM_B.$$

□

Using Corollary 3.1, one also gets the crude bound

$$wt_L(d) \leq 8w.$$

This estimate is conservative and is used only as a worst-case envelope in later parameter discussions.

5.4. Information set decoding.

We consider information set decoding (ISD) as a baseline for solving the preimage problem

$$y = Hz^{\mathsf{T}}, \qquad z \in R^n, \quad wt_L(z) \le w,$$

where $H \in R^{(n-k)\times n}$ is the public parity-check matrix and $wt_L(\cdot)$ denotes the Lee weight over R. By construction, the encoder produces vectors of the form

$$z = \sum_{\ell=1}^{w} b_\ell e_{i_\ell}, \qquad b_\ell \in B \subset R \setminus \{0\},$$

so that the syndrome admits the representation

$$y = \sum_{\ell=1}^{w} b_\ell \, H_{(\cdot,i_\ell)}.$$

This is a sparse linear combination problem over the ring R, generalizing the classical binary decoding setting.

Following the standard ISD strategy, one first permutes the columns of H and performs Gaussian elimination to obtain a systematic form $H' = (I_r \mid A)$. The problem then reduces to finding a subset of indices $\{i_1, \dots, i_w\}$ together with coefficients $b_\ell \in B$ such that the above relation holds.

To estimate the complexity, we adopt a meet-in-the-middle approach by splitting the support into two subsets of size p. Let

$$L(p) = \binom{(n-r)/2}{p} \cdot (|B|-1)^p,$$

which counts the number of partial sums of the form

$$\sum_{j=1}^{p} b_j \, H_{(\cdot,i_j)}.$$

The expected number of representations of the target syndrome is approximated by

$$N(p,\ell) \approx \frac{|R|^{\,r}}{L(p)^2 \cdot \binom{r-\ell}{w-2p} \, (|B|-1)^{\,w-2p}},$$

reflecting the size of the syndrome space and the number of admissible completions.

Let

$$L'(p,\ell) = L(p)\sqrt{N(p,\ell)}$$

denote the effective list size after matching. The cost of one iteration is modeled by

$$C(p,\ell,X) = 2X\big(\log_2(X) + \ell\big) + \frac{X^2}{|R|^{\,\ell}},$$

where the second term accounts for collisions between partial sums in the reduced space.

The total workfactor is then estimated as

$$WF(n,r,w) = \min_{p,\ell} K(p,\ell),$$

with

$$K(p,\ell) = \begin{cases} N(p,\ell)\, C(p,\ell,L(p)) & \text{if } N(p,\ell) > 1, \\ C(p,\ell,L'(p,\ell)) & \text{otherwise.} \end{cases}$$

5.5. Generalized birthday attacks.

The compression equation

$$y = Hz(t)^{\mathsf{T}} = \sum_{\ell=1}^{w} b_\ell \, H_{(\cdot,i_\ell)}, \qquad b_\ell \in B \setminus \{0\},$$

shows that preimage search can be viewed as a weighted sparse-sum problem in the additive group of R^r, where $r = n - k$. Since $|R| = q^4$, this group has cardinality

$$|R^r| = q^{4r},$$

and may be identified, through any fixed $\mathbb{F}_q$-basis of R, with an additive space of dimension $4r$ over $\mathbb{F}_q$. Accordingly, Wagner's generalized birthday method may be used as a heuristic baseline in the present setting, with the binary XOR of the classical formulation replaced by addition in R^r [11].

To model this attack, fix an integer $a \geq 1$, and partition the set of column indices $\{1, \ldots, n\}$ into 2^a disjoint blocks of sizes as equal as possible. For preimage search, the w nonzero positions are distributed among these blocks, and we write

$$m_a = \left\lceil \frac{w}{2^a} \right\rceil.$$

Each initial list is then formed from partial sums of the form

$$\sum_{j=1}^{m_a} b_j \, H_{(\cdot,i_j)},$$

where the indices i_j are taken inside one fixed block and the coefficients satisfy $b_j \in B \setminus \{0\}$. Hence the number of available partial sums in one block is bounded by

$$N_a^{\mathrm{pre}} \leq \binom{\lfloor n/2^a \rfloor}{m_a} (|B| - 1)^{m_a}. \tag{5.1}$$

Following Wagner's balancing principle, we heuristically target list sizes of order

$$L_a \approx q^{\frac{4r}{a+1}}, \tag{5.2}$$

which corresponds to forcing approximately $4r/(a+1)$ q-ary coordinates at each merging level. A necessary heuristic feasibility condition for this attack is therefore

$$q^{\frac{4r}{a+1}} \leq \binom{\lfloor n/2^a \rfloor}{m_a} (|B| - 1)^{m_a}. \tag{5.3}$$

For the largest admissible value of a, the corresponding Wagner-type preimage baseline is heuristically of order

$$\mathrm{GBA}_{\mathrm{pre}}(n, r, w; B) = \widetilde{O}\left(q^{\frac{4r}{a+1}}\right), \tag{5.4}$$

up to polynomial factors in r arising from sorting and list handling.

The collision problem is modeled in the same spirit after passing to differences. Indeed, if

$$\mathsf{Constash}(t) = \mathsf{Constash}(t'),$$

then

$$0 = H\big(z(t) - z(t')\big)^{\mathsf{T}}.$$

Writing

$$D := (B - B) \setminus \{0\},$$

the difference vector $d = z(t) - z(t')$ has support size at most $2w$ and coefficients in the difference alphabet D. Setting

$$m'_a = \left\lceil \frac{2w}{2^a} \right\rceil,$$

one obtains the coarse upper bound

$$N_a^{\mathrm{col}} \le \binom{\lfloor n/2^a \rfloor}{m'_a} |D|^{m'_a}. \tag{5.5}$$

Thus a Wagner-type collision search is heuristically feasible only if

$$q^{\frac{4r}{a+1}} \le \binom{\lfloor n/2^a \rfloor}{m'_a} |D|^{m'_a}, \tag{5.6}$$

and in that regime its heuristic workfactor is of order

$$\mathrm{GBA}_{\mathrm{col}}(n, r, w; B) = \widetilde{O}\left(q^{\frac{4r}{a+1}}\right). \tag{5.7}$$

This generalized birthday model plays, for the present ring-valued sparse encoder, the same conceptual role that Wagner-type analysis plays in earlier syndrome-based hash constructions [12,10]. The difference is that the counting is now performed from the actual sparse domain

$$z(t) = \sum_{\ell=1}^{w} b_\ell e_{i_\ell}, \qquad b_\ell \in B \setminus \{0\},$$

with global support sampling and ring-valued coefficients, rather than from the original one-choice-per-block binary model.

5.6. Discussion of the encoder-specific security features

The preceding analysis already shows why the deterministic encoder $t \mapsto z(t)$ is central to the security of the scheme.

First, the encoder constrains every input to a structured sparse subset $\mathcal{E}_{n,w,B} \subset R^n$, so both preimages and collisions are forced into low-weight algebraic relations. Second, the use of the ring alphabet

$$B = \left\{ \sum_{j=1}^{4} e_j s_j : \ s_j \in S \right\}$$

enlarges the coefficient space available at each nonzero position, which increases the sparse-domain size without changing the number w of selected positions. Third, because support positions are sampled globally without replacement, the encoder does not exhibit the original one-choice-per-block structure used in the classical FSB formulation. This does not by itself prove resistance to all specialized attacks, but it changes the attack surface and prevents a direct transfer of the original regular-word analysis.

For these reasons, the security discussion for *Constash* should be understood as combining two levels:

1. exact algebraic relations between hash attacks and sparse linear problems over R;
2. conservative practical evaluation against the best known decoding and generalized birthday baselines.

5.7. Heuristic combinatorial estimates

Let

$$\mathcal{Z} := \mathcal{E}_{n,w,B} = \{ z \in R^n : \ |\operatorname{supp}(z)| = w, \ z_i \in B \setminus \{0\} \text{ for } i \in \operatorname{supp}(z) \}.$$

By Proposition 5.1,

$$|\mathcal{Z}| = \binom{n}{w} (|B| - 1)^w.$$

This quantity plays, in the present setting, the same combinatorial role as the number of admissible regular words in earlier syndrome-based hash constructions [12,10], although our encoder is different and no blockwise regular-word model is assumed here.

Assuming heuristically that the map

$$z \longmapsto Hz^{\mathsf{T}} \in R^r, \qquad r = n - k,$$

is close to uniform on R^r when restricted to $\mathcal{Z}$, the expected number of sparse solutions to a fixed syndrome equation

$$Hz^{\mathsf{T}} = y$$

is approximately

$$\mathbb{E}[N_{\mathrm{pre}}] \approx \frac{|\mathcal{Z}|}{|R|^r} = \frac{\binom{n}{w}(|B|-1)^w}{q^{4r}}. \tag{5.8}$$

Similarly, the expected number of ordered pairs $(z, z') \in \mathcal{Z}^2$ satisfying

$$Hz^{\mathsf{T}} = Hz'^{\mathsf{T}}$$

is approximately

$$\mathbb{E}[N_{\mathrm{col}}] \approx \frac{|\mathcal{Z}|^2}{|R|^r} = \frac{\binom{n}{w}^2(|B|-1)^{2w}}{q^{4r}}. \tag{5.9}$$

A collision yields a difference vector

$$d = z - z' \in R^n$$

with support size at most $2w$. Writing

$$D := (B - B) \setminus \{0\},$$

a crude upper bound on the number of nonzero difference patterns is

$$N_{\mathrm{diff}} \le \sum_{j=1}^{2w} \binom{n}{j} |D|^j. \tag{5.10}$$

This estimate is useful when relating collision search to low-weight kernel relations of the public parity-check matrix.

Finally, since every encoder output has the form

$$Hz(t)^{\mathsf{T}} = \sum_{\ell=1}^{w} b_\ell H_{(\cdot, i_\ell)}, \qquad b_\ell \in B \setminus \{0\},$$

the problem also fits the general sparse-sum framework underlying generalized birthday attacks of Wagner type [11]. The relevant search space is now determined by global support choices and ring-valued coefficients, rather than by the original FSB one-choice-per-block model.

6. Parameter Selection and Concrete Instantiation

This section records the quantitative constraints attached to the proposed compression map

$$\mathsf{Constash}(t) = H\, z(t)^{\mathsf{T}} \in R^{n-k}.$$

The purpose is twofold: first, to make explicit the exact combinatorial quantities determined by the sparse encoder; second, to identify the baselines against which concrete security must be evaluated. Throughout this section, we write

$$r = n - k.$$

6.1. Exact quantities determined by the construction

For the encoder defined in Algorithm 1, the cardinal of the sparse domain is

$$|\mathcal{E}_{n,w,B}| = \binom{n}{w}(|S|^4 - 1)^w. \tag{6.1}$$

Accordingly, the associated domain size in bits is

$$\log_2 |\mathcal{E}_{n,w,B}| = \log_2 \binom{n}{w} + w \log_2(|S|^4 - 1). \tag{6.2}$$

The raw compression output lies in R^r. Since $|R| = q^4$, the cardinality of the output space is

$$|R^r| = q^{4r},$$

and its q-ary size in bits is

$$\log_2 |R^r| = 4r \log_2 q. \tag{6.3}$$

If a binary serialization is used, each element of $\mathbb{F}_q$ requires $\lceil \log_2 q \rceil$ bits, so the raw binary length is

$$L_{\text{raw}} = 4r \lceil \log_2 q \rceil. \tag{6.4}$$

From Proposition 3.3 and Corollary 3.1, every encoder output satisfies

$$wt_L(z(t)) \le wM_B \le 4w,$$

and every collision difference

$$d = z(t) - z(t')$$

satisfies

$$wt_L(d) \le 2wM_B \le 8w.$$

These bounds are exact consequences of the construction and are the weight parameters used in the security discussion.

6.2. Generic baselines

The exact quantities above already impose unavoidable generic limits. Let $D := |\mathcal{E}_{n,w,B}|$. Then the sparse encoder can never contribute more than D distinct internal vectors, and the compression output can never exceed the size of R^r. Hence the generic baselines are governed by

$$\min\{D,\ q^{4r}\}.$$

More precisely, generic preimage search is bounded by

$$\min\{D,\ q^{4r}\}, \tag{6.5}$$

whereas generic collision search is bounded by the corresponding birthday term

$$\sqrt{\min\{D,\ q^{4r}\}}. \tag{6.6}$$

If an external binary encoding or truncation to b bits is applied after the ring-valued compression output, then the generic bounds are further capped by 2^b for preimages and $2^{b/2}$ for collisions.

6.3. Concrete instantiation

We now record one concrete parameter set consistent with the preceding constraints. We take

$$q = 65537, \qquad n = 2048, \qquad k = 1024, \qquad r = 1024, \qquad \lambda = -1,$$

and we fix

$$S = \{0, \pm 1\} \subset \mathbb{F}_q.$$

Then

$$|S| = 3, \qquad |B| = |S|^4 = 81, \qquad |B| - 1 = 80.$$

We also take

$$w = 64.$$

The condition $2n \mid (q-1)$ is satisfied, since

$$2n = 4096 \qquad \text{and} \qquad q - 1 = 65536.$$

For these values, the sparse-domain size is

$$D = \binom{2048}{64} 80^{64}, \tag{6.7}$$

and its bitlength is

$$\begin{aligned} \log_2 D &= \log_2 \binom{2048}{64} + 64 \log_2 80 \\ &\approx 406.5698 + 404.6034 \\ &\approx 811.1732. \end{aligned} \tag{6.8}$$

Thus the sparse encoder ranges over approximately $2^{811.17}$ admissible vectors.

The raw output space has size

$$|R^r| = q^{4r} = 65537^{4096},$$

so that

$$\log_2 |R^r| = 4r \log_2 q = 4096 \log_2 65537 \approx 65536.0901. \tag{6.9}$$

Since $\lceil \log_2 q \rceil = 17$, the corresponding raw binary output length is

$$L_{\text{raw}} = 4r \cdot 17 = 4096 \cdot 17 = 69632 \text{ bits}. \tag{6.10}$$

Consequently, before any external truncation is applied, the generic baselines are determined by the smaller of D and q^{4r}, namely by D. Therefore

$$\log_2(\text{generic preimage baseline}) \approx 811.1732, \tag{6.11}$$

and

$$\log_2(\text{generic collision baseline}) \approx \frac{811.1732}{2} \approx 405.5866. \tag{6.12}$$

The corresponding Lee-weight bounds are

$$wt_L(z(t)) \le 4w = 256, \qquad wt_L\big(z(t) - z(t')\big) \le 8w = 512. \tag{6.13}$$

6.4. Practical attack baselines

For the above parameter set, the exact generic quantities are given by (6.8)–(6.12). The non-generic attack surface is governed by the two families identified in Section 5.

First, preimage and second-preimage attacks reduce to bounded Lee-weight syndrome decoding over R, with encoder weight bounded by (6.13). A conservative field-level comparison is obtained by expanding ring symbols into their q-ary representations, which leads to an effective ambient length $4n = 8192$ and weight bound at most 256 for encoder outputs.

Second, collision attacks reduce to low Lee-weight kernel relations, with collision differences bounded by Lee weight 512. At the same time, the sparse-sum form

$$Hz(t)^{\mathsf{T}} = \sum_{\ell=1}^{64} b_\ell H_{(\cdot,i_\ell)}$$

places the construction in the scope of generalized birthday methods of Wagner type [11]. In contrast with the original FSB setting, however, the present encoder is not based on one choice per block; support positions are sampled globally without replacement, and coefficients are taken from the ring alphabet $B \subset R$. Accordingly, the classical regular-word formulas of FSB are not reused here.

The practical security level of a concrete instance is therefore determined by the minimum of the following four baselines:

$$\binom{n}{w}(|B|-1)^w, \qquad q^{4r}, \qquad \text{best known decoding-style cost}, \qquad \text{best known generalized birthday cost}.$$

The first two are exact and have been computed above. The latter two depend on the chosen external cost model for decoding and sparse-sum search and must be evaluated consistently when reporting a final bit-security claim.

7. A Concrete Example

We give a small example illustrating the *Constash* hash function

$$t \longmapsto z(t) \longmapsto Hz(t)^{\mathsf{T}}.$$

The parameters below are chosen only for transparency and are not intended to provide security.

Let $q = 5$, $n = 3$, and $\lambda = 1$. We work over

$$R = \mathbb{F}_5 + u\mathbb{F}_5 + v\mathbb{F}_5 + uv\mathbb{F}_5, \qquad u^2 = u, \quad v^2 = v, \quad uv = vu,$$

with idempotents

$$e_1 = (1-u)(1-v), \qquad e_2 = u(1-v), \qquad e_3 = (1-u)v, \qquad e_4 = uv.$$

We take as component codes the cyclic length-3 code

$$C_i = \langle x-1 \rangle \subseteq \mathbb{F}_5[x]/\langle x^3-1 \rangle, \qquad 1 \le i \le 4.$$

Since

$$x^3 - 1 = (x-1)(x^2+x+1)$$

over $\mathbb{F}_5$, each C_i has dimension 2, and one possible generator matrix is

$$G_i = \begin{pmatrix} 4 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix}.$$

Accordingly, the code

$$C = e_1C_1 \oplus e_2C_2 \oplus e_3C_3 \oplus e_4C_4 \subseteq R^3$$

is a cyclic code over R. A generator matrix of C is obtained from the idempotent-weighted component blocks:

$$G = \begin{pmatrix} 4e_1 & e_1 & 0 \\ 0 & 4e_1 & e_1 \\ 4e_2 & e_2 & 0 \\ 0 & 4e_2 & e_2 \\ 4e_3 & e_3 & 0 \\ 0 & 4e_3 & e_3 \\ 4e_4 & e_4 & 0 \\ 0 & 4e_4 & e_4 \end{pmatrix}.$$

Since the dual of $\langle x-1 \rangle$ over $\mathbb{F}_5$ is generated by x^2+x+1, a parity-check matrix for each component code is

$$H_i = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}.$$

Hence a parity-check matrix for C may be taken as

$$H = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \in R^{1\times 3}.$$

We now fix the coefficient set

$$S = \{0, \pm 1\} = \{0, 1, 4\} \subseteq \mathbb{F}_5,$$

and let $B \subset R$ be the induced ring alphabet

$$B = \left\{ \sum_{j=1}^{4} e_j s_j : \ s_j \in S \right\}.$$

For this example, we take $w = 2$.

Consider an input message $t \in \{0,1\}^*$. Applying Algorithm 1 to the stream

$$\sigma = \mathrm{XOF}\Big(\text{“Constash”} \,\|\, \mathrm{enc}(5,3,2,2,1) \,\|\, t \Big),$$

suppose that the first accepted support positions are

$$i_1 = 0, \qquad i_2 = 2,$$

and that the corresponding accepted coefficients are

$$b_1 = 1, \qquad b_2 = u.$$

Then the encoder outputs

$$z(t) = (1, 0, u) \in R^3.$$

The compression output is now computed directly from the parity-check matrix:

$$\mathsf{Constash}(t) = Hz(t)^{\mathsf{T}} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ u \end{pmatrix} = 1 + u \in R.$$

Thus, in this example, the ring-valued compression output is the element

$$\mathsf{Constash}(t) = 1 + u.$$

If one wishes to compare this output with field-valued syndrome hashes, one may apply the Gray map only at the serialization stage. Using

$$\Phi(a + bu + cv + duv) = (d,\ c+d,\ b+d,\ a+b+c+d),$$

we obtain

$$\Phi(1+u) = \Phi(1 + 1\cdot u + 0\cdot v + 0\cdot uv) = (0,0,1,2) \in \mathbb{F}_5^4.$$

This Gray image is not the native hash output; it is only a q-ary representation of the ring-valued syndrome for comparison or serialization purposes.

8. Conclusion

In this work, we introduced *Constash*, a syndrome-based hash construction built from λ-constacyclic codes over the finite Frobenius ring

$$R = \mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q.$$

The design combines a deterministic sparse encoding with structured parity-check matrices over R, leading to a compression function defined through bounded-weight ring-valued syndromes. The resulting construction fits naturally within the Merkle–Damgård paradigm and provides a new instance of code-based hashing beyond the classical finite-field setting.

From a computational perspective, the constacyclic structure enables efficient evaluation via polynomial arithmetic, with further acceleration available in the negacyclic case through componentwise transforms over $\mathbb{F}_q$. On the security side, the analysis reduces to well-identified problems related to bounded-weight syndrome decoding and low-weight kernel relations, placing the construction within the established landscape of code-based cryptography.

Future work will focus on a detailed exploration of parameter choices and their concrete security levels, including systematic evaluation against classical and quantum attack models. In particular, we plan to derive heuristic complexity estimates for both decoding-based attacks and generalized birthday-type strategies, and to complement the theoretical analysis with implementations and experimental validation.

Acknowledgments

We gratefully acknowledge the organizing committee of ICAME'25 for the announcement and for providing the opportunity to contribute our work to the special issue dedicated to the theme "Applied Mathematics, Modeling, and Engineering." We also extend our sincere appreciation to the journal *Boletim da Sociedade Paranaense de Matemática (BSPM)* for hosting this special issue, and to the editorial teams and reviewers for their valuable efforts in the selection and evaluation process.

References

1. A. Otmani, J.-P. Tillich, and L. Dallot, *Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes*, Math. Comput. Sci. **3**, 129–140, (2010).
2. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, (1996).
3. A. A. Nechaev, *Kerdock code in a cyclic form*, Discrete Math. Appl. **1**, 365–384, (1991).
4. A. Cherkaoui, S. Abdelalim, A. Lkouiza, and I. Elmouki, *A Fisher–Yates shuffle in a hardened Merkle–Damgård hash for the blockchain's PoW*, Statist. Optim. Inf. Comput., to appear, (2025).
5. C. Peters, *Information-set decoding for linear codes over $\mathbb{F}_q$*, in *Post-Quantum Cryptography – PQCrypto 2010*, Lecture Notes in Comput. Sci. **6061**, 81–94, (2010).
6. C. Carlet, *$\mathbb{Z}_{2^k}$-linear codes*, IEEE Trans. Inform. Theory **44**, 1543–1547, (1998).
7. D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe, *FSBday: Implementing Wagner's generalized birthday attack against the SHA-3 round-1 candidate FSB*, in *Progress in Cryptology – INDOCRYPT 2009*, Lecture Notes in Comput. Sci. **5922**, 18–38, (2009).
8. D. J. Bernstein, T. Lange, and C. Peters, *Faster 2-regular information-set decoding*, in *Coding and Cryptology – IWCC 2011*, Lecture Notes in Comput. Sci. **6639**, 1–18, (2011).
9. D. E. Knuth, *The Art of Computer Programming, Volume II: Seminumerical Algorithms*, 3rd ed., Addison–Wesley, (1998).
10. D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe, *Really fast syndrome-based hashing*, in *Progress in Cryptology – AFRICACRYPT 2011*, Lecture Notes in Comput. Sci. **6737**, 134–152, (2011).
11. D. A. Wagner, *A generalized birthday problem*, in *Advances in Cryptology – CRYPTO 2002*, Lecture Notes in Comput. Sci. **2442**, 288–303, (2002).
12. D. Augot, M. Finiasz, and N. Sendrier, *A family of fast syndrome based cryptographic hash functions*, in *Progress in Cryptology – Mycrypt 2005*, Lecture Notes in Comput. Sci. **3715**, 64–83, (2005).
13. D. Stehlé and R. Steinfeld, *Making NTRU as secure as worst-case problems over ideal lattices*, in *Advances in Cryptology – EUROCRYPT 2011*, Lecture Notes in Comput. Sci. **6632**, 27–47, (2011).

14. E. Prange, *The use of information sets in decoding cyclic codes*, IRE Trans. Inform. Theory **8**, 5–9, (1962).

15. E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. Inform. Theory **24**, 384–386, (1978).

16. E. Kirshanova, *Improved quantum information set decoding*, in *Post-Quantum Cryptography – PQCrypto 2018*, Lecture Notes in Comput. Sci. **10786**, 507–527, (2018).

17. G. L. Mullen and D. Panario (eds.), *Handbook of Finite Fields*, CRC Press, Boca Raton, (2013).

18. G. Kachigar and J.-P. Tillich, *Quantum information set decoding algorithms*, in *Post-Quantum Cryptography – PQCrypto 2017*, Lecture Notes in Comput. Sci. **10346**, 69–89, (2017).

19. H. Q. Dinh and S. R. López-Permouth, *Cyclic and negacyclic codes over finite chain rings*, IEEE Trans. Inform. Theory **50**, 1728–1744, (2004).

20. H. Q. Dinh and S. R. López-Permouth, *Constacyclic codes over finite commutative chain rings*, in *Concise Encyclopedia of Coding Theory*, Chapman and Hall/CRC, 385–428, (2021).

21. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 2nd ed., Cambridge Univ. Press, Cambridge, (2003).

22. J. Kaboré and M. E. Charkani, *Constacyclic codes over* $\mathbb{F}_q + u\mathbb{F}_q + v\mathbb{F}_q + uv\mathbb{F}_q$, arXiv:1507.03084, (2015).

23. L. Dallot, *Towards a concrete security proof of Courtois, Finiasz and Sendrier signature scheme*, in *Research in Cryptology – WEWoRC 2007*, Lecture Notes in Comput. Sci. **4945**, 65–77, (2008).

24. M. Greferath and S. E. Schmidt, *Gray isometries for finite chain rings and a nonlinear ternary* $(36, 3^{12}, 15)$ *code*, IEEE Trans. Inform. Theory **45**, 2522–2524, (1999).

25. National Institute of Standards and Technology, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, FIPS PUB 202, (2015).

26. N. T. Courtois, M. Finiasz, and N. Sendrier, *How to achieve a McEliece-based digital signature scheme*, in *Advances in Cryptology – ASIACRYPT 2001*, Lecture Notes in Comput. Sci. **2248**, 157–174, (2001).

27. P.-A. Fouque and G. Leurent, *Cryptanalysis of a hash function based on quasi-cyclic codes*, in *Topics in Cryptology – CT-RSA 2008*, Lecture Notes in Comput. Sci. **4964**, 19–35, (2008).

28. P. Rogaway and T. Shrimpton, *Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance*, in *Fast Software Encryption – FSE 2004*, Lecture Notes in Comput. Sci. **3017**, 371–388, (2004).

29. R. Durstenfeld, *Algorithm 235: Random permutation*, Commun. ACM **7**, 420, (1964).

30. S. Abdelalim, A. Lkoaiza, A. Cherkaoui, I. Elmouki, and N. Abghour, *A Python programming initiative for hash construction through the example of SHA-2*, in *Finite Abelian Groups, Elliptic Curves, Blockchain With Hashing and Graphs*, 264–278, (2025).

31. S. Abdelalim, A. Cherkaoui, A. Lkoaiza, I. Elmouki, and N. Abghour, *Advancing blockchain security using graph theory: A Python programming perspective*, in *Finite Abelian Groups, Elliptic Curves, Blockchain With Hashing and Graphs*, 279–293, (2025).

32. T. W. Judson, *Abstract Algebra: Theory and Applications*, (2020).

33. T. Debris-Alazard, N. Sendrier, and J.-P. Tillich, *Wave: A new family of trapdoor one-way preimage sampleable functions based on codes*, in *Advances in Cryptology – ASIACRYPT 2019, Part I*, Lecture Notes in Comput. Sci. **11921**, 21–51, (2019).

34. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen, *SWIFFT: A modest proposal for FFT hashing*, in *Fast Software Encryption – FSE 2008*, Lecture Notes in Comput. Sci. **5086**, 54–72, (2008).

Asmaa CHERKAOUI,

Laboratory of Mathematical Analysis, Algebra and Applications (LAM2A),

Faculty of Sciences Ain Chock (FSAC)

,University Hassan II of Casablanca,

Casablanca, Morocco.

E-mail address: esma1maysan@gmail.com

and

Seddik ABDELALIM,

Laboratory of Mathematical Analysis, Algebra and Applications (LAM2A),

Faculty of Sciences Ain Chock (FSAC)

,University Hassan II of Casablanca,

Casablanca, Morocco.

E-mail address: seddikabd@hotmail.com