Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes

No Author Given

Abstract. We present here a new family of trapdoor one-way functions that are Preimage Sampleable on Average (PSA) based on codes, the Wave-PSA family. The trapdoor function is one-way under two computational assumptions: the hardness of generic decoding for high weights and the indistinguishability of generalized $(U, U+V)$-codes. Our proof follows the GPV strategy [28]. By including rejection sampling, we ensure the proper distribution for the trapdoor inverse output. The domain sampling property of our family is ensured by using and proving a variant of the left-over hash lemma. We instantiate the new Wave-PSA family with ternary generalized $(U, U+V)$-codes to design a "hash-and-sign" signature scheme which achieves *existential unforgeability under adaptive chosen message attacks* (EUF-CMA) in the random oracle model. For 128 bits of classical security, signature sizes are in the order of 13 thousand bits, the public key size in the order of 3 megabytes, and the rejection rate is limited to one rejection every 10 to 12 signatures.

1 Introduction

Code-Based Signature Schemes. It is a long standing open problem to build an efficient and secure digital signature scheme based on the hardness of decoding a linear code which could compete with widespread schemes like DSA or RSA. Those signature schemes are well known to be broken by quantum computers and code-based schemes could indeed provide a valid quantum resistant replacement. A first answer to this question was given by the CFS scheme proposed in [15]. It consisted in finding parity-check matrices $\mathbf{H} \in \mathbb{F}_2^{r \times n}$ such that the solution $\mathbf{e}$ of smallest weight of the equation

$$\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}. \tag{1}$$

could be found for a non-negligible proportion of all $\mathbf{s}$ in $\mathbb{F}_2^r$. This task was achieved by using high rate Goppa codes. This signature scheme has however two drawbacks: (i) for high rates Goppa codes the indistinguishability assumption used in its security proof has been invalidated in [21], (ii) security scales only weakly superpolynomially in the keysize for polynomial time signature generation. A crude extrapolation of parallel CFS [22] and its implementations [36, 10] yields for 128 bits of classical security a public key size of several gigabytes and a signature time of several seconds. Those figures even grow to terabytes and hours for quantum-safe security levels, making the scheme unpractical.

This scheme was followed by other proposals using other code families such as for instance [4, 29, 37]. All of them were broken, see for instance [44, 42].

Other signature schemes based on codes were also given in the literature such as for instance the KKS scheme [35, 34], its variants [7, 27] or the RaCoSS proposal [25] to the NIST. But they can be considered at best to be one-time signature schemes and great care has to be taken to choose the parameters of these schemes in the light of the attacks given in [13, 43, 32]. Finally, another possibility is to use the Fiat-Shamir heuristic. For instance by turning the Stern zero-knowledge authentication scheme [49] into a signature scheme but this leads to rather large signature lengths (hundred(s) of kilobits). There has been some recent progress in this area for another metric, namely the rank metric. A hash and sign signature scheme was proposed, RankSign [26], that enjoys remarkably small key sizes, but it got broken too in [19]. On the other hand, following the Schnorr-Lyubashevsky [38] approach, a new scheme was recently proposed, namely Durandal [2]. This scheme enjoys small key sizes and managed to meet the challenge of adapting the Lyubashevsky [39] approach for code-based cryptography. However, there is a lack of genericity in its security reduction, the security of Durandal is reduced to a rather convoluted problem, namely PSSI^{+} (see [2, §4.1]), capturing the problem of using possibly information leakage in the signatures to break the secret key. This is due to the fact that it is not proven in their scheme that their signatures do not leak information.

One-Way Preimage Sampleable Trapdoor Functions. There is a very powerful tool for building a hash-and-sign signature scheme. It is based on the notion of *one-way trapdoor preimage sampleable function* [28, §5.3]. Roughly speaking, this is a family of trapdoor one-way functions $(f_a)_a$ such that with overwhelming probability over the choice of f_a (i) the distribution of the images $f_a(e)$ is very close to the uniform distribution over its range (ii) the distribution of the output of the trapdoor algorithm inverting f_a samples from all possible preimages in an appropriate way. This trapdoor inversion algorithm should namely sample for any x in the output domain of f_a its outputs e such that the distribution of e is indistinguishable in a statistical sense from the input distribution to f_a conditioned on $f_a(e) = x$. This notion and its lattice-based instantiation allowed in [28] to give a full-domain hash (FDH) signature scheme with a tight security reduction based on lattice assumptions, namely that the Short Integer Solution (SIS) problem is hard on average. Furthermore, this approach also allowed to build the first identity based encryption scheme that could be resistant to a quantum computer. We will call in this paper, this approach for obtaining a FDH scheme, the GPV strategy (the authors of [28] are namely Gentry, Peikert and Vaikuntanathan). This strategy has also been adopted in Falcon [24], a lattice based signature submission to the NIST call for post-quantum cryptographic primitives that was recently selected as a second round candidate.

This preimage sampleable primitive is notoriously difficult to obtain when the functions f_a are not trapdoor permutations but many-to-one functions. This is typically the case when one wishes quantum resistant primitives based on lattice based assumptions. The reason is the following. The hard problem on which this primitive relies is the SIS problem where we want to find for a matrix $\mathbf{A}$ in $\mathbb{Z}_q^{n\times m}$

(with $m \geq n$) and an element $\mathbf{s} \in \mathbb{Z}_q^n$ a short enough (for the Euclidean norm) solution $\mathbf{e} \in \mathbb{Z}_q^m$ to the equation

$$\mathbf{e}\mathbf{A}^{\intercal} = \mathbf{s} \mod q. \tag{2}$$

$\mathbf{A}$ defines a preimage sampleable function as $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{e}\mathbf{A}^{\intercal}$ and the input to $f_{\mathbf{A}}$ is chosen according to a (discrete) Gaussian distribution of some variance σ^2. Obtaining a nearly uniform distribution for the $f_{\mathbf{A}}(\mathbf{e})$'s over its range requires to choose σ^2 so large so that there are actually *exponentially many* solutions to (2). It is a highly non-trivial task to build in this case a trapdoor inversion algorithm that samples appropriately among all possible preimages, i.e. that is oblivious of the trapdoor.

The situation is actually exactly the same if we want to use another candidate problem for building this preimage sampleable primitive for being resistant to a quantum computer, namely the decoding problem in code-based cryptography. Here we rely on the difficulty of finding a solution $\mathbf{e}$ of Hamming weight *exactly* w with coordinates in a finite field $\mathbb{F}_q$ for the equation

$$\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}. \tag{3}$$

where $\mathbf{H}$ is a given matrix and $\mathbf{s}$ (usually called a syndrome) a given vector with entries in $\mathbb{F}_q$. The weight w has to be chosen large enough so that this equation has always exponentially many solutions (in n the length of $\mathbf{e}$). As in the lattice based setting, it is non-trivial to build trapdoor candidates with a trapdoor inversion algorithm for $f_{\mathbf{H}}$ (defined as $f_{\mathbf{H}}(\mathbf{e}) = \mathbf{e}\mathbf{H}^{\intercal}$) that is oblivious of the trapdoor.

Our Contribution: a Code-Based PSA Family and a FDH Scheme. Our main contribution is to give here a code-based one way trapdoor function that meets the preimage sampleable property in a slighty relaxed way: it meets this property on average. We call such a function Premiage Sampleable on Average, PSA in short. This property on average turns out to be enough for giving a security proof for the signature scheme built from it. Our family relies here on the difficulty of solving (3). We derive from it a FDH signature scheme which is shown to be existentially unforgeable under a chosen-message attack (EUF-CMA) with a tight reduction to solving two code-based problems: one is a distinguishing problem related to the trapdoor used in our scheme, the other one is a multiple targets version of the decoding problem (3), the so called "Decoding One Out of Many" problem (DOOM in short) [46]. In [28] a signature scheme based on preimage sampleable functions is given that is shown to be strongly existentially unforgeable under a chosen-message attack if in addition the preimage sampleable functions are also collision resistant. With our choice of w and $\mathbb{F}_q$, our preimage sampleable functions are not collision resistant. However, as observed in [28], collision resistance allows a tight security reduction but is not necessary: a security proof could also be given when the function is "only" preimage sampleable. Here we will show that it is even enough to have such a property on average. Moreover, contrarily to the lattice setting where the size of the alphabet q grows with n, our alphabet size will be constant in our proposal, it is fixed to $q = 3$.

Our Trapdoor: Generalized $(U, U+V)$-Codes. In [28] the trapdoor consists in a short basis of the lattice considered in the construction. Our trapdoor will be of a different nature, it consists in choosing parity-check matrices of generalized $(U, U + V)$-codes. In our construction, U and V are chosen as random codes. The number of such generalized $(U, U + V)$-codes of dimension k and length n is of the same order as the number of linear codes with the same parameters, namely $q^{\Theta\left(n^2\right)}$ when $k = \Theta\left(n\right)$. A generalized $(U, U+V)$ code $\mathcal{C}$ of length n over $\mathbb{F}_q$ is built from two codes U and V of length $n/2$ and 4 vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ in $\mathbb{F}_q^{n/2}$ as the following "mixture" of U and V:

$$\mathcal{C} = \{(\mathbf{a} \odot \mathbf{u} + \mathbf{b} \odot \mathbf{v}, \mathbf{c} \odot \mathbf{u} + \mathbf{d} \odot \mathbf{v}) : \mathbf{u} \in U,\ \mathbf{v} \in V\}$$

where $\mathbf{x}\odot\mathbf{y}$ stands here for the component-wise product, also called the Hadamard or Schur product. It is defined as: $\mathbf{x} \odot \mathbf{y} \overset{\triangle}{=} (x_1 y_1, \cdots, x_{n/2} y_{n/2})$. Standard $(U, U + V)$-codes correspond to $\mathbf{a} = \mathbf{c} = \mathbf{d} = \mathbf{1}_{n/2}$ and $\mathbf{b} = \mathbf{0}_{n/2}$, the all-one and the all-zero vectors respectively.

The point of introducing such codes is that they have a natural decoding algorithm $\mathtt{D}_{UV}$ solving the decoding problem (3) that is based on a generic decoding algorithm $\mathtt{D}_{\text{gen}}$ for linear codes. $\mathtt{D}_{\text{gen}}$ will be here a very simple decoder, namely a variation of the Prange decoder [45] that is able to produce for *any* parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{r\times n}$ at will a solution of (3) when w is in the range $[\![\frac{q-1}{q} r, n - \frac{r}{q}]\!]$. Note that this algorithm works in polynomial time and that outside this range of weights, the complexity of the best known algorithms is exponential in n for weights w of the form $w = \omega n$ where ω is a constant that lies outside the interval $[\frac{q-1}{q}\rho, 1 - \frac{\rho}{q}]$ with $\rho \overset{\triangle}{=} \frac{r}{n}$. $\mathtt{D}_{UV}$ works by combining the decoding of V with $\mathtt{D}_{\text{gen}}$ with the decoding of U by $\mathtt{D}_{\text{gen}}$. The nice feature is that $\mathtt{D}_{UV}$ is more powerful than $\mathtt{D}_{\text{gen}}$ applied directly on the generalized $(U, U + V)$-code: the weight of the error produced by $\mathtt{D}_{UV}$ can be made in polynomial time to lie outside the interval $[\![\frac{q-1}{q} r, n - \frac{r}{q}]\!]$. This is in essence the trapdoor of our signature scheme. A tweak in this decoder consisting in performing only a small amount of rejection sampling (with our choice of parameters one rejection every 10 or 12 signatures) allows to obtain solutions that are uniformly distributed over the words of weight w. This is the key for obtaining a PSA family and a signature scheme from it.

Finally, a variation of the proof technique of [28] allows to give a tight security proof of our signature scheme that relies only on the hardness of two problems, namely

Decoding Problem: Solving at least one instance of the decoding problem (1) out of multiple instances for a certain w that is outside the range $[\![\frac{q-1}{q} r, n - \frac{r}{q}]\!]$

Distinguishing Problem: Deciding whether a linear code is a permuted generalized $(U, U + V)$ code or not.

Hardness of the Decoding Problem. All code-based cryptography relies upon that problem. Here we are in a case where there are multiple solutions of (3) and the adversary may produce any number of instances of (3) with the same matrix $\mathbf{H}$ and various syndromes $\mathbf{s}$ and is interested in solving only one of them. This relates to the, so called, Decoding One Out of Many (DOOM) problem. This problem was first considered in [33]. It was shown there how to adapt the known algorithms for decoding a linear code in order to solve this modified problem. This modification was later analyzed in [46]. The parameters of the known algorithms for solving (3) can be easily adapted to this scenario where we have to decode simultaneously multiple instances which all have multiple solutions.

Hardness of the Distinguishing Problem. This problem might seem at first sight to be ad-hoc. However, even in the very restricted case of $(U, U+V)$-codes, deciding whether a code is a permuted $(U, U+V)$-code or not is an NP-complete problem. Therefore the Distinguishing Problem is also NP-complete for generalized $(U, U+V)$-codes. This theorem is proven in the case of binary $(U, U+V)$-codes in [17, §7.1, Thm 3] and the proof carries over to an arbitrary finite field $\mathbb{F}_q$. However as observed in [17, p. 3], these NP-completeness reductions hold in the particular case where the dimensions k_U and k_V of the code U and V satisfy $k_U < k_V$. If we stick to the binary case, i.e. $q = 2$, then in order that our $(U, U+V)$ decoder works outside the integer interval $[\![\frac{r}{2}, n-\frac{r}{2}]\!]$ it is necessary that $k_U > k_V$. Unfortunately in this case there is an efficient probabilistic algorithm solving the distinguishing problem that is based on the fact that in this case the hull of the permuted $(U, U+V)$-code is typically of large dimension, namely $k_U - k_V$ (see [16, §1 p.1-2]). This problem can not be settled in the binary case by considering generalized $(U, U+V)$-codes instead of just plain $(U, U+V)$-codes, since it is only for the restricted class of $(U, U+V)$-codes that the decoder considered in [16] is able to work properly outside the critical interval $[\![\frac{r}{2}, n-\frac{r}{2}]\!]$. This explains why the ancestor Surf [16] of the scheme proposed here that relies on binary $(U, U+V)$-codes can not work.

This situation changes drastically when we move to larger finite fields. In order to have a decoding algorithm $\mathtt{D}_{UV}$ that has an advantage over the generic decoder $\mathtt{D}_{\text{gen}}$ we do not need to have $\mathbf{a} = \mathbf{c} = \mathbf{d} = \mathbf{1}_{n/2}$ and $\mathbf{b} = \mathbf{0}_{n/2}$ (i.e. $(U, U+V)$-codes) we just need that $\mathbf{a} \odot \mathbf{c}$ and $\mathbf{a} \odot \mathbf{d} - \mathbf{b} \odot \mathbf{c}$ are vectors with only non-zero components. This freedom of choice for the $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ thwarts completely the attacks based on hull considerations and changes completely the nature of the distinguishing problem. In this case, it seems that the best approach for solving the distinguishing problem is based on the following observation. The generalized $(U, U+V)$-code has codewords of weight slightly smaller than the minimum distance of a random code of the same length and dimension. It is very tempting to conjecture that the best algorithms for solving the Distinguishing Problem come from detecting such codewords. This approach can be easily thwarted by choosing the parameters of the scheme in such a way that the best algorithms for solving this task are of prohibitive complexity. Notice that the best algorithms that we have for detecting such codewords are in essence precisely the generic algorithms for solving the Decoding Problem. In some sense, it seems that we might rely on the very same problem, namely solving the Decoding Problem, even if our proof technique does not show this.

$q = 3$ and Large weights Decoding. In terms of simplicity of the decoding procedure used in the signing process, it seems that defining our codes over the finite field $\mathbb{F}_3$ is particularly attractive. In such a case, the biggest advantage of $\mathtt{D}_{UV}$ over $\mathtt{D}_{\text{gen}}$ is obtained for large weights rather than for small weights (there is an explanation for this asset in the paragraph *"Why is the trapdoor more powerful for large weights than for small weights?"* §4.3). This is a bit unusual in code-based cryptography to rely on the difficulty of finding solutions of large weight to the decoding problem. However, it also opens the issue whether it would not be advantageous to make certain (non-binary) code-based primitives rely on the hardness of solving the decoding problem for large weights rather than for small weights. Of course these two problems are equivalent in the binary case, i.e. $q = 2$, but this is not the case for larger alphabets anymore and still everything seems to point to the direction that large weights problem is by no means easier than its small weight counterpart.

All in all, this gives the first practical signature scheme based on ternary codes which comes with a security proof and which scales well with the parameters: it can be shown that if one wants a security level of 2^λ, then signature size is of order $O(\lambda)$, public key size is of order $O(\lambda^2)$, signature generation is of order $O(\lambda^3)$, whereas signature verification is of order $O(\lambda^2)$. It should be noted that contrarily to the current thread of research in code-based or lattice-based cryptography which consists in relying on structured codes or lattices based on ring structures in order to decrease the key-sizes we did not follow this approach here. This allows for instance to rely on the NP-complete Decoding Problem which is generally believed to be hard on average rather that on decoding in quasi-cyclic codes for instance whose status is still unclear with a constant number of circulant blocks. Despite the fact that we did not use the standard approach for reducing the key sizes relying on quasi-cyclic codes for instance, we obtain acceptable key sizes (about 3.2 megabytes for 128 bits of security) which compare very favorably to unstructured lattice-based signature schemes such as TESLA for instance [1]. This is due in part to the tightness of our security reduction.

2 Notation

General Notation. The notation $x \overset{\triangle}{=} y$ means that x is defined to be equal to y. We denote by $\mathbb{F}_q$ the finite field with q elements and by $S_{w,n}$, or S_w when n is clear from the context, the subset of $\mathbb{F}_q^n$ of words of weight w. For a and b integers with $a \leq b$, we denote by $[\![a, b]\!]$ the set of integers $\{a, a+1, \dots, b\}$. Furthermore, h_3 will denote the function: $h_3(x) \overset{\triangle}{=} -x \log_3(x) - (1-x) \log_3(1-x)$ defined on $[0, 1]$.

Vector and Matrix Notation. Vectors will be written with bold letters (such as $\mathbf{e}$) and uppercase bold letters are used to denote matrices (such as $\mathbf{H}$). Vectors are in row notation. Let $\mathbf{x}$ and $\mathbf{y}$ be two vectors, we will write $(\mathbf{x}, \mathbf{y})$ to denote their concatenation. We also denote by $\mathbf{x}_{\mathcal{I}}$ the vector whose coordinates are those of $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ which are indexed by $\mathcal{I}$, i.e. $\mathbf{x}_{\mathcal{I}} = (x_i)_{i \in cI}$. We will denote by $\mathbf{H}_{\mathcal{I}}$ the matrix whose columns are those of $\mathbf{H}$ which are indexed by $\mathcal{I}$. Sometimes we denote for a vector $\mathbf{x}$ by $\mathbf{x}(i)$ its i-th entry, or for a matrix $\mathbf{A}$, by $\mathbf{A}(i,j)$ its entry in row i and column j. We define the support of $\mathbf{x} = (x_i)_{1 \leq i \leq n}$ as $\mathrm{Supp}(\mathbf{x}) \overset{\triangle}{=} \{i \in \{1, \cdots, n\}$ such that $x_i \neq 0\}$. The Hamming weight of $\mathbf{x}$ is denoted by $|\mathbf{x}|$. By some abuse of notation, we will use the same notation to denote the size of a finite set: $|S|$ stands for the size of the finite set S. For a vector $\mathbf{a} \in \mathbb{F}_q^n$, we denote by $\mathbf{Diag}(\mathbf{a})$ the $n \times n$ diagonal matrix $\mathbf{A}$ with its entries given by $\mathbf{a}$, i.e. $\mathbf{A}(i,i) = a_i$ for all $i \in [\![1, n]\!]$ and $\mathbf{A}(i,j) = 0$ for $i \neq j$.

Probabilistic Notation. Let S be a finite set, then $x \hookleftarrow S$ means that x is assigned to be a random element chosen uniformly at random in S. For two random variables X, Y, $X \sim Y$ means that X and Y are identically distributed. We will also use the same notation for a random variable and a distribution $\mathcal{D}$, where $X \sim \mathcal{D}$ means that that X is distributed according to $\mathcal{D}$. We denote the uniform distribution on S_w by $\mathcal{U}_w$. The statistical distance between two discrete probability distributions over a same space $\mathcal{E}$ is defined as: $\rho(\mathcal{D}_0, \mathcal{D}_1) \overset{\triangle}{=} \frac{1}{2} \sum_{x \in \mathcal{E}} |\mathcal{D}_0(x) - \mathcal{D}_1(x)|$. Recall that a function $f(n)$ is said to be negligible, and we denote this by $f \in \mathrm{negl}(n)$, if for all polynomials $p(n)$, $|f(n)| < p(n)^{-1}$ for all sufficiently large n.

Coding Theory. For any matrix $\mathbf{M}$ we denote by $\langle \mathbf{M} \rangle$ the vector space spanned by its rows. A q-ary linear code $\mathcal{C}$ of length n and dimension k is a subspace of $\mathbb{F}_q^n$ of dimension k and is often defined by a *parity-check matrix* $\mathbf{H}$ over $\mathbb{F}_q$ of size $r \times n$ as $\mathcal{C} = \langle \mathbf{H} \rangle^{\perp} = \left\{ \mathbf{x} \in \mathbb{F}_q^n : \mathbf{x}\mathbf{H}^{\intercal} = \mathbf{0} \right\}$. When $\mathbf{H}$ is of full rank we have $r = n - k$. The code rate, usually denoted by R, is defined as the ratio k/n. An *information set* of a code $\mathcal{C}$ of length n is a set of k coordinate indices $\mathcal{I} \subset [\![1, n]\!]$ such that its complement indexes $n - k$ independent columns on any parity-check matrix. For any $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$, and any information set $\mathcal{I}$ of $\mathcal{C} = \langle \mathbf{H} \rangle^{\perp}$, for all $\mathbf{x} \in \mathbb{F}_q^n$ there exists a unique $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}$ and $\mathbf{x}_{\mathcal{I}} = \mathbf{e}_{\mathcal{I}}$.

3 The Wave-family of Trapdoor One-Way Preimage Sampleable Functions

3.1 One-way Preimage Sampleable Code-based Functions

In this work we will use the FDH paradigm [9, 14] using as one-way the syndrome function:

$$f_{w,\mathbf{H}} : \mathbf{e} \in S_w \longmapsto \mathbf{e}\mathbf{H}^{\intercal} \in \mathbb{F}_q^{n-k}$$

The corresponding FDH signature uses a trapdoor to choose $\boldsymbol{\sigma} \in f^{-1}_{w,\mathbf{H}}(\mathbf{h})$ where $\mathbf{h}$ is the digest of the message to be signed. Here, the signature domain is S_w and its range is the set of syndromes $\mathbb{F}_q^{n-k}$ according to $\mathbf{H}$, an $(n-k) \times n$ parity check matrix of some q-ary linear $[n, k]$ code. The weight w is chosen such that the one-way function $f_{w,\mathbf{H}}$ is surjective but not bijective. Building a secure FDH signature in this situation can be achieved by imposing additional properties [28] to the one-way function (we will speak of the GPV strategy). This is mostly captured by the notion of Preimage Sampleable Functions, see [28, Definition 5.3.1]. We express below this notion in our code-based context with a slightly relaxed definition dropping the collision resistance condition and only assuming that the preimage sampleable property holds on average and not for any possible element in the function range. This will be sufficient for proving the security of our code-based FDH scheme.

Definition 1 (Trapdoor One-way Preimage Sampleable on Average Code-based Functions). *It is a pair of probabilistic polynomial-time algorithms* (`Trapdoor`, `InvertAlg`) *together with a triple of functions* $(n(\lambda), k(\lambda), w(\lambda))$ *growing polynomially with the security parameter* λ *and giving the length and dimension of the codes and the weights we consider for the syndrome decoding problem, such that*

- `Trapdoor` *when given* λ*, outputs* $(\mathbf{H}, T)$ *where* $\mathbf{H}$ *is an* $(n-k) \times n$ *matrix over* $\mathbb{F}_q$ *and* T *the trapdoor corresponding to* $\mathbf{H}$*.*
- `InvertAlg` *is a probabilistic algorithm which takes as input* T *and an element* $\mathbf{s} \in \mathbb{F}_q^{n-k}$ *and outputs an* $\mathbf{e} \in S_{w,n}$ *such that* $\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}$*.*

The following properties have to hold for all but a negligible fraction of $\mathbf{H}$ *output by* `Trapdoor`*.*

1. Domain Sampling with uniform output:

$$\rho(\mathbf{e}\mathbf{H}^{\intercal}, \mathbf{s}) \in \text{negl}(\lambda), \ \textit{where } \mathbf{e} \hookleftarrow S_{w,n} \textit{ and } \mathbf{s} \hookleftarrow \mathbb{F}_q^{n-k}.$$

2. Preimage Sampling on Average (PSA) with trapdoor:

$$\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}\right) \in \text{negl}(\lambda), \ \textit{where } \mathbf{e} \hookleftarrow S_{w,n} \textit{ and } \mathbf{s} \hookleftarrow \mathbb{F}_q^{n-k}.$$

3. One wayness without trapdoor: *for any probabilistic poly-time algorithm* $\mathcal{A}$ *outputting an element* $\mathbf{e} \in S_{w,n}$ *when given* $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ *and* $\mathbf{s} \in \mathbb{F}_q^{n-k}$*, the probability that* $\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}$ *is negligible, where the probability is taken over the choice of* $\mathbf{H}$*, the target value* $\mathbf{s}$ *chosen uniformly at random, and* $\mathcal{A}$*'s random coins.*

Remark 1. 1. The preimage property as defined in [28] would translate in our setting in the following way. For any $\mathbf{s} \in \mathbb{F}_q^{n-k}$ we should have

$$\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right) \in \text{negl}(\lambda), \ \text{where } \mathbf{e}_s \hookleftarrow \left\{\mathbf{e} \in S_{w,n} : \mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}\right\}.$$

Note that in our case we have

$$\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}\right) = \frac{1}{q^{n-k}} \sum_{\mathbf{s} \in \mathbb{F}_q^{n-k}} \rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right).$$

In other words, we just require that the average of the $\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right)$'s is negligible, whereas [28] requires that all terms $\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right)$ are negligible. Note that our property that holds for the average implies that this property holds for all but a negligible fraction of $\mathbf{s}$'s. Indeed, if we have

$$\frac{1}{q^{n-k}} \sum_{\mathbf{s} \in \mathbb{F}_q^{n-k}} \rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right) = \varepsilon,$$

then

$$\frac{\#\left\{\mathbf{s} : \rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right) \geq \sqrt{\varepsilon}\right\}}{q^{n-k}} \leq \sqrt{\varepsilon}.$$

2. It turns out that this relaxed definition of preimage sampleable function is enough to prove the security of the associated signature scheme using a salt as given in the next paragraph. This relaxed definition is of independent interest, since it can be easier to find trapdoor one-way functions meeting this property than the more stringent definition given in [28].

Given a one-way preimage sampleable code-based function ($\texttt{Trapdoor}$, $\texttt{InvertAlg}$) we easily define a code-based FDH signature scheme as follows. We generate the public/secret key as $(\mathrm{pk}, \mathrm{sk}) = (\mathbf{H}, T) \leftarrow \texttt{Trapdoor}(\lambda)$. We also select a cryptographic hash function $\texttt{Hash} : \{0,1\}^* \to \mathbb{F}_q^{n-k}$ and a salt $\mathbf{r}$ of size λ_0. The algorithms $\texttt{Sgn}^{\mathrm{sk}}$ and $\texttt{Vrfy}^{\mathrm{pk}}$ are defined as follows

```
$\texttt{Sgn}^{\mathrm{sk}}(\mathbf{m})$:
  $\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$
  $\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$
  $\mathbf{e} \leftarrow \texttt{InvertAlg}(\mathbf{s}, T)$
  $\texttt{return}(\mathbf{e}, \mathbf{r})$
```

```
$\texttt{Vrfy}^{\mathrm{pk}}(\mathbf{m}, (\mathbf{e}', \mathbf{r}))$:
  $\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$
  if $\mathbf{e}'\mathbf{H}^{\intercal} = \mathbf{s}$ and $|\mathbf{e}'| = w$ return 1
  else return 0
```

A tight security reduction in the random oracle model is given in [28] for PSF signature schemes. It requires collision resistance. Our construction uses a ternary alphabet $q = 3$ together with large values of w and collision resistance is not met. Still, we achieve a tight security proof by considering in §**??** a reduction to the multiple target decoding problem.

3.2 The Wave Family of PSA Functions

The trapdoor family of codes which gives an advantage for inverting $f_{w,\mathbf{H}}$ is built upon the following transformation:

Definition 2. *Let* $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ *and* $\mathbf{d}$ *be vectors of* $\mathbb{F}_q^{n/2}$. *We define*

$$\varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}} : (\mathbf{x},\mathbf{y}) \in \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} \rightarrow (\mathbf{a} \odot \mathbf{x} + \mathbf{b} \odot \mathbf{y}, \mathbf{c} \odot \mathbf{x} + \mathbf{d} \odot \mathbf{y}) \in \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2}$$

We will say that $\varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}$ *is UV-normalized if*

$$\forall i \in [\![1, n/2]\!], \quad a_i d_i - b_i c_i = 1,\ a_i c_i \neq 0. \tag{4}$$

For any two subspaces U *and* V *of* $\mathbb{F}_q^{n/2}$, *we extend the notation*

$$\varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}(U,V) \overset{\triangle}{=} \{\varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}(\mathbf{u},\mathbf{v}) : \mathbf{u} \in U, \mathbf{v} \in V\}$$

Proposition 1 (Normalized Generalized $(U, U+V)$**-code).** *Let* n *be an even integer and let* $\varphi = \varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}$ *be a UV-normalized mapping. The mapping* φ *is bijective with* $\varphi^{-1}(\mathbf{x},\mathbf{y}) = (\mathbf{d} \odot \mathbf{x} - \mathbf{b} \odot \mathbf{y}, -\mathbf{c} \odot \mathbf{x} + \mathbf{a} \odot \mathbf{y})$.

For any two subspaces U *and* V *of* $\mathbb{F}_q^{n/2}$ *of parity check matrices* $\mathbf{H}_U$ *and* $\mathbf{H}_V$, *the vector space* $\varphi(U,V)$ *is called a* normalized generalized $(U, U+V)$-code. *It has dimension* $\dim U + \dim V$ *and admits the following parity check matrix*

$$\mathcal{H}(\varphi, \mathbf{H}_U, \mathbf{H}_V) \overset{\triangle}{=} \left(\begin{array}{c|c} \mathbf{H}_U\mathbf{D} & -\mathbf{H}_U\mathbf{B} \\ \hline -\mathbf{H}_V\mathbf{C} & \mathbf{H}_V\mathbf{A} \end{array}\right) \tag{5}$$

where $\mathbf{A} \overset{\triangle}{=} \mathbf{Diag}(\mathbf{a})$, $\mathbf{B} \overset{\triangle}{=} \mathbf{Diag}(\mathbf{b})$, $\mathbf{C} \overset{\triangle}{=} \mathbf{Diag}(\mathbf{c})$ *and* $\mathbf{D} \overset{\triangle}{=} \mathbf{Diag}(\mathbf{d})$.

In the sequel, a UV-normalized mapping φ implicitly defines a quadruple of vectors $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ such that $\varphi = \varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}}$. We will use this implicit notation and drop the subscript whenever no ambiguity may arise.

Remark 2.
- This construction can be viewed as taking two codes of length $n/2$ and making a code of length n by "mixing" together a codeword $\mathbf{u}$ in U and a codeword $\mathbf{v}$ in V as the vector formed by the set of $a_i u_i + b_i v_i$'s and $c_i u_i + d_i v_i$'s.
- The condition $a_i c_i \neq 0$ is here to ensure that coordinates of U appear in all the coordinates of the normalized generalized $(U, U+V)$ codeword. This is essential for having a decoding algorithm for the generalized $(U, U+V)$-code that has an advantage over standard information set decoding algorithms for linear codes. The trapdoor of our scheme builds upon this advantage. It can really be viewed as the "interesting" generalization of the standard $(U, U+V)$ construction.
- We have fixed $a_i d_i - b_i c_i = 1$ for every i to simplify some of the expressions in what follows. It is readily seen that any generalized $(U, U+V)$-code that can be obtained in the more general case $a_i d_i - b_i c_i \neq 0$ can also be obtained in the restricted case $a_i d_i - b_i c_i = 1$ by choosing U and V appropriately.

Defining `Trapdoor` and `InvertAlg`. From the security parameter λ, we derive the system parameters n, k, w and split $k = k_U + k_V$ as described in §F.2. The secret key is a tuple sk $= (\varphi, \mathbf{H}_U, \mathbf{H}_V, \mathbf{S}, \mathbf{P})$ where φ is a UV-normalized mapping, $\mathbf{H}_U \in \mathbb{F}_q^{(n/2-k_U)\times n/2}$, $\mathbf{H}_V \in \mathbb{F}_q^{(n/2-k_V)\times n/2}$, $\mathbf{S} \in \mathbb{F}_q^{(n-k)\times(n-k)}$ is nonsingular with $k = k_U + k_V$, and $\mathbf{P} \in \mathbb{F}_q^{n\times n}$ is a permutation matrix. Each element of sk is chosen randomly and uniformly in its domain.

From $(\varphi, \mathbf{H}_U, \mathbf{H}_V)$ we derive the parity check matrix $\mathbf{H}_{\text{sk}} = \mathcal{H}(\varphi, \mathbf{H}_U, \mathbf{H}_V)$ as in Proposition 1. The public key is $\mathbf{H}_{\text{pk}} = \mathbf{S}\mathbf{H}_{\text{sk}}\mathbf{P}$. Next, we need to produce an algorithm $D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}$ which inverts $f_{w,\mathbf{H}_{\text{sk}}}$. The parameter w is such that this can be achieved using the underlying $(U, U+V)$ structure while the generic problem remains hard. In §5 we will show how to use rejection sampling to devise $D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}$ such that its output is uniformly distributed over S_w when $\mathbf{s}$ is uniformly distributed over $\mathbb{F}_q^{n-k}$. This enables us to instantiate algorithm `InvertAlg`. To summarize:

$$\begin{array}{r|l}
\text{sk} \leftarrow (\varphi, \mathbf{H}_U, \mathbf{H}_V, \mathbf{S}, \mathbf{P}) & \quad \texttt{InvertAlg}(\text{sk}, \mathbf{s}) \\
\text{pk} \leftarrow \mathbf{H}_{\text{pk}} \quad\quad\quad & \quad\quad \mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}\left(\mathbf{s}\left(\mathbf{S}^{-1}\right)^{\intercal}\right) \\
(\text{pk}, \text{sk}) \leftarrow \texttt{Trapdoor}(\lambda) & \quad\quad \texttt{return } \mathbf{eP}
\end{array}$$

As in [28], putting this together with a domain sampling condition –which we prove in §6 from a variation of the left-over hash lemma– allows us to define a family of trapdoor preimage sampleable functions, later referred to as the Wave-PSF family.

4 Inverting the Syndrome Function

This section is devoted to the inversion of $f_{w,\mathbf{H}}$ which amounts to solve:

Problem 1 (Syndrome Decoding with fixed weight). Given $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and an integer w, find $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{e}\mathbf{H}^{\intercal} = \mathbf{s}$ and $|\mathbf{e}| = w$.

We consider three nested intervals $[\![w^-_{\text{easy}}, w^+_{\text{easy}}]\!] \subset [\![w^-_{\text{UV}}, w^+_{\text{UV}}]\!] \subset [\![w^-, w^+]\!]$ for w such that for $\mathbf{s}$ randomly chosen in $\mathbb{F}_q^{n-k}$:

- $f^{-1}_{w,\mathbf{H}}(\mathbf{s})$ is likely/very likely to exist if $w \in [\![w^-, w^+]\!]$ (Gilbert-Varshamov bound)
- $\mathbf{e} \in f^{-1}_{w,\mathbf{H}}(\mathbf{s})$ is easy to find if $w \in [\![w^-_{\text{easy}}, w^+_{\text{easy}}]\!]$ for all $\mathbf{H}$ (Prange algorithm)
- $\mathbf{e} \in f^{-1}_{w,\mathbf{H}}(\mathbf{s})$ is easy to find if $w \in [\![w^-_{\text{UV}}, w^+_{\text{UV}}]\!]$ and $\mathbf{H}$ is the parity check matrix of a generalized $(U, U+V)$-code. This is the key for exploiting the underlying $(U, U+V)$ structure as a trapdoor for inverting $f_{w,\mathbf{H}}$.

4.1 Surjective Domain of the Syndrome Function

The issue is here for which value of w we may expect that $f_{w,\mathbf{H}}$ is surjective. This clearly implies that $|S_w| \geq q^{n-k}$. In other words we have:

Fact 1 *If $f_{w,\mathbf{H}}$ is surjective, then $w \in [\![w^-, w^+]\!]$ where $w^- < w^+$ are the extremum of the set $\left\{w \in [\![0, n]\!] \mid \binom{n}{w}(q-1)^w \geq q^{n-k}\right\}$.*

For a fixed rate $R = k/n$, let us define $\omega^- \overset{\triangle}{=} \lim_{n\to+\infty} w^-/n$ and $\omega^+ \overset{\triangle}{=} \lim_{n\to+\infty} w^+/n$. Note that ω^- is known as the asymptotic Gilbert-Varshamov distance. A straightforward computation of the expected number of errors $\mathbf{e}$ of weight w such that $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$ when $\mathbf{H}$ is random shows that we expect an exponential number of solutions when w/n lies in (ω^-, ω^+). However, coding theory has never come up with an efficient algorithm for finding a solution to this problem in the whole range (ω^-, ω^+).

4.2 Easy Domain of the Syndrome Function

The subrange of (ω^-, ω^+) for which we know how to solve efficiently Problem 1 is given by the condition $w/n \in [\omega^-_{\text{easy}}, \omega^+_{\text{easy}}]$ where

$$\omega^-_{\text{easy}} \overset{\triangle}{=} \frac{q-1}{q}(1-R) \quad \text{and} \quad \omega^+_{\text{easy}} \overset{\triangle}{=} \frac{q-1}{q} + \frac{R}{q}, \tag{6}$$

where R is the code rate k/n. This is achieved by a slightly generalized version of the Prange decoder [45]. Prange algorithm is able to complement any word whose coordinates are fixed on an information set into a word of prescribed syndrome. In practice, it outputs in polynomial time using linear algebra, a word $\mathbf{e}$ of prescribed syndrome and of the form $(\mathbf{e}'', \mathbf{e}')$ up to a permutation. The word $\mathbf{e}' \in \mathbb{F}_q^k$ has its support on an information set and can be chosen. The word $\mathbf{e}'' \in \mathbb{F}_q^{n-k}$ is random, thus of average weight $\frac{q-1}{q}(n-k)$. By properly choosing $|\mathbf{e}'|$ the algorithm average output relative weight can thus take any value between $\frac{q-1}{q}\frac{n-k}{n} = \omega^-_{\text{easy}}$ and $k + \frac{q-1}{q}\frac{n-k}{n} = \omega^+_{\text{easy}}$. This procedure, that we call PrangeOne$(\cdot)$, is formalized in Algorithm 1.

```
Algorithm 1 PrangeOne($\mathbf{H}, \mathbf{s}$) — One iteration of the Prange decoder
Parameters: $q, n, k$, $\mathcal{D}$ a distribution over $[\![0, k]\!]$
Require: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$
Ensure: $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$
  $t \hookleftarrow \mathcal{D}$
  $\mathcal{I} \leftarrow$ InfoSet($\mathbf{H}$)        ▷ InfoSet($\mathbf{H}$) returns an information set of $\langle \mathbf{H} \rangle^\perp$
  $\mathbf{x} \hookleftarrow \{\mathbf{x} \in \mathbb{F}_q^n \mid |\mathbf{x}_\mathcal{I}| = t\}$
  $\mathbf{e} \leftarrow$ PrangeStep($\mathbf{H}, \mathbf{s}, \mathcal{I}, \mathbf{x}$)
  return $\mathbf{e}$

function PrangeStep($\mathbf{H}, \mathbf{s}, \mathcal{I}, \mathbf{x}$) — Prange vector completion
Require: $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $\mathcal{I}$ an information set of $\langle \mathbf{H} \rangle^\perp$, $\mathbf{x} \in \mathbb{F}_q^n$
Ensure: $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$ and $\mathbf{e}_\mathcal{I} = \mathbf{x}_\mathcal{I}$
  $\mathbf{P} \leftarrow$ any $n \times n$ permutation matrix sending $\mathcal{I}$ on the last $k$ coordinates
  $(\mathbf{A} \mid \mathbf{B}) \leftarrow \mathbf{H}\mathbf{P}$        ▷ $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times(n-k)}$
  $(\mathbf{0} \mid \mathbf{e}') \leftarrow \mathbf{x}$        ▷ $\mathbf{e}' \in \mathbb{F}_q^k$
  $\mathbf{e} \leftarrow \left(\left(\mathbf{s} - \mathbf{e}'\mathbf{B}^\intercal\right)\left(\mathbf{A}^{-1}\right)^\intercal, \mathbf{e}'\right)\mathbf{P}^\intercal$
  return $\mathbf{e}$
```

Proposition 2. *When* $\mathbf{H}$ *is chosen uniformly at random in* $\mathbb{F}_q^{(n-k)\times n}$ *and* $\mathbf{s}$ *uniformly at random in* $\mathbb{F}_q^{n-k}$*, for the output* $\mathbf{e}$ *of* PrangeOne$(\mathbf{H},\mathbf{s})$ *we have* $|\mathbf{e}| = S + T$ *where* S *and* T *are independent random variables,* $S \in [\![0, n-k]\!]$*,* $T \in [\![0,k]\!]$*,* S *is the Hamming weight of a vector that is uniformly distributed over* $\mathbb{F}_q^{n-k}$ *and* $\mathbb{P}(T = t) = \mathcal{D}(t)$*. Let* $\overline{\mathcal{D}} = \sum_{t=0}^{k} t\mathcal{D}(t)$*, we have:*

$$\mathbb{P}\left(|\mathbf{e}| = w\right) = \sum_{t=0}^{w} \frac{\binom{n-k}{w-t}(q-1)^{w-t}}{q^{n-k}}\mathcal{D}(t), \quad \mathbb{E}(|\mathbf{e}|) = \overline{\mathcal{D}} + \tfrac{q-1}{q}(n-k) = \overline{\mathcal{D}} + n\omega_{easy}^{-}$$

From this proposition, we deduce immediately that any weight w in $[\![\omega_{\text{easy}}^{-}n, \omega_{\text{easy}}^{+}n]\!]$ can be reached by this Prange decoder with a probabilistic polynomial time algorithm that uses a distribution $\mathcal{D}$ such that $\overline{\mathcal{D}} = w - \omega_{\text{easy}}^{-}n$ and which is sufficiently concentrated around its expectation. It will be helpful in what follows to be able to choose a probability distribution $\mathcal{D}$ as this gives a rather large degree of freedom in the distribution of $|\mathbf{e}|$ that will come very handy to simulate an output distribution that is uniform over the words of weight w in the generalized $(U, U+V)$-decoder that we will consider in what follows.

Enlarging the Easy Domain $[\![w_{\text{easy}}^{-}, w_{\text{easy}}^{+}]\!]$. Inverting the syndrome function $f_{w,\mathbf{H}}$ is the basic problem upon which all code-based cryptography relies. This problem has been studied for a long time for relative weights $\omega \overset{\triangle}{=} \frac{w}{n}$ in $(0, \omega_{\text{easy}}^{-})$ and despite many efforts the best algorithms [48, 20, 6, 40, 8, 41, 18, 11] for solving this problem are all exponential in n for such fixed relative weights. In other words, after more than fifty years of research, none of those algorithms came up with a polynomial complexity for relative weights ω in $(0, \omega_{\text{easy}}^{-})$. Furthermore, by adapting all the previous algorithms beyond this point we observe for them the same behaviour: they are all polynomial in the range of relative weights $[\omega_{\text{easy}}^{-}, \omega_{\text{easy}}^{+}]$ and become exponential once again when ω is in $(\omega_{\text{easy}}^{+}, 1)$. All these results point towards the fact that inverting $f_{w,\mathbf{H}}$ in polynomial time on a larger range is fundamentally a hard problem.

4.3 Solution with Trapdoor

Let us recall that our trapdoor to invert $f_{w,\mathbf{H}}$ is given by the family of normalized generalized $(U, U+V)$-codes (Proposition 1 in §3.2). As we will see, this family comes with a simple procedure which enables to invert $f_{w,\mathbf{H}}$ with errors of weight which belongs to $[\![w_{\text{UV}}^{-}, w_{\text{UV}}^{+}]\!] \subset [\![w^{-}, w^{+}]\!]$ but with $[\![w_{\text{easy}}^{-}, w_{\text{easy}}^{+}]\!] \subsetneq [\![w_{\text{UV}}^{-}, w_{\text{UV}}^{+}]\!]$. We summarize this situation in Figure 1. We wish to point out here, to avoid any misunderstanding that the procedure we give here is not the one we use at the end to instantiate Wave, but is merely here to give the underlying idea of the trapdoor. Rejection sampling will be needed as explained in the following section to avoid any information leakage on the trapdoor coming from the outputs of the algorithm given here. It turns out that in the case of a normalized generalized $(U, U+V)$-code, a simple tweak of the Prange decoder will be able

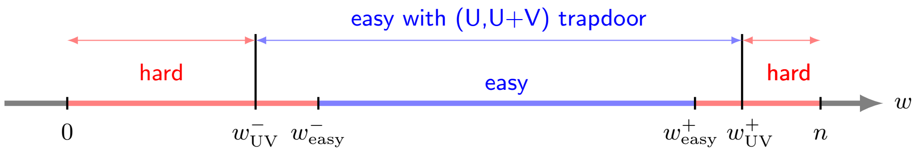

Fig. 1. Hardness of $(U, U+V)$ Decoding

to reach relative weights w/n outside the "easy" region $[\omega^-_{\mathrm{easy}}, \omega^+_{\mathrm{easy}}]$. It exploits the fundamental leverage of the Prange decoder : it consists in choosing the error $\mathbf{e}$ satisfying $\mathbf{eH}^\intercal = \mathbf{s}$ as we want in k positions when the code that we decode is random and of dimension k. When we want an error of low weight, we put zeroes on those positions, whereas if we want an error of large weight, we put non-zero values. This idea leads to even smaller or larger weights in the case of a normalized generalized $(U, U+V)$-code. To explain this point, recall that we want to solve the following decoding problem in this case.

Problem 2 (decoding problem for normalized generalized $(U, U+V)$-codes). Given a normalized generalized $(U, U+V)$ code $(\varphi, \mathbf{H}_U, \mathbf{H}_V)$ (see Proposition 1) of parity-check matrix $\mathbf{H} = \mathcal{H}(\varphi, \mathbf{H}_U, \mathbf{H}_V) \in \mathbb{F}_q^{(n-k)\times n}$, and a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, find $\mathbf{e} \in \mathbb{F}_q^n$ of weight w such that $\mathbf{eH}^\intercal = \mathbf{s}$.

The following notation will be very useful to explain how we solve this problem.

Notation 1 *For a vector $\mathbf{e}$ in $\mathbb{F}_q^n$, we denote by $\mathbf{e}_U$ and $\mathbf{e}_V$ the vectors in $\mathbb{F}_q^{n/2}$ such that $(\mathbf{e}_U, \mathbf{e}_V) = \varphi^{-1}(\mathbf{e})$.*

The decoding algorithm will recover $\mathbf{e}_V$ and then $\mathbf{e}_U$. From $\mathbf{e}_U$ and $\mathbf{e}_V$ we recover $\mathbf{e}$ since $\mathbf{e} = \varphi(\mathbf{e}_U, \mathbf{e}_V)$. The point of introducing such an $\mathbf{e}_U$ and a $\mathbf{e}_V$ is that

Proposition 3. *Solving the decoding problem 2 is equivalent to find an $\mathbf{e} \in \mathbb{F}_q^n$ of weight w satisfying*

$$\mathbf{e}_U \mathbf{H}_U^\intercal = \mathbf{s}^U \text{ and } \mathbf{e}_V \mathbf{H}_V^\intercal = \mathbf{s}^V \tag{7}$$

where $\mathbf{s} = (\mathbf{s}^U, \mathbf{s}^V)$ with $\mathbf{s}^U \in \mathbb{F}_q^{n/2-k_U}$ and $\mathbf{s}^V \in \mathbb{F}_q^{n/2-k_V}$.

Remark 3. We have put U and V as superscripts in $\mathbf{s}^U$ and $\mathbf{s}^V$ to avoid any confusion with the notation we have just introduced for $\mathbf{e}_U$ and $\mathbf{e}_V$.

Proof. Let us observe that $\mathbf{e} = \varphi(\mathbf{e}_U, \mathbf{e}_V) = (\mathbf{a} \odot \mathbf{e}_U + \mathbf{b} \odot \mathbf{e}_V, \mathbf{c} \odot \mathbf{e}_U + \mathbf{d} \odot \mathbf{e}_V) = (\mathbf{e}_U \mathbf{A} + \mathbf{e}_V \mathbf{B}, \mathbf{e}_U \mathbf{C} + \mathbf{e}_V \mathbf{D})$ with $\mathbf{A} = \mathbf{Diag}(\mathbf{a}), \mathbf{B} = \mathbf{Diag}(\mathbf{b}), \mathbf{C} = \mathbf{Diag}(\mathbf{c}), \mathbf{D} = \mathbf{Diag}(\mathbf{d})$. By using this, $\mathbf{eH}^\intercal = \mathbf{s}$ translates into

$$\begin{cases} \mathbf{e}_U \mathbf{AD}^\intercal \mathbf{H}_U^\intercal + \mathbf{e}_V \mathbf{BD}^\intercal \mathbf{H}_U^\intercal - \mathbf{e}_U \mathbf{CB}^\intercal \mathbf{H}_U^\intercal - \mathbf{e}_V \mathbf{DB}^\intercal \mathbf{H}_U^\intercal & = \mathbf{s}^U \\ -\mathbf{e}_U \mathbf{AC}^\intercal \mathbf{H}_V^\intercal - \mathbf{e}_V \mathbf{BC}^\intercal \mathbf{H}_V^\intercal + \mathbf{e}_U \mathbf{CA}^\intercal \mathbf{H}_V^\intercal + \mathbf{e}_V \mathbf{DA}^\intercal \mathbf{H}_V^\intercal & = \mathbf{s}^V \end{cases}$$

which amounts to $\mathbf{e}_U(\mathbf{AD} - \mathbf{BC})\mathbf{H}_U^\intercal = \mathbf{s}^U$ and $\mathbf{e}_V(\mathbf{AD} - \mathbf{BC})\mathbf{H}_V^\intercal = \mathbf{s}^V$, since $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are diagonal matrices, they are therefore symmetric and commute with each other. We finish the proof by observing that $\mathbf{AD} - \mathbf{BC} = \mathbf{I}_{n/2}$. □

Performing the two decoding in (7) independently with the Prange algorithm gains nothing. However if we first solve in V with the Prange algorithm, and then seek a solution in U which properly depends on $\mathbf{e}_V$ we increase the range of weights accessible in polynomial time for $\mathbf{e}$. It then turns out that the range $[\omega_{\text{UV}}^-, \omega_{\text{UV}}^+]$ of relative weights w/n for which the $(U, U+V)$-decoder works in polynomial time is larger than $[\omega_{\text{easy}}^-, \omega_{\text{easy}}^+]$. This will provide an advantage to the trapdoor owner.

Tweaking the Prange Decoder for Reaching Large Weights. When $q = 2$, small and large weights play a symmetrical role. This is not the case anymore for $q \geq 3$. In what follows we will suppose that $q \geq 3$. In order to find a solution $\mathbf{e}$ of large weight to the decoding problem $\mathbf{eH}^\intercal = \mathbf{s}$, we use Proposition 3 and first find an arbitrary solution $\mathbf{e}_V$ to $\mathbf{e}_V{\mathbf{H}_V}^\intercal = \mathbf{s}^V$. The idea, now for performing the second decoding $\mathbf{e}_U{\mathbf{H}_U}^\intercal = \mathbf{s}^U$, is to take advantage of $\mathbf{e}_V$ to find a solution $\mathbf{e}_U$ that maximizes the weight of $\mathbf{e} = \varphi(\mathbf{e}_U, \mathbf{e}_V)$. On any information set of the U code, we can fix arbitrarily $\mathbf{e}_U$. Such a set is of size k_U and on those positions i we can always choose $\mathbf{e}_U(i)$ such that this induces *simultaneously* two positions in $\mathbf{e}$ that are non-zero. These are $\mathbf{e}_i$ and $\mathbf{e}_{i+n/2}$. We just have to choose $\mathbf{e}_U(i)$ so that we have simultaneously $a_i\mathbf{e}_U(i) + b_i\mathbf{e}_V(i) \neq 0$ and $c_i\mathbf{e}_U(i) + d_i\mathbf{e}_V(i) \neq 0$. This is always possible since $q \geq 3$ and it gives an expected weight of $\mathbf{e}$:

$$\mathbb{E}(|\mathbf{e}|) = 2\left(k_U + \frac{q-1}{q}(n/2 - k_U)\right) = \frac{q-1}{q}n + \frac{2k_U}{q} \tag{8}$$

The best choice for k_U is to take $k_U = k$ up to the point where $\frac{q-1}{q}n + \frac{2k}{q} = n$, that is $k = n/2$ and for larger values of k we choose $k_U = n/2$ and $k_V = k - k_U$.

Why Is the Trapdoor More Powerful for Large Weights than for Small Weights? This strategy can be clearly adapted for small weights. However, it is less powerful in this case. Indeed, to minimize the final error weight we would like to choose $\mathbf{e}_U(i)$ in k_U positions such that $a_i\mathbf{e}_U(i) + b_i\mathbf{e}_V(i) = 0$ and $c_i\mathbf{e}_U(i) + d_i\mathbf{e}_V(i) = 0$. Here as $a_id_i - b_ic_i = 1$ and $a_ic_i \neq 0$ in the family of codes we consider, this is possible if and only if $\mathbf{e}_V(i) = 0$. Therefore, contrarily to the case where we want to reach errors of large weight, the area of positions where we can gain twice is constrained to be of size $n/2 - |\mathbf{e}_V|$. The minimal weight for $\mathbf{e}_V$ we can reach in polynomial time with the Prange decoder is given by $\frac{q-1}{q}(n/2 - k_V)$. In this way the set of positions where we can double the number of 0 will be of size $n/2 - \frac{q-1}{q}(n/2 - k_V) = \frac{n}{2q} + \frac{q-1}{q}k_V$. It can be verified that this strategy would give the following expected weight for the final error we get:

$$\mathbb{E}(|\mathbf{e}|) = \frac{q-1}{q}n - 2\frac{q-1}{q}k_U \quad \text{if } k_U \leq \frac{n}{2q} + \frac{q-1}{q}k_V \quad \text{and} \quad \frac{2(q-1)^2}{(2q-1)q}(n-k) \quad \text{else.}$$

5 Preimage Sampling with Trapdoor: Achieving a Uniformly Distributed Output

We restrict our study to $q = 3$ but it can be generalized to larger q. To be a trapdoor one-way preimage sampleable function, we have to enforce that the outputs of our algorithm, which inverts our trapdoor function, are very close to be uniformly distributed over S_w. The procedure described in the previous section using directly the Prange decoder, does not meet this property. As we will prove, by changing it slightly, we will achieve this task by still keeping the property to output errors of weight w for which it is hard to solve the decoding problem for this weight. However, the parameters will have to be chosen carefully and the area of weights w for which we can output errors in polynomial time decreases. Figure 2 gives a rough picture of what will happen.

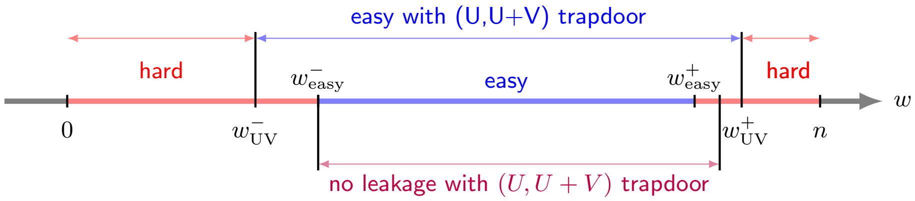

Fig. 2. Hardness of $(U, U+V)$ Decoding with no leakage of signature

5.1 Rejection Sampling to reach Uniformly Distributed Output

We will tweak slightly the generalized $(U, U+V)$-decoder from the previous section by performing in particular rejection sampling on $\mathbf{e}_U$ and $\mathbf{e}_V$ in order to obtain an error $\mathbf{e}$ satisfying $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$ that is uniformly distributed over the words of weight w when the syndrome $\mathbf{s}$ is randomly chosen in $\mathbb{F}_3^{n-k}$. Solving the decoding problem 2 of the generalized $(U, U+V)$-code will be done by solving (7) through an algorithm whose skeleton is given in Algorithm 2. $\textsc{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$ returns a vector satisfying $\mathbf{e}_V {\mathbf{H}_V}^\intercal = \mathbf{s}^V$, whereas $\textsc{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$ returns a vector satisfying $\mathbf{e}_U {\mathbf{H}_U}^\intercal = \mathbf{s}^U$ *and* such that $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$. Here $\mathbf{s} = (\mathbf{s}^U, \mathbf{s}^V)$ with $\mathbf{s}^U \in \mathbb{F}_3^{n/2-k_U}$ and $\mathbf{s}^V \in \mathbb{F}_3^{n/2-k_V}$.

Algorithm 2 $\textsc{DecodeUV}(\mathbf{H}_V, \mathbf{H}_U, \varphi, \mathbf{s})$

```
repeat
    $\mathbf{e}_V \leftarrow \textsc{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$
until Condition 1 is met
repeat
    $\mathbf{e}_U \leftarrow \textsc{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$        ▷ We assume that $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$ here.
    $\mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$
until Condition 2 is met
return $\mathbf{e}$
```

What we want to achieve by rejection sampling is that the distribution of $\mathbf{e}$ output by this algorithm is the same as the distribution of $\mathbf{e}^{\text{unif}}$ that denotes a vector that is chosen uniformly at random among the words of weight w in $\mathbb{F}_3^n$. This will be achieved by ensuring that:

1. the $\mathbf{e}_V$ fed into DecodeU$(\cdot)$ at Step 5 has the same distribution as $\mathbf{e}_V^{\text{unif}}$,
2. the distribution of $\mathbf{e}_U$ surviving to Condition 2 at Step 7 conditioned on the value of $\mathbf{e}_V$ is the same as the distribution of $\mathbf{e}_U^{\text{unif}}$ conditioned on $\mathbf{e}_V^{\text{unif}}$.

There is a property of the decoders DecodeV$(\cdot)$ and DecodeU$(\cdot)$ derived from Prange decoders that we will consider that will be very helpful here.

Definition 3. DecodeV$(\cdot)$ *is said to be weightwise uniform if the output* $\mathbf{e}_V$ *of* DecodeV$(\mathbf{H}_V, \mathbf{s}^V)$ *is such that* $\mathbb{P}(\mathbf{e}_V)$ *is just a function of* $|\mathbf{e}_V|$ *when* $\mathbf{s}^V$ *is chosen uniformly at random in* $\mathbb{F}_3^{n/2-k_V}$. DecodeU$(\cdot)$ *is* m_1*-uniform if the outputput* $\mathbf{e}_U$ *of* DecodeU$(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$ *satisfies that the conditional probability* $\mathbb{P}(\mathbf{e}_U|\mathbf{e}_V)$ *is just a function of the pair* $(|\mathbf{e}_V|, m_1(\varphi(\mathbf{e}_U, \mathbf{e}_V))$ *where*

$$m_1(\mathbf{x}) \stackrel{\triangle}{=} \left|\left\{1 \leq i \leq n/2 : |(x_i, x_{i+n/2})| = 1\right\}\right|.$$

It is readily observed that $\mathbb{P}(\mathbf{e}_V^{\text{unif}})$ and $\mathbb{P}(\mathbf{e}_U^{\text{unif}}|\mathbf{e}_V^{\text{unif}})$ are also only functions of $|\mathbf{e}_V^{\text{unif}}|$ and $(|\mathbf{e}_V^{\text{unif}}|, m_1(\mathbf{e}^{\text{unif}}))$ respectively. From this it is readily seen that we obtain the right distributions for $\mathbf{e}_V$ and $\mathbf{e}_U$ conditioned on $\mathbf{e}_V$ by just ensuring that the distribution of $|\mathbf{e}_V|$ follows the same distribution as $|\mathbf{e}_V^{\text{unif}}|$ and that the distribution of $m_1(\mathbf{e})$ conditioned on $|\mathbf{e}_V|$ is the same as the distribution of $m_1(\mathbf{e}^{\text{unif}})$ conditioned on $|\mathbf{e}_V^{\text{unif}}|$. This is shown by the following lemma.

Lemma 1. *Let* $\mathbf{e}$ *be the output of Algorithm 2 when* $\mathbf{s}^V$ *and* $\mathbf{s}^U$ *are chosen uniformly at random in* $\mathbb{F}_3^{n/2-k_V}$ *and* $\mathbb{F}_3^{n/2-k_U}$ *respectively. Assume that* DecodeU$(\cdot)$ *is* m_1*-uniform whereas* DecodeV$(\cdot)$ *is weightwise uniform. If for any possible* y *and* z, $|\mathbf{e}_V| \sim |\mathbf{e}_V^{\text{unif}}|$ *and* $\mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y) = \mathbb{P}(m_1(\mathbf{e}^{\text{unif}}) = z \mid |\mathbf{e}_V^{\text{unif}}| = y)$*, then* $\mathbf{e} \sim \mathbf{e}^{\text{unif}}$*. The probabilities are taken here over the choice of* $\mathbf{s}^U$ *and* $\mathbf{s}^V$ *and over the internal coins of* DecodeU$(\cdot)$ *and* DecodeV$(\cdot)$.

Proof. We have for any $\mathbf{x}$ in S_w

$$\begin{aligned}\mathbb{P}(\mathbf{e} = \mathbf{x}) &= \mathbb{P}(\mathbf{e}_U = \mathbf{x}_U \mid \mathbf{e}_V = \mathbf{x}_V)\mathbb{P}(\mathbf{e}_V = \mathbf{x}_V)\\ &= \mathbb{P}(\text{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V) = \mathbf{x}_U \mid \mathbf{e}_V = \mathbf{x}_V)\mathbb{P}(\text{DecodeV}(\mathbf{H}_V, \mathbf{s}^V) = \mathbf{x}_V)\\ &= \frac{\mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y)}{n(y,z)} \frac{\mathbb{P}(|\mathbf{e}_V| = y)}{n(y)} \stackrel{\triangle}{=} P \qquad (9)\end{aligned}$$

where $n(y)$ is the number of vectors of $\mathbb{F}_3^{n/2}$ of weight y and $n(y,z)$ is the number of vectors $\mathbf{e}$ in $\mathbb{F}_3^n$ such that $\mathbf{e}_V = \mathbf{x}_V$ and such that $m_1(\mathbf{e}) = z$ (this last number only depends on $\mathbf{x}_V$ through its weight y). Equation (9) is here a consequence of the weightwise uniformity of DecodeV$(\cdot)$ on one hand and the m_1-uniformity of DecodeU$(\cdot)$ on the other hand. We conclude by noticing that

$$\begin{aligned}P &= \frac{\mathbb{P}(m_1(\mathbf{e}^{\text{unif}}) = z \mid |\mathbf{e}_V^{\text{unif}}| = y)}{n(y,z)} \frac{\mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = y)}{n(y)} \qquad (10)\\ &= \mathbb{P}(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V)\mathbb{P}(\mathbf{e}_V^{\text{unif}} = \mathbf{x}_V) = \mathbb{P}(\mathbf{e}^{\text{unif}} = \mathbf{x})\end{aligned}$$

Equation (10) follows from the assumptions on the distribution of $|\mathbf{e}_V|$ and of the conditional distribution of $m_1(\mathbf{e})$ for a given weight $|\mathbf{e}_V|$. □

This shows that in order to obtain that $\mathbf{e}$ is uniformly distributed over S_w it is enough to perform rejection sampling based on the weight $|\mathbf{e}_V|$ for DecodeV$(\cdot)$ and based on the pair $(|\mathbf{e}_V|, m_1(\mathbf{e}))$ for DecodeU$(\cdot)$. In other words, our decoding algorithm with rejection sampling will use a rejection vector $\mathbf{r}_V$ on the weights of $\mathbf{e}_V$ for DecodeV$(\cdot)$ and a two-dimensional rejection vector $\mathbf{r}_U$ for the values of $(|\mathbf{e}_V|, m_1(\mathbf{e}))$ for DecodeU$(\cdot)$. The corresponding algorithm is specified in Algorithm 3. Standard results on rejection sampling yield the following proposition:

Algorithm 3 DecodeUV$(\mathbf{H}_V, \mathbf{H}_U, \varphi, \mathbf{s})$

```
repeat
    $\mathbf{e}_V \leftarrow$ DecodeV$(\mathbf{H}_V, \mathbf{s}^V)$
until rand$([0,1]) \leq \mathbf{r}_V(|\mathbf{e}_V|)$
repeat
    $\mathbf{e}_U \leftarrow$ DecodeU$(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$
    $\mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$
until rand$([0,1]) \leq \mathbf{r}_U(|\mathbf{e}_V|, m_1(\mathbf{e}))$
return $\mathbf{e}$
```

Proposition 4. *Let for any $i, t \in [\![0, n/2]\!]$ and $s \in [\![0, t]\!]$*

$$q_1(i) \stackrel{\triangle}{=} \mathbb{P}\left(|\mathbf{e}_V| = i\right) \;\; ; \;\; q_1^{\text{unif}}(i) \stackrel{\triangle}{=} \mathbb{P}\left(|\mathbf{e}_V^{\text{unif}}| = i\right) \tag{11}$$

$$q_2(s,t) \stackrel{\triangle}{=} \mathbb{P}\left(m_1(\mathbf{e}) = s \mid |\mathbf{e}_V| = t\right) \;\; ; \;\; q_2^{\text{unif}}(s,t) \stackrel{\triangle}{=} \mathbb{P}\left(m_1(\mathbf{e}^{\text{unif}}) = s \mid |\mathbf{e}_V^{\text{unif}}| = t\right) \tag{12}$$

$$\mathbf{r}_V(i) \stackrel{\triangle}{=} \frac{1}{M_V^{rs}} \frac{q_1^{\text{unif}}(i)}{q_1(i)} \quad \textit{and} \quad \mathbf{r}_U(s,t) \stackrel{\triangle}{=} \frac{1}{M_U^{rs}(t)} \frac{q_2^{\text{unif}}(s,t)}{q_2(s,t)}$$

with

$$M_V^{rs} \stackrel{\triangle}{=} \max_{0 \leq i \leq n/2} \frac{q_1^{\text{unif}}(i)}{q_1(i)} \quad \textit{and} \quad M_U^{rs}(t) \stackrel{\triangle}{=} \max_{0 \leq s \leq t} \frac{q_2^{\text{unif}}(s,t)}{q_2(s,t)}$$

Then if DecodeV$(\cdot)$ *is weightwise uniform and* DecodeU$(\cdot)$ *is m_1-uniform, the output* $\mathbf{e}$ *of Algorithm 3 satisfies* $\mathbf{e} \sim \mathbf{e}^{\text{unif}}$.

5.2 Application to the Prange Decoder

To instantiate rejection sampling, we have to provide here (i) how DecodeV$(\cdot)$ and DecodeU$(\cdot)$ are instantiated and (ii) how $q_1^{\text{unif}}, q_2^{\text{unif}}, q_1$ and q_2 are computed. Let us begin by the following proposition which gives q_1^{unif} and q_2^{unif}.

Proposition 5. *Let n be an even integer, $w \leq n$, $i, t \leq n/2$ and $s \leq t$ be integers. We have,*

$$q_1^{\text{unif}}(i) = \frac{\binom{n/2}{i}}{\binom{n}{w} 2^{w/2}} \sum_{\substack{p=0 \\ w+p \equiv 0 \mod 2}}^{i} \binom{i}{p} \binom{n/2 - i}{(w+p)/2 - i} 2^{3p/2} \tag{13}$$

$$q_2^{\text{unif}}(s,t) = \frac{\binom{t}{s}\binom{n/2-t}{\frac{w+s}{2}-t}2^{\frac{3s}{2}}}{\sum_p \binom{t}{p}\binom{n/2-t}{\frac{w+p}{2}-t}2^{\frac{3p}{2}}} \textit{ if } w+s \equiv 0 \mod 2 \quad \textit{and} \quad 0 \textit{ else} \tag{14}$$

Algorithm 4 $\text{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$ the Decoder outputting an $\mathbf{e}_V$ such that $\mathbf{e}_V \mathbf{H}_V^{\intercal} = \mathbf{s}^V$.

```
$\mathcal{J}, \mathcal{I} \leftarrow \text{FreeSet}(\mathbf{H}_V)$
$\ell \hookleftarrow \mathcal{D}_V$
$\mathbf{x}_V \hookleftarrow \left\{\mathbf{x} \in \mathbb{F}_3^{n/2} \mid |\mathbf{x}_{\mathcal{J}}| = \ell, \text{Supp}(\mathbf{x}) \subseteq \mathcal{I}\right\}$        ▷ $(\mathbf{x}_V)_{\mathcal{I}\setminus\mathcal{J}}$ is random
$\mathbf{e}_V \leftarrow \text{PrangeStep}(\mathbf{H}_V, \mathbf{s}^V, \mathcal{I}, \mathbf{x}_V)$
return $\mathbf{e}_V$
function FreeSet($\mathbf{H}$)
Require: $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$
Ensure: $\mathcal{I}$ an information set of $\langle \mathbf{H} \rangle^{\perp}$ and $\mathcal{J} \subset \mathcal{I}$ of size $k-d$
repeat
    $\mathcal{J} \hookleftarrow [\![1, n]\!]$ of size $k-d$
until rank $\mathbf{H}_{\overline{\mathcal{J}}} = n-k$
repeat
    $\mathcal{J}' \hookleftarrow [\![1, n]\!] \setminus \mathcal{J}$ of size $d$
    $\mathcal{I} \leftarrow \mathcal{J} \sqcup \mathcal{J}'$
until $\mathcal{I}$ is an information set of $\langle \mathbf{H} \rangle^{\perp}$
return $\mathcal{J}, \mathcal{I}$
```

Algorithms $\text{DecodeV}(\cdot), \text{DecodeU}(\cdot)$ are described in Algorithms 4 and 5. These two algorithms both use the Prange decoder in the same way as we did with the procedure described in §4.3 to reach large weights, except that here we introduced some internal distributions $\mathcal{D}_V$ and the $\mathcal{D}_U^t$'s. These distributions are here to tweak the weight distributions of $\text{DecodeV}(\cdot)$ and $\text{DecodeU}(\cdot)$ in order to reduce the rejection rate. We have:

Proposition 6. *Let n be an even integer, $w \leq n$, $i, t, k_U \leq n/2$ and $s \leq t$ be integers. Let d be an integer, $k'_V \stackrel{\triangle}{=} k_V - d$ and $k'_U \stackrel{\triangle}{=} k_U - d$. Let X_V (resp. X_U^t) be a random variable distributed according to $\mathcal{D}_V$ (resp. $\mathcal{D}_U^t$). We have,*

$$q_1(i) = \sum_{t=0}^{i} \frac{\binom{n/2-k'_V}{i-t}2^{i-t}}{3^{n/2-k'_V}} \mathbb{P}(X_V = t) \tag{15}$$

$$q_2(s,t) = \begin{cases} \sum\limits_{\substack{t+k'_U-n/2 \leq k_{\neq 0} \leq t \\ k_0 \stackrel{\triangle}{=} k'_U - k_{\neq 0}}} \frac{\binom{t-k_{\neq 0}}{s}\binom{n/2-t-k_0}{\frac{w+s}{2}-t-k_0}2^{\frac{3s}{2}}}{\sum_p \binom{t-k_{\neq 0}}{p}\binom{n/2-t-k_0}{\frac{w+p}{2}-t-k_0}2^{\frac{3p}{2}}} \mathbb{P}(X_U^t = k_{\neq 0}) & \textit{if } w \equiv s \text{ mod } 2. \\ 0 & \textit{else} \end{cases} \tag{16}$$

Algorithm 5 DecodeU($\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V$) the U-Decoder outputting an $\mathbf{e}_U$ such that $\mathbf{e}_U \mathbf{H}_U^\intercal = \mathbf{s}^U$ and $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$.

```
$t \leftarrow |\mathbf{e}_V|$
$k_{\neq 0} \hookleftarrow \mathcal{D}_U^t$
$k_0 \leftarrow k'_U - k_{\neq 0}$                                  ▷ $k'_U \stackrel{\triangle}{=} k_U - d$
repeat
    $\mathcal{J}, \mathcal{I} \leftarrow$ FreeSetW($\mathbf{H}_U, \mathbf{e}_V, k_{\neq 0}$)
    $\mathbf{x}_U \hookleftarrow \{\mathbf{x} \in \mathbb{F}_3^{n/2} \mid \forall j \in \mathcal{J},\ \mathbf{x}(j) \notin \{-\frac{b_i}{a_i}\mathbf{e}_V(i), -\frac{d_i}{c_i}\mathbf{e}_V(i)\}$ and $\mathrm{Supp}(\mathbf{x}) \subseteq \mathcal{I}\}$
    $\mathbf{e}_U \leftarrow$ PrangeStep($\mathbf{H}_U, \mathbf{s}^U, \mathcal{I}, \mathbf{x}_U$)
until $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$
return $\mathbf{e}_U$
```

function FreeSetW($\mathbf{H}, \mathbf{x}, k_{\neq 0}$)

```
Require: $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}, \mathbf{x} \in \mathbb{F}_q^n$ and $k_{\neq 0} \in [\![0, k]\!]$.
Ensure: $\mathcal{J}$ and $\mathcal{I}$ an information set of $\langle \mathbf{H} \rangle^\perp$ such that $|\{i \in \mathcal{J} : x_i \neq 0\}| = k_{\neq 0}$ and
    $\mathcal{J} \subset \mathcal{I}$ of size $k - d$.
repeat
    $\mathcal{J}_1 \hookleftarrow \mathrm{Supp}(\mathbf{x})$ of size $k_{\neq 0}$
    $\mathcal{J}_2 \hookleftarrow [\![1, n]\!] \setminus \mathrm{Supp}(\mathbf{x})$ of size $k - d - k_{\neq 0}$.
    $\mathcal{J} \leftarrow \mathcal{J}_1 \sqcup \mathcal{J}_2$
until rank $\mathbf{H}_{\overline{\mathcal{J}}} = n - k$
repeat
    $\mathcal{J}' \hookleftarrow [\![1, n]\!] \setminus \mathcal{J}$ of size $d$
    $\mathcal{I} \leftarrow \mathcal{J} \sqcup \mathcal{J}'$
until $\mathcal{I}$ is an information set of $\langle \mathbf{H} \rangle^\perp$
return $\mathcal{J}, \mathcal{I}$
```

A parameter d is introduced in Proposition 6 and in Algorithms 4 and 5. When $3^d \approx 2^\lambda$ the probability for not being able to complete a set of $k - d$ positions into an information set of an $[n, k]$ code is of order $\frac{1}{2^\lambda}$. In Algorithm 4 (resp. 5) we pick a set of $k_V - d$ (resp. $k_U - d$) random positions. Those positions will be filled with the ad-hoc rule using $\mathcal{D}_V$ (resp. $\mathcal{D}_U^t$). With probability at least $1 - \frac{1}{2^\lambda}$ those sets can be completed with d extra positions to reach an information set. Those d positions are filled randomly. We perform the Prange decoder and also fill the remaining $n/2 - k_V$ (resp. $n/2 - k_U$) positions with random values. Doing things this way will allow us to prove that we are close enough to the two uniformity conditions of Definition 3. This is stated as follows.

Theorem 1. *Let* $\mathbf{e}$ *be the output of Algorithm 3 based on Algorithms 4,5 and* $\mathbf{e}^{\mathrm{unif}}$ *be a uniformly distributed error of weight* w. *We have*

$$\mathbb{P}\left(\rho(\mathbf{e}, \mathbf{e}^{\mathrm{unif}}) > 3^{-d/2}\right) \leq 3^{-d/2}.$$

A much stronger result showing that $\rho(\mathbf{e}, \mathbf{e}^{\mathrm{unif}})$ is typically smaller than $n^2 3^{-d}$ will be given in the appendix. It will be helpful to consider now the following definition.

Definition 4 (Bad and Good Subsets). *Let $d \leq k \leq n$ be integers and $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$. A subset $\mathcal{E} \subseteq [\![1,n]\!]$ of size $k-d$ is defined as a good set for $\mathbf{H}$ if $\mathbf{H}_{\overline{\mathcal{E}}}$ is of full rank where $\overline{\mathcal{E}}$ denotes the complementary of $\mathcal{E}$. Otherwise, $\mathcal{E}$ is defined as a bad set for $\mathbf{H}$.*

The proof of this theorem relies on introducing a variant of the decoder based on variants of the U and V decoders VarDecodeV$(\cdot)$ and VarDecodeU$(\cdot)$ of algorithms DecodeV$(\cdot)$ and DecodeU$(\cdot)$ respectively that work as DecodeV$(\cdot)$ and DecodeU$(\cdot)$ when $\mathcal{J}$ is a good set and depart from it when $\mathcal{J}$ is a bad set. In the later case, the Prange decoder is not used anymore and an error is output that simulates what the Prange decoder would do with the exception that there is no guarantee that the error $\mathbf{e}_V$ that is output by VarDecodeV$(\cdot)$ satisfies $\mathbf{e}_V\mathbf{H}_V = \mathbf{s}_V$ or that the $\mathbf{e}_U$ that is output by VarDecodeU$(\cdot)$ satisfies $\mathbf{e}_U\mathbf{H}_U = \mathbf{s}_U$. The $\mathbf{e}_V$ and $\mathbf{e}_U$ that are output are chosen on the positions of $\mathcal{J}$ as DecodeV() and DecodeU() as would have done it, but the rest of the positions are chosen uniformly at random in $\mathbb{F}_3$. It is clear that in this case

Fact 2 VarDecodeV$(\cdot)$ *is weightwise uniform and* VarDecodeU$(\cdot)$ *is m_1-uniform.*

The point of considering VarDecodeV$(\cdot)$ and VarDecodeU$(\cdot)$ is that they are very good approximations of DecodeV$(\cdot)$ and DecodeU$(\cdot)$ that meet the uniformity conditions that ensure by using Lemma 1 that the output of Algorithm 3 using VarDecodeV$(\cdot)$ and VarDecodeU$(\cdot)$ instead of DecodeV$(\cdot)$ and DecodeU$(\cdot)$ produces an error $\mathbf{e}$ that is uniformly distributed over the words of weight w. The outputs of VarDecodeV$(\cdot)$ and VarDecodeU$(\cdot)$ only differ from the output of DecodeV$(\cdot)$ and DecodeU$(\cdot)$ when a bad set $\mathcal{J}$ is encountered. These considerations can be used to prove the following proposition.

Proposition 7. *Algorithm 3 based on* VarDecodeV$(\cdot)$ *and* VarDecodeU$(\cdot)$ *produces uniformly distributed errors $\mathbf{e}^{\mathrm{unif}}$ of weight w. Let $\mathbf{e}$ be the output of Algorithm 3 with the use of* DecodeV$(\cdot)$ *and* DecodeU$(\cdot)$. *Let J^{unif} be uniformly distributed over the subsets of $[\![1,n/2]\!]$ of size $k_V - d$ whereas $J^{\mathbf{H}_V}$ is uniformly distributed over the same subsets that are good for $\mathbf{H}_V$. Let $I^{\mathrm{unif}}_{\mathbf{x}_V,\ell}$ be uniformly distributed over the subsets of $[\![1,n/2]\!]$ of size $k_U - d$ such that their intersection with $\mathbf{x}_V$ is of size ℓ whereas $I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}$ is the uniform distribution over the same subsets that are good for $\mathbf{H}_U$. We have:*

$$\rho\left(\mathbf{e};\mathbf{e}^{\mathrm{unif}}\right) \leq \rho\left(J^{\mathbf{H}_V};J^{\mathrm{unif}}\right)+\sum_{\mathbf{x}_V,\ell}\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell};I^{\mathrm{unif}}_{\mathbf{x}_V,\ell}\right)\mathbb{P}\left(k_{\neq 0}=\ell \mid \mathbf{e}_V=\mathbf{x}_V\right)\mathbb{P}\left(\mathbf{e}_V^{\mathrm{unif}}=\mathbf{x}_V\right)$$

Proof. The first statement about the output of Algorithm 3 is a direct consequence of Fact 2 and Lemma 1. The proof of the rest of the proposition relies on the following proposition [30, Proposition 8.10]:

Proposition 8. *Let X, Y be two random variables over a common set A. For any randomized function f with domain A using internal coins independent from X and Y, we have:*

$$\rho\left(f(X);f(Y)\right) \leq \rho\left(X;Y\right).$$

Let us define for $\mathbf{x}_V \in \mathbb{F}_3^{n/2}$ and $\mathbf{x}_U \in \mathbb{F}_3^{n/2}$, $p(\mathbf{x}_V) \stackrel{\triangle}{=} \mathbb{P}\left(\mathbf{e}_V = \mathbf{x}_V\right)$, $q(\mathbf{x}_V) \stackrel{\triangle}{=} \mathbb{P}\left(\mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right)$, $p(\mathbf{x}_U|\mathbf{x}_V) \stackrel{\triangle}{=} \mathbb{P}\left(\mathbf{e}_U = \mathbf{x}_U \mid \mathbf{e}_V = \mathbf{x}_V\right)$, $q(\mathbf{x}_U|\mathbf{x}_V) \stackrel{\triangle}{=} \mathbb{P}\left(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right)$. We have,

$$
\begin{aligned}
\rho\left(\mathbf{e}; \mathbf{e}^{\text{unif}}\right) &= \rho\left(\mathbf{e}_U, \mathbf{e}_V; \mathbf{e}_U^{\text{unif}}, \mathbf{e}_V^{\text{unif}}\right) \\
&= \sum_{\mathbf{x}_V, \mathbf{x}_U} |p(\mathbf{x}_V)p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_V)q(\mathbf{x}_U|\mathbf{x}_V)| \\
&= \sum_{\mathbf{x}_V, \mathbf{x}_U} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))p(\mathbf{x}_U|\mathbf{x}_V) + (p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V))q(\mathbf{x}_V)| \\
&\leq \sum_{\mathbf{x}_V, \mathbf{x}_U} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))p(\mathbf{x}_U|\mathbf{x}_V)| + |(p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V)q(\mathbf{x}_V)| \\
&= \sum_{\mathbf{x}_V} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))| + \sum_{\mathbf{x}_V, \mathbf{x}_U} |p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V)|\, q(\mathbf{x}_V) \quad (17)
\end{aligned}
$$

where in the last line we used that $\sum_{\mathbf{x}_U} |p(\mathbf{x}_U|\mathbf{x}_V)| = 1$ for any $\mathbf{x}_V$. Thanks to Proposition 8:

$$
\sum_{\mathbf{x}_V} |p(\mathbf{x}_V) - q(\mathbf{x}_V)| \leq \rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) \tag{18}
$$

as the internal distribution $\mathcal{D}_V$ of DecodeV($\cdot$) is independent of $J^{\mathbf{H}_V}$ and J^{unif}. Let us upper-bound the second term of the inequality. The distribution of $k_{\neq 0}$ is only function of the weight of the vector given as input to DecodeU($\cdot$) or VarDecodeU($\cdot$). Therefore,

$$
\mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V = \mathbf{x}_V\right) = \mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right) \tag{19}
$$

Let us define $p(\mathbf{x}_U|\mathbf{x}_V, \ell) \stackrel{\triangle}{=} \mathbb{P}(\mathbf{e}_U = \mathbf{x}_U \mid k_{\neq 0} = \ell, \mathbf{e}_V = \mathbf{x}_V)$ and $q(\mathbf{x}_U|\mathbf{x}_V, \ell) \stackrel{\triangle}{=} \mathbb{P}(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid k_{\neq 0} = \ell, \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V)$. With this notation we obtain from (19)

$$
p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V) = \sum_{\ell} \left(p(\mathbf{x}_U|\mathbf{x}_V, \ell) - q(\mathbf{x}_U|\mathbf{x}_V, \ell)\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V = \mathbf{x}_V\right) \tag{20}
$$

The internal coins of DecodeU($\cdot$) and VarDecodeU($\cdot$) are independent of $I^{\mathbf{H}_U}_{\mathbf{x}_V, \ell}$ and $I^{\text{unif}}_{\mathbf{x}_V, \ell}$ and by using Proposition 8 we have for any $\mathbf{x}_V$ and ℓ:

$$
\sum_{x_U} |p(\mathbf{x}_U|\mathbf{x}_V, \ell) - q(\mathbf{x}_U|\mathbf{x}_V, \ell)| \leq \rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V, \ell}; I^{\text{unif}}_{\mathbf{x}_V, \ell}\right) \tag{21}
$$

Combining Equations (17), (18), (20) and (21) concludes the proof. □

The expectations of $\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right)$ and $\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V, \ell}; I^{\text{unif}}_{\mathbf{x}_V, \ell}\right)$ are upperbounded by

Lemma 2. *We have*

$$\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) = \frac{\#\{\textit{subsets of } [\![1, n/2]\!] \textit{ of size } k-d \textit{ bad for } \mathbf{H}\}}{\binom{n/2}{k-d}} \tag{22}$$

$$\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right) = \frac{N_{\mathbf{x},\ell}}{\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}} \tag{23}$$

$$\mathbb{E}\left\{\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right)\right\} \leq \frac{3^{-d}}{2} \tag{24}$$

$$\mathbb{E}\left\{\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right)\right\} \leq \frac{3^{-d}}{2\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}},, \tag{25}$$

where $N_{\mathbf{x},\ell}$ *is the number of subsets of* $[\![1, n/2]\!]$ *of size* $k-d$ *such that their intersection with* $\operatorname{Supp}(\mathbf{x})$ *is of size* ℓ *and that are bad for* $\mathbf{H}$.

Proof. (22) (23) follow from the fact that that the statistical distance between the uniform distribution over $[\![1, s]\!]$ and the uniform distribution over $[\![1, t]\!]$ (with $t \geq s$) is equal to $\frac{t-s}{t}$. Let us index from 1 to $\binom{n/2}{k-d}$ the subsets of size $k-d$ of $[\![1, n/2]\!]$ and let X_i be the indicator of the event "the subset of index i is bad". We have

$$N = \sum_{i=1}^{\binom{n/2}{k-d}} X_i. \tag{26}$$

Recall now (for a proof see Lemma 7 in the appendix) that for integers $d < m$:

$$\mathbb{P}(\mathbf{rank}\,\mathbf{M} < m-d) \leq \frac{1}{2 \cdot 3^d}$$

when $\mathbf{M}$ is chosen uniformly at random in $\mathbb{F}_3^{(m-d)\times m}$. This implies $\mathbb{P}(X_i = 1) \leq \frac{1}{2 \cdot 3^d}$ and $\mathbb{E}\left\{\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right)\right\} = \mathbb{E}\left\{\frac{N}{\binom{n/2}{k-d}}\right\} = \sum_{i=1}^{\binom{n/2}{k-d}} \frac{\mathbb{P}(X_i=1)}{\binom{n/2}{k-d}} \leq \frac{1}{2 \cdot 3^d}$. This proves (24). (25) follows from similar arguments. □

We are ready now to prove Theorem 1.

Proof (of Theorem 1). By using Markov's inequality we have

$$\begin{aligned}
\mathbb{P}\left(\rho(\mathbf{e}, \mathbf{e}^{\text{unif}}) > 3^{-d/2}\right) &\leq 3^{d/2}\mathbb{E}\left\{\rho(\mathbf{e}, \mathbf{e}^{\text{unif}})\right\} \\
&\leq 3^{d/2}\mathbb{E}\Big\{\rho\Big(J^{\mathbf{H}_V}; J^{\text{unif}}\Big) + \textstyle\sum_{\mathbf{x}_V,\ell} \rho\Big(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\Big)\mathbb{P}(k_{\neq 0} = \ell | \mathbf{e}_V = \mathbf{x}_V)\mathbb{P}\Big(\mathbf{e}^{\text{unif}}_V = \mathbf{x}_V\Big)\Big\} \\
&\quad \text{(by Prop. 7)} \\
&\leq 3^{d/2}\left\{\frac{3^{-d}}{2} + \sum_{\mathbf{x}_V,\ell} \frac{3^{-d}}{2\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}}\right\} \quad \text{(by Lem. 2)} \\
&\leq 3^{-d/2}.
\end{aligned}$$

□

6 Achieving Uniform Domain Sampling

$\mathbf{H}_{\mathrm{pk}}$ denotes the public parity-check matrix of a normalized generalized $(U, U+V)$-code as described in §3.2. The random structure of $\mathbf{H}_{\mathrm{pk}}$ makes the syndromes associated to $\mathbf{H}_{\mathrm{pk}}$ indistinguishable in a very strong sense from random syndromes as the following proposition shows. This achieves the Domain Sampling property of Definition 1. The following definition will be useful.

Definition 5. (number of V blocks of type I). *In a normalized generalized $(U, U+V)$ code of length n associated to $(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})$, the number of V blocks of type I, which we denote by n_I, is defined as: $n_I \stackrel{\triangle}{=} |\{1 \le i \le n/2 : b_i d_i = 0\}|$.*

Proposition 9. *Let $\mathcal{D}_w^{\mathbf{H}}$ be the distribution of $\mathbf{e}\mathbf{H}^{\intercal}$ when $\mathbf{e}$ is drawn uniformly at random among S_w and let $\mathcal{U}$ be the uniform distribution over $\mathbb{F}_3^{n-k}$. We have*

$$\mathbb{E}_{\mathbf{H}_{\mathrm{pk}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\mathrm{pk}}},\mathcal{U})\right) \le \frac{1}{2}\sqrt{\varepsilon} \quad \textit{with,}$$

$$\varepsilon=\frac{3^{n-k}}{2^w\binom{n}{w}}+\sum_{j=0}^{\frac{n}{2}}\frac{3^{\frac{n}{2}-k_V}\binom{\frac{n}{2}}{j}\left(\sum\limits_{p=0:p\equiv w \pmod 2}^{j}\binom{j}{p}\binom{\frac{n}{2}-j}{\frac{w+p}{2}-j}2^{\frac{3p}{2}}\right)^2}{2^{w+j}\binom{n}{w}^2}+3^{\frac{n}{2}-k_U}\left(\sum_{j=0}^{n_I}\frac{\binom{n_I}{j}\binom{n-n_I}{w-j}^2}{\binom{n}{w}^2 2^j}\right).$$

This bound decays exponentially in n in a certain regime of parameters as shown by

Proposition 10. *Let $R_U \stackrel{\triangle}{=} \frac{2k_U}{n}$, $R_V \stackrel{\triangle}{=} \frac{2k_V}{n}$, $R \stackrel{\triangle}{=} \frac{k}{n}$, $\omega \stackrel{\triangle}{=} \frac{w}{n}$, $\nu \stackrel{\triangle}{=} \frac{n_I}{n}$, then under the same assumptions as in Proposition 9, we have as n tends to infinity*

$$\mathbb{E}_{\mathbf{H}_{\mathrm{pk}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\mathrm{pk}}},\mathcal{U})\right) \le 2^{(\alpha+o(1))n}$$

where $\alpha \stackrel{\triangle}{=} \frac{1}{2}\min\left((1-R)\log_2(3)-\omega-h_2(\omega),\alpha_1,\alpha_2\right)$ and

$$\alpha_1 \stackrel{\triangle}{=} \min_{(x,y)\in\mathcal{R}} \tfrac{1}{2}(1-R_V)\log_2 3-\omega-2h_2(\omega)+\tfrac{h_2(x)}{2}+x\left(h_2(y)+\tfrac{3}{2}y-\tfrac{1}{2}\right)+(1-x)h_2\left(\tfrac{\omega-x(1-y)}{1-x}\right)$$

$$\mathcal{R} \stackrel{\triangle}{=} \{(x,y)\in[0,1)\times[0,1]:0\le\omega-x(1-y)\le 1-x\}$$

$$\alpha_2 \stackrel{\triangle}{=} \min_{\max(0,\omega+\nu-1)\le x\le\min(\nu,\omega)} \tfrac{1}{2}(1-R_U)\log_2 3-2h_2(\omega)+\nu h_2\left(\tfrac{x}{\nu}\right)+2(1-\nu)h_2\left(\tfrac{\omega-x}{1-\nu}\right)-x.$$

Remark 4. For the set of parameters we present in the appendix, we have $\varepsilon \approx 2^{-354}$ and $\alpha \approx -0.02135$. Note that the upper-bound of Proposition 9 is by no means sharp, this comes from the $3^{\frac{n}{2}-k_U}\left(\sum_{j=0}^{n_I}\frac{\binom{n_I}{j}\binom{n-n_I}{w-j}^2}{\binom{n}{w}^2 2^j}\right)$ term which is a very crude upper-bound which is given here to avoid more complicated terms. It is straightforward to come up with a much sharper bound by improving this part of the upper-bound.

The proof of this proposition relies among other things on a variation of the left-over hash lemma [5] that is adapted to our case: here the hash function to which we apply the left-over hash lemma is defined as $\mathcal{H}(\mathbf{e}) = \mathbf{e}\mathbf{H}_{\mathrm{pk}}^{\intercal}$. $\mathcal{H}$ does not form a universal family of hash functions (essentially because the distribution of the $\mathbf{H}_{\mathrm{pk}}$'s is not the uniform distribution over $\mathbb{F}_3^{(n-k)\times n}$).

Lemma 3. *Consider a finite family $\mathcal{H} = (h_i)_{i \in I}$ of functions from a finite set E to a finite set F. Denote by ε the bias of the collision probability, i.e. the quantity such that*

$$\mathbb{P}_{h,e,e'}(h(e) = h(e')) = \frac{1}{|F|}(1+\varepsilon)$$

where h is drawn uniformly at random in $\mathcal{H}$, e and e' are drawn uniformly at random in E. Let $\mathcal{U}$ be the uniform distribution over F and $\mathcal{D}(h)$ be the distribution of the outputs $h(e)$ when e is chosen uniformly at random in E. We have

$$\mathbb{E}_h \left\{ \rho(\mathcal{D}(h), \mathcal{U}) \right\} \leq \frac{1}{2}\sqrt{\varepsilon}.$$

In order to use this lemma to bound the statistical distance we are interested in, we perform now the following computation

Lemma 4. *Assume that $\mathbf{x}$ and $\mathbf{y}$ are random vectors of S_w that are drawn uniformly at random in this set. We have*

$$\mathbb{P}_{\mathbf{H}_{\text{pk}},\mathbf{x},\mathbf{y}} \left(\mathbf{x}\mathbf{H}_{\text{pk}}^{\intercal} = \mathbf{y}\mathbf{H}_{\text{pk}}^{\intercal} \right) \leq \frac{1}{3^{n-k}}(1+\varepsilon) \text{ with } \varepsilon \text{ given in Proposition 9.}$$

Proof. By using Notation 1 and Proposition 3, the probability we are looking for is:

$$\mathbb{P}\left((\mathbf{x}_U - \mathbf{y}_U)\mathbf{H}_U^{\intercal} = \mathbf{0} \text{ and } (\mathbf{x}_V - \mathbf{y}_V)\mathbf{H}_V^{\intercal} = \mathbf{0} \right)$$

where the probability is taken over $\mathbf{H}_U, \mathbf{H}_V, \mathbf{x}, \mathbf{y}$. To compute the previous probability we will use a standard result, namely Lemma 12 proved in the appendix that gives that

$$\mathbb{P}\left(\mathbf{y}\mathbf{H}^{\intercal} = \mathbf{s} \right) = \frac{1}{3^r}, \tag{27}$$

when $\mathbf{y}$ is a non-zero vector of $\mathbb{F}_3^n$ and $\mathbf{s}$ an arbitrary element in $\mathbb{F}_3^r$ and when $\mathbf{H}$ is chosen uniformly at random in $\mathbb{F}_3^{r \times n}$. This motivates to distinguish between four disjoint events:

Event 1: $\mathcal{E}_1 \overset{\triangle}{=} \{\mathbf{x}_U = \mathbf{y}_U \quad \text{and} \quad \mathbf{x}_V \neq \mathbf{y}_V\}$; **Event 2:** $\mathcal{E}_2 \overset{\triangle}{=} \{\mathbf{x}_U \neq \mathbf{y}_U \quad \text{and} \quad \mathbf{x}_V = \mathbf{y}_V\}$

Event 3: $\mathcal{E}_3 \overset{\triangle}{=} \{\mathbf{x}_U \neq \mathbf{y}_U \quad \text{and} \quad \mathbf{x}_V \neq \mathbf{y}_V\}$; **Event 4:** $\mathcal{E}_4 \overset{\triangle}{=} \{\mathbf{x}_U = \mathbf{y}_U \quad \text{and} \quad \mathbf{x}_V = \mathbf{y}_V\}$

Under these events we get thanks to Lemma 12 and $k = k_U + k_V$:

$$\begin{aligned}
&\mathbb{P}_{\mathbf{H}_{\text{sk}},\mathbf{x},\mathbf{y}} \left(\mathbf{x}\mathbf{H}_{\text{sk}}^{\intercal} = \mathbf{y}\mathbf{H}_{\text{sk}}^{\intercal} \right) \\
&= \sum_{i=1}^{4} \mathbb{P}_{\mathbf{H}_{\text{sk}}} \left(\mathbf{x}\mathbf{H}_{\text{sk}}^{\intercal} = \mathbf{y}\mathbf{H}_{\text{sk}}^{\intercal} | \mathcal{E}_i \right) \mathbb{P}_{\mathbf{x},\mathbf{y}} \left(\mathcal{E}_i \right) \\
&= \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}} \left(\mathcal{E}_1 \right)}{3^{n/2-k_V}} + \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}} \left(\mathcal{E}_2 \right)}{3^{n/2-k_U}} + \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}} \left(\mathcal{E}_3 \right)}{3^{n-k}} + \mathbb{P}_{\mathbf{x},\mathbf{y}} \left(\mathcal{E}_4 \right) \\
&\leq \frac{1}{3^{n-k}} \left(1 + 3^{n/2-k_U} \mathbb{P}\left(\mathcal{E}_1 \right) + 3^{n/2-k_V} \mathbb{P}\left(\mathcal{E}_2 \right) + 3^{n-k} \mathbb{P}(\mathcal{E}_4) \right), \qquad (28)
\end{aligned}$$

where we used for the last inequality the trivial upper-bound $\mathbb{P}\left(\mathcal{E}_3\right) \leq 1$. Let us now upper-bound (or compute) the probabilities of the events $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_4$. For $\mathcal{E}_4$, recall that from the definition of normalized generalized $(U, U+V)$-codes, we clearly have

$$\mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_4\right) = \mathbb{P}(\mathbf{x} = \mathbf{y}) = \frac{1}{2^w \binom{n}{w}}. \tag{29}$$

Let us now estimate the probability of $\mathcal{E}_2$ for which we first derive the following upper-bound:

$$\mathbb{P}\left(\mathcal{E}_2\right) \leq \mathbb{P}\left(\mathbf{x}_V = \mathbf{y}_V\right)$$

To upper-bound this probability, we first observe that for any error $\mathbf{e} \in \mathbb{F}_3^{n/2}$ of weight j:

$$\begin{aligned} \mathbb{P}(\mathbf{x}_V = \mathbf{e}) &= \mathbb{P}\left(\mathbf{x}_V = \mathbf{e} \mid |\mathbf{x}_V| = j\right) \mathbb{P}(|\mathbf{x}_V| = j) \\ &= \frac{1}{2^j \binom{n/2}{j}} q_1(j) \end{aligned}$$

where $q_1^{\text{unif}}(j)$ denotes $\mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = j)$ and is computed in Proposition 5. From this we deduce that

$$\begin{aligned} \mathbb{P}(\mathbf{x}_V = \mathbf{y}_V) &= \sum_{j=0}^{n/2} \sum_{\mathbf{e} \in \mathbb{F}_3^{n/2} : |\mathbf{e}| = j} \mathbb{P}_{\mathbf{x}}(\mathbf{x}_V = \mathbf{e})^2 \\ &= \sum_{j=0}^{n/2} \frac{1}{2^j \binom{n/2}{j}} q_1^{\text{unif}}(j)^2 \end{aligned}$$

which gives:

$$\mathbb{P}\left(\mathcal{E}_2\right) \leq \sum_{j=0}^{n/2} \frac{q_1^{\text{unif}}(j)^2}{2^j \binom{n/2}{j}}. \tag{30}$$

The upper-bound on $\mathcal{E}_1$ is obtained in a similar way by using first that $\mathbb{P}(\mathcal{E}_1) \leq \mathbb{P}(\mathbf{x}_u \neq \mathbf{y}_U)$. As proved in Lemma 13 in the appendix, we have

$$\mathbb{P}(\mathbf{x}_u \neq \mathbf{y}_U) \leq \sum_{j=0}^{n_I} \binom{n_I}{j} 2^{-j} \left(\frac{\binom{n-n_I}{w-j}}{\binom{n}{w}} \right)^2.$$

□

7 Concluding Remarks and Further Work

We have presented Wave the first code-based "hash-and-sign" signature scheme which strictly follows the GPV strategy [28]. This strategy provides a very high level of security, but because of the multiple constraints it imposes, very few schemes managed to comply to it. For instance, only one such scheme based on hard lattice problems [24] was proposed to the recent NIST standardization effort. Our scheme is secure under two assumptions from coding theory. Both of those assumptions relate closely to hard decoding problems. Using rejection sampling, we have shown how to efficiently avoid key leakage from any number of signatures. The main purpose of our work was to propose this new scheme and assess its security. Still, it has a few issues and extensions that are of interest.

The Far Away Decoding Problem. The message security of Wave relates to the hardness of finding a codeword *far* from a given word. A recent work [12] adapts the best ISD techniques for low weight [40, 8] and goes even further with a higher order generalized birthday algorithm [50]. Interestingly enough, in the non-binary case, this work gives a worst case exponent for the far away codeword that is significantly larger than the close codeword worst case exponent. This seems to point to the fact that the far away codeword problem may even be more difficult to solve than the close codeword problem. This raises the issue of obtaining code-based primitives with better parameters that build upon the far away codeword rather than on the usual close codeword problem.

Distinguishability. Deciding whether a matrix is a parity check matrix of a generalized $(U, U + V)$-code is also a new problem. As shown in [17] it is hard in the worst case since the problem is NP-complete. In the binary case, $(U, U+V)$ codes have a large hull dimension for some set of parameters which are precisely those used in [17]. In the ternary case the normalized generalized $(U, U + V)$-codes do not suffer from this flaw. The freedom of the choice on vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ is very likely to make the distinguishing problem much harder for generalized $(U, U + V)$-codes than for plain $(U, U + V)$-codes. Coming up with non-metric based distinguishers in the generalized case seems a tantalizing problem here.

On the Tightness of the Security Reduction. It could be argued that one of the reasons of why we have a tight security-reduction comes from the fact that we reduce to the multiple instances version of the decoding problem, namely DOOM, instead of the decoding problem itself. This is true to some extent, however this problem is as natural as the decoding problem itself. It has already been studied in some depth [46] and the decoding techniques for linear codes have a natural extension to DOOM as noticed in [46]. We also note that with our approach, where a message has many possible signatures, we avoid the tightness impossibility results given in [3] for instance.

Rejection Sampling. Rejection sampling in our algorithm is relatively unobtrusive: a rejection every few signatures with a crude tuning of the decoder. We believe that it can be further improved. Our decoding has two steps. Each step is parametrized by a weight distribution which conditions the output weight distribution. We believe that we can tune those distributions to reduce the probability of rejection to an arbitrarily small value and thus to avoid the rejection phase.

Improving Parameters. In order to predict accurately enough the output distribution of the signature algorithm, we had to restrict the decoders by excluding d positions from the information sets. Our result almost certainly applies when $d = 0$. By either proving it or stating it as a conjecture we may reduce the block size by more than 10%.

References

1. Alkim, E., Bindel, N., Buchmann, J.A., Dagdelen, Ö., Eaton, E., Gutoski, G., Krämer, J., Pawlega, F.: Revisiting TESLA in the quantum random oracle model. In: Post-Quantum Cryptography 2017. LNCS 10346, pp. 143–162. Springer (2017).
2. Aragon, N., Blazy, O., Gaborit, P., Hauteville, A., Zémor, G.: Durandal: a rank metric based signature scheme. IACR Cryptology ePrint Archive (2018), `https://eprint.iacr.org/2018/1192`
3. Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: Fischlin, M., Coron, J. (eds.) Advances in Cryptology - EUROCRYPT 2016. LNCS, vol. 9666, pp. 273–304. Springer (2016)
4. Baldi, M., Bianchi, M., Chiaraluce, F., Rosenthal, J., Schipani, D.: Using LDGM codes and sparse syndromes to achieve digital signatures. In: Post-Quantum Cryptography 2013. LNCS, vol. 7932, pp. 1–15. Springer (2013)
5. Barak, B., Dodis, Y., Krawczyk, H., Pereira, O., Pietrzak, K., Standaert, F., Yu, Y.: Leftover hash lemma, revisited. In: Advances in Cryptology - CRYPTO 2011. LNCS, vol. 6841, pp. 1–20. Springer (2011)
6. Barg, A.: Complexity issues in coding theory. Electronic Colloquium on Computational Complexity (Oct 1997), `https://eccc.weizmann.ac.il/eccc-reports/1997/TR97-046/Paper.pdf`
7. Barreto, P.S., Misoczki, R., Simplicio, M.A.J.: One-time signature scheme from syndrome decoding over generic error-correcting codes. Journal of Systems and Software 84(2), 198–204 (2011)
8. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1+1 = 0$ improves information set decoding. In: Advances in Cryptology - EUROCRYPT 2012. LNCS, Springer (2012)
9. Bellare, M., Rogaway, P.: The exact security of digital signatures-how to sign with rsa and rabin. In: Advances in Cryptology - EUROCRYPT '96. LNCS, vol. 1070, pp. 399–416. Springer (1996)
10. Bernstein, D.J., Chou, T., Schwabe, P.: Mcbits: Fast constant-time code-based cryptography. In: Bertoni, G., Coron, J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2013. LNCS, vol. 8086, pp. 250–272. Springer (2013)
11. Both, L., May, A.: Decoding linear codes with high error rate and its impact for LPN security. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography 2018. LNCS, vol. 10786, pp. 25–46. Springer, Fort Lauderdale, FL, USA (Apr 2018)
12. Bricout, R., Chailloux, A., Debris-Alazard, T., Lequesne, M.: Ternary syndrome decoding with large weights. preprint (Feb 2019), arXiv:1903.07464
13. Cayrel, P.L., Otmani, A., Vergnaud, D.: On Kabatianskii-Krouk-Smeets signatures. In: Arithmetic of Finite Fields - WAIFI 2007. LNCS, vol. 4547, pp. 237–251. Madrid, Spain (Jun 21–22 2007)
14. Coron, J.: Optimal security proofs for PSS and other signature schemes. In: Advances in Cryptology - EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer (2002)
15. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Advances in Cryptology - ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer (2001)
16. Debris-Alazard, T., Sendrier, N., Tillich, J.P.: A new signature scheme based on $(U|U + V)$ codes. preprint (Jun 2017), arXiv:1706.08065v1
17. Debris-Alazard, T., Sendrier, N., Tillich, J.P.: The problem with the surf scheme. preprint (Nov 2017), arXiv:1706.08065

18. Debris-Alazard, T., Tillich, J.P.: Statistical decoding. preprint (Jan 2017), arXiv:1701.07416
19. Debris-Alazard, T., Tillich, J.P.: Two attacks on rank metric code-based schemes: Ranksign and an identity-based-encryption scheme. In: Advances in Cryptology - ASIACRYPT 2018. LNCS, Springer (2018)
20. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52. Moscow (1991)
21. Faugère, J.C., Gauthier, V., Otmani, A., Perret, L., Tillich, J.P.: A distinguisher for high rate McEliece cryptosystems. In: Proc. IEEE Inf. Theory Workshop- ITW 2011. pp. 282–286. Paraty, Brasil (Oct 2011)
22. Finiasz, M.: Parallel-CFS - strengthening the CFS McEliece-based signature scheme. In: Selected Areas in Cryptography 17th International Workshop, 2010, Waterloo, Ontario, Canada, August 12-13, 2010, revised selected papers. LNCS, vol. 6544, pp. 159–170. Springer (2010)
23. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui, M. (ed.) Advances in Cryptology - ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer (2009)
24. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over ntru
25. Fukushima, K., Roy, P.S., Xu, R., Kiyomoto, S., Morozov, K., Takagi, T.: RaCoSS (random code-based signature scheme). first round submission to the NIST post-quantum cryptography call (Nov 2017), `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/RaCoSS.zip`
26. Gaborit, P., Ruatta, O., Schrek, J., Zémor, G.: New results for rank-based cryptography. In: Progress in Cryptology - AFRICACRYPT 2014. LNCS, vol. 8469, pp. 1–12 (2014)
27. Gaborit, P., Schrek, J.: Efficient code-based one-time signature from automorphism groups with syndrome compatibility. In: Proc. IEEE Int. Symposium Inf. Theory - ISIT 2012. pp. 1982–1986. Cambridge, MA, USA (Jul 2012)
28. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206. ACM (2008)
29. Gligoroski, D., Samardjiska, S., Jacobsen, H., Bezzateev, S.: McEliece in the world of Escher. IACR Cryptology ePrint Archive, Report2014/360 (2014)
30. Goldwasser, S., Micciancio, D.: Complexity of lattice problems: A cryptographic perspective, volume 671 of the kluwer international series in engineering and computer science. Kluwer Academic Publishers (Mar 2002)
31. Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010. LNCS, vol. 6110. Sringer (2010)
32. Huelsing, A., Bernstein, D.J., Panny, L., Lange, T.: Official NIST comments made for RaCoSS (2018), `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/RaCoSS-official-comment.pdf`, official NIST comments made for RaCoSS
33. Johansson, T., Jönsson, F.: On the complexity of some cryptographic problems based on the general decoding problem. IEEE Trans. Inform. Theory 48(10), 2669–2678 (Oct 2002)

34. Kabatianskii, G., Krouk, E., Semenov, S.: Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept. John Wiley & Sons (2005)
35. Kabatianskii, G., Krouk, E., Smeets, B.J.M.: A digital signature scheme based on random error-correcting codes. In: IMA Int. Conf. LNCS, vol. 1355, pp. 161–167. Springer (1997)
36. Landais, G., Sendrier, N.: Implementing CFS. In: Progress in Cryptology - INDOCRYPT 2012. LNCS, vol. 7668, pp. 474–488. Springer (2012)
37. Lee, W., Kim, Y.S., Lee, Y.W., No, J.S.: Post quantum signature scheme based on modified Reed-Muller code pqsigRM. first round submission to the NIST post-quantum cryptography call (Nov 2017), `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/pqsigRM.zip`
38. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: ASIACRYPT (2009)
39. Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 598–616. Springer (2009)
40. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $O(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer (2011)
41. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015. LNCS, vol. 9056, pp. 203–228. Springer (2015)
42. Moody, D., Perlner, R.A.: Vulnerabilities of "McEliece in the World of Escher". In: Post-Quantum Cryptography 2016. LNCS, Springer (2016)
43. Otmani, A., Tillich, J.P.: An efficient attack on all concrete KKS proposals. In: Post-Quantum Cryptography 2011. LNCS, vol. 7071, pp. 98–116 (2011)
44. Phesso, A., Tillich, J.: An efficient attack on a code-based signature scheme. In: Post-Quantum Cryptography 2016. LNCS, vol. 9606, pp. 86–103. Springer, Fukuoka, Japan (Feb 2016)
45. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory 8(5), 5–9 (1962)
46. Sendrier, N.: Decoding one out of many. In: Post-Quantum Cryptography 2011. LNCS, vol. 7071, pp. 51–67 (2011)
47. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004)
48. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications. LNCS, vol. 388, pp. 106–113. Springer (1988)
49. Stern, J.: A new identification scheme based on syndrome decoding. In: Stinson, D. (ed.) Advances in Cryptology - CRYPTO'93. LNCS, vol. 773, pp. 13–21. Springer (1993)
50. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer (2002)