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

Thomas Debris-Alazard[1,2], Nicolas Sendrier[2], and Jean-Pierre Tillich[2]

[1] Sorbonne Universités, UPMC Univ Paris 06
[2] Inria, Paris
{thomas.debris,nicolas.sendrier,jean-pierre.tillich}@inria.fr

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 [GPV08]. 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 below one rejection every 100 signatures.

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 [CFS01]. 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 [FGO$^+$11], (ii) security scales only weakly superpolynomially in the keysize for polynomial time signature generation. A crude extrapolation of parallel CFS [Fin10] and its implementations [LS12, BCS13] 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 [BBC$^+$13, GSJB14, LKLN17]. All of them were broken, see for instance [PT16, MP16]. Other signature schemes based on codes were also given in the literature such as for instance the KKS scheme [KKS97, KKS05], its variants [BMS11, GS12] or the RaCoSS proposal [FRX$^+$17] 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 [COV07, OT11, HBPL18]. Finally, another possibility is to use the Fiat-Shamir heuristic. For instance by turning the Stern zero-knowledge authentication scheme [Ste93] 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 [GRSZ14], that enjoys remarkably small key sizes, but it got broken too

⋆ This work was supported by the ANR CBCRYPT project, grant ANR-17-CE39-0007 of the French Agence Nationale de la Recherche.

in [DT18]. On the other hand, following the Schnorr-Lyubashevsky [Lyu09a] approach, a new scheme was recently proposed, namely Durandal [ABG+18]. This scheme enjoys small key sizes and managed to meet the challenge of adapting the Lyubashevsky [Lyu09b] 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 [ABG+18, §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* [GPV08, §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 [GPV08] 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 [GPV08] are namely Gentry, Peikert and Vaikuntanathan). This strategy has also been adopted in Falcon [FHK+17], 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 Preimage 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) [Sen11]. In [GPV08] 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 [GPV08], 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 [GPV08] 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 $k = \Theta(n)$. 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:

$$\mathbf{x} \odot \mathbf{y} \stackrel{\triangle}{=} (x_1 y_1, \cdots, x_{n/2} y_{n/2}).$$

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 [Pra62] 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 $\rho \stackrel{\triangle}{=} \frac{r}{n}$. $\mathtt{D}_{UV}$ 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, less than one rejection every 100 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 [GPV08] 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 [JJ02]. 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 [Sen11]. 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 [DST17b, §7.1, Thm 3] and the proof carries over to an arbitrary finite field $\mathbb{F}_q$. However as observed in [DST17b, 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 [DST17a, §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 [DST17a] is able to work properly outside the critical interval $[\![\frac{r}{2}, n - \frac{r}{2}]\!]$. This explains why the ancestor Surf [DST17a] 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?"* §3.3). 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 [ABB+17]. This is due in part to the tightness of our security reduction.

1 Notation

We provide here some notation that will be used throughout the paper.

General Notation. The notation $x \stackrel{\triangle}{=} y$ 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 $\{a, a+1, \ldots, b\}$.

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. $\mathbf{x}_\mathcal{I} = (x_i)_{i \in \mathcal{I}}$. 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

$$\mathrm{Supp}(\mathbf{x}) \stackrel{\triangle}{=} \{i \in [\![1, n]\!] \text{ such that } x_i \neq 0\}$$

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. It will be clear from the context whether $|\mathbf{x}|$ means the Hamming weight or the size of a finite set. Note that $|\mathbf{x}| = |\mathrm{Supp}(\mathbf{x})|$. 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 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:

$$\rho(\mathcal{D}_0, \mathcal{D}_1) \stackrel{\triangle}{=} \frac{1}{2} \sum_{x \in \mathcal{E}} |\mathcal{D}_0(x) - \mathcal{D}_1(x)|.$$

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 (which is usually the case) we have $r = n - k$. A *generator matrix* of $\mathcal{C}$ is a $k \times n$ full rank matrix $\mathbf{G}$ over $\mathbb{F}_q$ such that $\langle \mathbf{G} \rangle = \mathcal{C}$. 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 $\mathcal{I} \subset [\![1, n]\!]$ which indexes k independent columns on any generator matrix. 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 $\mathbf{x}_{\mathcal{I}} = \mathbf{e}_{\mathcal{I}}$.

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

2.1 One-way Preimage Sampleable Code-based Functions

In this work we will use the FDH paradigm [BR96, Cor02] 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 $\boldsymbol{\sigma} \in f_{w,\mathbf{H}}^{-1}(\mathbf{h})$ 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 [GPV08] to the one-way function (we will speak of the GPV strategy). This is mostly captured by the notion of Preimage Sampleable Functions, see [GPV08, 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* $(\texttt{Trapdoor}, \texttt{InvertAlg})$ *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:

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

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

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

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 $\mathbf{eH}^\intercal = \mathbf{s}$ 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 [GPV08] would translate in our setting in the following way. For any $\mathbf{s} \in \mathbb{F}_q^{n-k}$ we should have

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

As pointed out in [S19], we have

$$\begin{aligned}
\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}\right) &= \sum_{\mathbf{s}} \sum_{\mathbf{e} \in f_{w,\mathbf{H}}^{-1}(\mathbf{s})} \left| \frac{1}{|S_w|} - \frac{1}{q^{n-k}} \mathbb{P}(\texttt{InvertAlg}(\mathbf{s}, T) = \mathbf{e}) \right| \\
&= \sum_{\mathbf{s}} \sum_{\mathbf{e} \in f_{w,\mathbf{H}}^{-1}(\mathbf{s})} \left| \frac{1}{|S_w|} - \frac{1}{q^{n-k}|f_{w,\mathbf{H}}^{-1}(\mathbf{s})|} + \frac{1}{q^{n-k}|f_{w,\mathbf{H}}^{-1}(\mathbf{s})|} - \frac{1}{q^{n-k}} \mathbb{P}(\texttt{InvertAlg}(\mathbf{s}, T) = \mathbf{e}) \right| \\
&\geq \sum_{\mathbf{s}} \frac{1}{q^{n-k}} \sum_{\mathbf{e} \in f_{w,\mathbf{H}}^{-1}(\mathbf{s})} \left| \frac{1}{|f_{w,\mathbf{H}}^{-1}(\mathbf{s})|} - \mathbb{P}(\texttt{InvertAlg}(\mathbf{s}, T) = \mathbf{e}) \right| - \sum_{\mathbf{s}} \left| \frac{|f_{w,\mathbf{H}}^{-1}(\mathbf{s})|}{|S_w|} - \frac{1}{q^{n-k}} \right| \\
&= \sum_{\mathbf{s} \in \mathbb{F}_q^{n-k}} \frac{1}{q^{n-k}} \rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right) - \rho(\mathbf{eH}^\intercal, \mathbf{s}).
\end{aligned}$$

Therefore with the domain sampling property and our definition of the preimage sampling property the average of the $\rho\left(\texttt{InvertAlg}(\mathbf{s}, T), \mathbf{e}_s\right)$'s is negligible too, whereas [GPV08] 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

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

As noted by the anonymous reviewer, this relaxed property is enough to apply the GPV proof technique.

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 [GPV08].

Given a one-way preimage sampleable on average 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

$\texttt{Sgn}^{\mathrm{sk}}(\mathbf{m})$:	$\texttt{Vrfy}^{\mathrm{pk}}(\mathbf{m}, (\mathbf{e}', \mathbf{r}))$:
$\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$	$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$
$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m}, \mathbf{r})$	`if` $\mathbf{e}'\mathbf{H}^\intercal = \mathbf{s}$ `and` $\lvert\mathbf{e}'\rvert = w$ `return` 1
$\mathbf{e} \leftarrow \texttt{InvertAlg}(\mathbf{s}, T)$	`else return` 0
$\texttt{return}(\mathbf{e}, \mathbf{r})$	

A tight security reduction in the random oracle model is given in [GPV08] 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 §6 a reduction to the multiple target decoding problem.

2.2 The Wave Family of One-Way Trapdoor Preimage Sampleable 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*

$$\begin{aligned}\varphi_{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}} : \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} &\to \mathbb{F}_q^{n/2} \times \mathbb{F}_q^{n/2} \\ (\mathbf{x},\mathbf{y}) &\mapsto (\mathbf{a}\odot\mathbf{x} + \mathbf{b}\odot\mathbf{y}, \mathbf{c}\odot\mathbf{x} + \mathbf{d}\odot\mathbf{y}).\end{aligned}$$

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

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

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

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

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*

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

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

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 $a_iu_i + b_iv_i$'s and $c_iu_i + d_iv_i$'s.
- The condition $a_ic_i \neq 0$ 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 $a_id_i - b_ic_i = 1$ 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 $a_id_i - b_ic_i \neq 0$ can also be obtained in the restricted case $a_id_i - b_ic_i = 1$ 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 §4.4. The secret key is a tuple $\mathsf{sk} = (\varphi, \mathbf{H}_U, \mathbf{H}_V, \mathbf{S}, \mathbf{P})$ 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 non-singular 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 $\mathbf{H}_{\rm sk} = \mathcal{H}(\varphi, \mathbf{H}_U, \mathbf{H}_V)$ as in Proposition 1. The public key is $\mathbf{H}_{\rm pk} = \mathbf{S}\mathbf{H}_{\rm sk}\mathbf{P}$. Next, we need to produce an algorithm $D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}$ which inverts $f_{w,\mathbf{H}_{\rm sk}}$. The parameter w is such that this can be achieved using the underlying $(U, U+V)$ structure while the generic problem remains hard. In §4 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:

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

As in [GPV08], putting this together with a domain sampling condition –which we prove in §5 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.

3 Inverting the Syndrome Function

This section is devoted to the inversion of $f_{w,\mathbf{H}}$. It amounts to solve the following problem.

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 $[\![w^-_{\rm easy}, w^+_{\rm easy}]\!] \subset [\![w^-_{\rm UV}, w^+_{\rm UV}]\!] \subset [\![w^-, w^+]\!]$ 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 $w \in [\![w^-_{\rm easy}, w^+_{\rm easy}]\!]$ for all $\mathbf{H}$ (Prange algorithm)
- $\mathbf{e} \in f^{-1}_{w,\mathbf{H}}(\mathbf{s})$ is easy to find if $w \in [\![w^-_{\rm UV}, w^+_{\rm UV}]\!]$ 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 $f_{w,\mathbf{H}}$.

3.1 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 $\omega^- \triangleq \lim_{n\to+\infty} w^-/n$ and $\omega^+ \triangleq \lim_{n\to+\infty} w^-/n$. 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^+).

3.2 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 $w/n \in [\omega^-_{\rm easy}, \omega^+_{\rm easy}]$ where

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

where $R \triangleq \frac{k}{n}$. This is achieved by a slightly generalized version of the Prange decoder [Pra62]. We want to find for a given $\mathbf{s}$ an error $\mathbf{e}$ of weight w such that $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$. The matrix $\mathbf{H}$ is a full-rank matrix and it therefore contains an invertible submatrix $\mathbf{A}$ of size $(n-k)\times(n-k)$. We choose a set of positions $\mathcal{I}$ of size $n-k$ for which $\mathbf{H}$ restricted to these positions is a full rank matrix. For simplicity assume that this matrix is in the first $n-k$ positions: $\mathbf{H} = \big(\mathbf{A}|\mathbf{B}\big)$. We look for an $\mathbf{e}$ of the form $\mathbf{e} = (\mathbf{e}'', \mathbf{e}')$ where $\mathbf{e}' \in \mathbb{F}_q^k$ and $\mathbf{e}'' \in \mathbb{F}_q^{n-k}$. We should therefore have $\mathbf{e}'' = (\mathbf{s} - \mathbf{e}'\mathbf{B}^\intercal)(\mathbf{A}^{-1})^\intercal$. In this way we can arbitrarily choose the error $\mathbf{e}'$ of length k but in any case we expect for the remaining part a vector $\mathbf{e}''$ with about $\frac{q-1}{q}(n-k)$ positions that are non zero. Therefore, the weights that are easily attainable by this strategy are between $\frac{q-1}{q}(n-k) = n\omega^-_{\text{easy}}$ and $k + \frac{q-1}{q}(n-k) = n\omega^+_{\text{easy}}$ by choosing appropriately the weight of $\mathbf{e}'$ between 0 and k. 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: $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}$

1: $t \hookleftarrow \mathcal{D}$
2: $\mathcal{I} \leftarrow$ InfoSet$(\mathbf{H})$ ▷ InfoSet$(\mathbf{H})$ *returns an information set of* $\langle\mathbf{H}\rangle^\perp$
3: $\mathbf{x} \hookleftarrow \{\mathbf{x} \in \mathbb{F}_q^n \mid |\mathbf{x}_\mathcal{I}| = t\}$
4: $\mathbf{e} \leftarrow$ PrangeStep$(\mathbf{H}, \mathbf{s}, \mathcal{I}, \mathbf{x})$
5: **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
$(\mathbf{A} \mid \mathbf{B}) \leftarrow \mathbf{H}\mathbf{P}$ ▷ $\mathbf{A} \in \mathbb{F}_q^{(n-k)\times(n-k)}$
$(\mathbf{0} \mid \mathbf{e}') \leftarrow \mathbf{x}$ ▷ $\mathbf{e}' \in \mathbb{F}_q^k$
$\mathbf{e} \leftarrow \Big(\big(\mathbf{s} - \mathbf{e}'\mathbf{B}^\intercal\big)\big(\mathbf{A}^{-1}\big)^\intercal, \mathbf{e}'\Big)\,\mathbf{P}^\intercal$
return $\mathbf{e}$

Proposition 2. *Let* $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ *of rank* $n-k$ *and* $\mathbf{s}$ *which is uniformly distributed in* $\mathbb{F}_q^{n-k}$. *Then, for for the output* $\mathbf{e}$ *of* PrangeOne$(\mathbf{H}, \mathbf{s})$ *we have*

$$|\mathbf{e}| = S + T$$

where $S \in [\![0, n-k]\!]$ *and* $T \in [\![0, k]\!]$ *are independent random variables,* S *is the Hamming weight of a vector that is uniformly distributed over* $\mathbb{F}_q^{n-k}$ *and* $\mathbb{P}(T = t) = \mathcal{D}(t)$. *The distribution of* $|\mathbf{e}|$ *is given by*

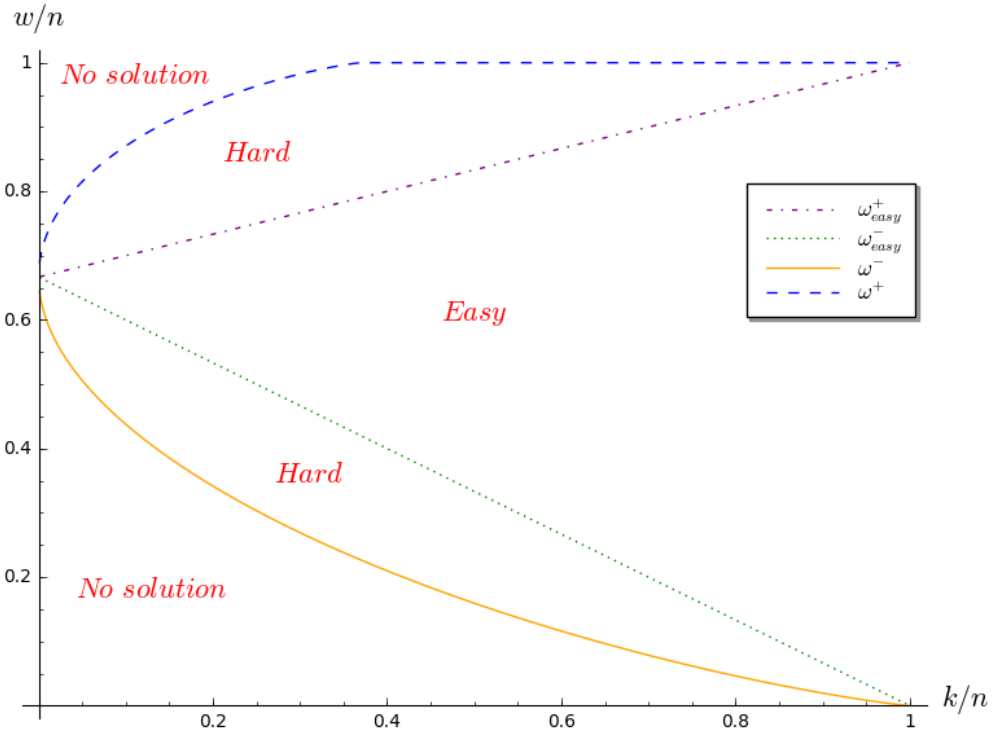
$$\mathbb{P}\left(|\mathbf{e}| = w\right) = \sum_{t=0}^{w} \frac{\binom{n-k}{w-t}(q-1)^{w-t}}{q^{n-k}}\mathcal{D}(t), \quad \mathbb{E}(|\mathbf{e}|) = \overline{\mathcal{D}} + \tfrac{q-1}{q}(n-k) = \overline{\mathcal{D}} + n\omega^-_{easy}$$

where $\overline{\mathcal{D}} = \sum_{t=0}^{k} t\mathcal{D}(t)$.

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.

To summarize this discussion we have shown that when we want to ensure that $f_{\mathbf{H}}$ is surjective, w has to verify $w^- \leq w \leq w^+$. However, in a cryptographic setting w/n cannot lie in $[\omega^-_{\text{easy}}, \omega^+_{\text{easy}}] \subseteq [\omega^-, \omega^+]$ otherwise anybody that uses the generalized Prange algorithm would be able to invert $f_{w,\mathbf{H}}$. All of this is summarized in Figure 1 where we draw the above different areas asymptotically in n of w/n when k/n is fixed and $q = 3$.

Fig. 1. Areas of relative signature distances when $q = 3$.



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 $\omega \triangleq \frac{w}{n}$ in $(0, \omega^-_{\text{easy}})$ and despite many efforts the best algorithms [Ste88, Dum91, Bar97, MMT11, BJMM12, MO15, DT17, BM18] 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. In the following subsection we present a trapdoor on the matrices $\mathbf{H}$ that enables to invert in polynomial time $f_{w,\mathbf{H}}$ on a larger range by tweaking the Prange decoder.

3.3 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 (see Proposition 1 in §2.2). As we will see in what follows, this family comes with a simple procedure which enables to invert $f_{w,\mathbf{H}}$ with errors of weight which belongs to

$[\![w^-_{\mathrm{UV}}, w^+_{\mathrm{UV}}]\!] \subset [\![w^-, w^+]\!]$ but with $[\![w^-_{\mathrm{easy}}, w^+_{\mathrm{easy}}]\!] \subsetneq [\![w^-_{\mathrm{UV}}, w^+_{\mathrm{UV}}]\!]$. We summarize this situation in Figure 2.

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.

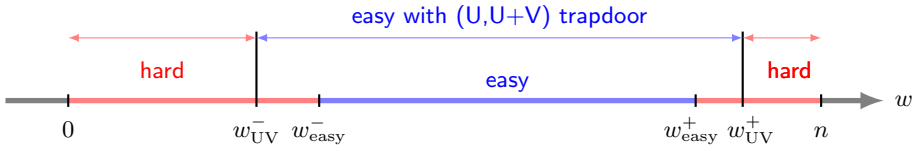


Fig. 2. Hardness of $(U, U+V)$ Decoding

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 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 we will consider recovers $\mathbf{e}_V$ 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 an $\mathbf{e}_V$ 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*

$$\mathbf{e}_U \mathbf{H}_U^\intercal = \mathbf{s}^U \tag{7}$$
$$\mathbf{e}_V \mathbf{H}_V^\intercal = \mathbf{s}^V \tag{8}$$

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,

$$\begin{cases} \mathbf{e}_U\mathbf{A}\mathbf{D}^\intercal\mathbf{H}_U^\intercal + \mathbf{e}_V\mathbf{B}\mathbf{D}^\intercal\mathbf{H}_U^\intercal - \mathbf{e}_U\mathbf{C}\mathbf{B}^\intercal\mathbf{H}_U^\intercal - \mathbf{e}_V\mathbf{D}\mathbf{B}^\intercal\mathbf{H}_U^\intercal & = \mathbf{s}^U \\ -\mathbf{e}_U\mathbf{A}\mathbf{C}^\intercal\mathbf{H}_V^\intercal - \mathbf{e}_V\mathbf{B}\mathbf{C}^\intercal\mathbf{H}_V^\intercal + \mathbf{e}_U\mathbf{C}\mathbf{A}^\intercal\mathbf{H}_V^\intercal + \mathbf{e}_V\mathbf{D}\mathbf{A}^\intercal\mathbf{H}_V^\intercal & = \mathbf{s}^V \end{cases}$$

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 $\mathbf{AD}-\mathbf{BC} = \mathbf{I}_{n/2}$, the identity matrix of size $n/2$. □

Performing the two decoding (7) and (8) independently with the Prange algorithm gains nothing. However if we first solve (8) with the Prange algorithm, and then seek a solution of (7) 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 $[\omega^-_{\mathrm{UV}}, \omega^+_{\mathrm{UV}}]$ of relative weights w/n for which the $(U, U+V)$-decoder works in polynomial time is larger than $[\omega^-_{\mathrm{easy}}, \omega^+_{\mathrm{easy}}]$. 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

$$\begin{cases} a_i\mathbf{e}_U(i) + b_i\mathbf{e}_V(i) \neq 0 \\ c_i\mathbf{e}_U(i) + d_i\mathbf{e}_V(i) \neq 0. \end{cases}$$

This is always possible since $q \geq 3$ and $a_ic_i \neq 0$ for all i which gives an expected weight of $\mathbf{e}$:

$$\mathbb{E}(|\mathbf{e}|) = 2\left(k_U + \frac{q-1}{q}(n/2 - k_U)\right) = \frac{q-1}{q}n + \frac{2k_U}{q} \tag{9}$$

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 weight of the final error we would like to choose $\mathbf{e}_U(i)$ in k_U positions such that

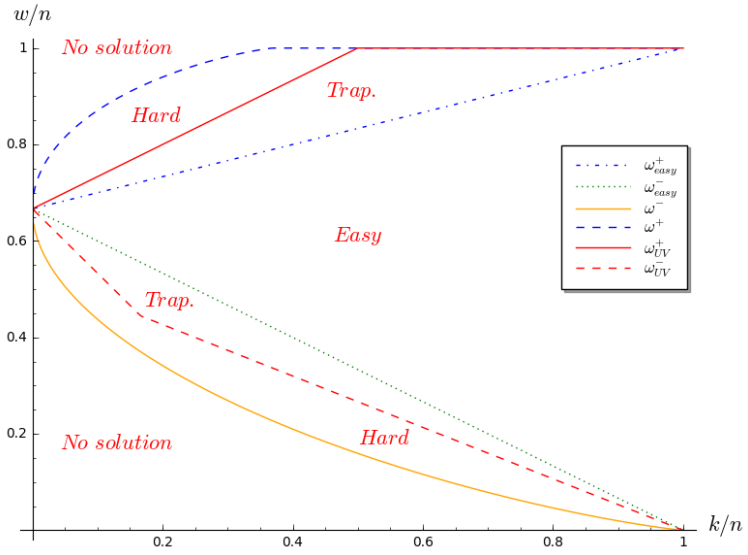
$$\begin{cases} a_i\mathbf{e}_U(i) + b_i\mathbf{e}_V(i) = 0 \\ c_i\mathbf{e}_U(i) + d_i\mathbf{e}_V(i) = 0 \end{cases}$$

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:

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

This discussion is summarized in Figure 3 where we draw ω^-_{UV} and ω^+_{UV} which are the highest and the smallest relative distances that our decoder can reach asymptotically in n when k/n is fixed and $q = 3$.

Fig. 3. Areas of relative signature distances with our trapdoor when $q = 3$



4 Preimage Sampling with Trapdoor: Achieving a Uniformly Distributed Output

We restrict here our study to the case,

$$\boxed{q = 3.}$$

All the results we are going to give can be generalized to larger values of 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 4 gives a rough picture of what will happen.

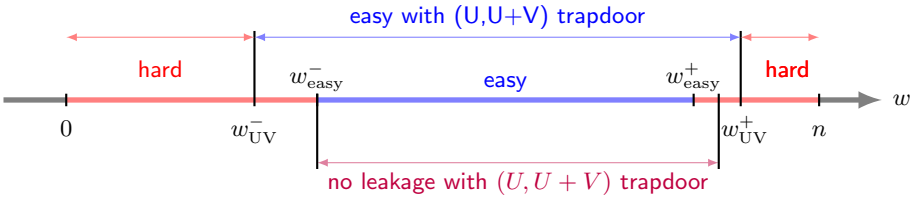


Fig. 4. Hardness of $(U, U + V)$ Decoding with no leakage of signature

4.1 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 $\mathbf{eH}^\intercal = \mathbf{s}$ 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) and (8) through an algorithm whose skeleton is given in Algorithm 2. $\text{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$ returns a vector satisfying $\mathbf{e}_V \mathbf{H}_V^\intercal = \mathbf{s}^V$, whereas $\text{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$ is assumed to return 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 $\text{DecodeUV}(\mathbf{H}_V, \mathbf{H}_U, \varphi, \mathbf{s})$

1: **repeat**
2: $\quad \mathbf{e}_V \leftarrow \text{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$
3: **until** Condition 1 is met
4: **repeat**
5: $\quad \mathbf{e}_U \leftarrow \text{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$ ▷ We assume that $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$ here.
6: $\quad \mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$
7: **until** Condition 2 is met
8: **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 $\text{DecodeU}(\cdot)$ 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 $\text{DecodeV}(\cdot)$ and $\text{DecodeU}(\cdot)$ derived from Prange decoders that we will consider that will be very helpful here. They will namely be very close to meet the following conditions.

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

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

It is readily observed that for all $\mathbf{x} \in S_w$,

$$\mathbb{P}(\mathbf{e}_V^{\text{unif}} = \mathbf{x}_V) \quad \text{and} \quad \mathbb{P}(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V)$$

are also only functions of $|\mathbf{x}_V|$ and $(|\mathbf{x}_V|, m_1(\mathbf{x}))$ 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* $\text{DecodeU}(\cdot)$ *is* m_1*-uniform whereas* $\text{DecodeV}(\cdot)$ *is weightwise uniform. If for any possible* y *and* z*,*

$$|\mathbf{e}_V| \sim |\mathbf{e}_V^{\text{unif}}| \quad \textit{and} \quad \mathbb{P}(m_1(\mathbf{e}) = z \mid |\mathbf{e}_V| = y) = \mathbb{P}(m_1(\mathbf{e}^{\text{unif}}) = z \mid |\mathbf{e}_V^{\text{unif}}| = y) \tag{10}$$

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* $\text{DecodeU}(\cdot)$ *and* $\text{DecodeV}(\cdot)$.

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

$$\begin{aligned}\mathbb{P}(\mathbf{e}=\mathbf{x}) &= \mathbb{P}(\mathbf{e}_U=\mathbf{x}_U \mid \mathbf{e}_V=\mathbf{x}_V)\mathbb{P}(\mathbf{e}_V=\mathbf{x}_V)\\ &= \mathbb{P}(\textsc{DecodeU}(\mathbf{H}_U,\varphi,\mathbf{s}^U,\mathbf{e}_V)=\mathbf{x}_U \mid \mathbf{e}_V=\mathbf{x}_V)\mathbb{P}(\textsc{DecodeV}(\mathbf{H}_V,\mathbf{s}^V)=\mathbf{x}_V)\\ &= \frac{\mathbb{P}(m_1(\mathbf{e})=z \mid |\mathbf{e}_V|=y)}{n(y,z)}\frac{\mathbb{P}(|\mathbf{e}_V|=y)}{n(y)} \triangleq P \end{aligned} \tag{11}$$

where $n(y)$ is the number of vectors of $\mathbb{F}_3^n$ 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 (11) is here a consequence of the weightwise uniformity of $\textsc{DecodeV}(\cdot)$ on one hand and the m_1-uniformity of $\textsc{DecodeU}(\cdot)$ on the other hand. We conclude by noticing that

$$P = \frac{\mathbb{P}(m_1(\mathbf{e}^{\text{unif}})=z \mid |\mathbf{e}_V^{\text{unif}}|=y)}{n(y,z)}\frac{\mathbb{P}(|\mathbf{e}_V^{\text{unif}}|=y)}{n(y)} \tag{12}$$

$$\begin{aligned} &= \mathbb{P}(\mathbf{e}_U^{\text{unif}}=\mathbf{x}_U \mid \mathbf{e}_V^{\text{unif}}=\mathbf{x}_V)\mathbb{P}(\mathbf{e}_V^{\text{unif}}=\mathbf{x}_V)\\ &= \mathbb{P}(\mathbf{e}^{\text{unif}}=\mathbf{x}). \end{aligned} \tag{13}$$

Equation (12) 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 $\textsc{DecodeV}(\cdot)$ and based on the pair $(|\mathbf{e}_V|, m_1(\mathbf{e}))$ for $\textsc{DecodeU}(\cdot)$. 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 $\textsc{DecodeV}(\cdot)$ and a two-dimensional rejection vector $\mathbf{r}_U$ for the values of $(|\mathbf{e}_V|, m_1(\mathbf{e}))$ for $\textsc{DecodeU}(\cdot)$. The corresponding algorithm is specified in Algorithm 3.

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

1: **repeat**
2: $\quad \mathbf{e}_V \leftarrow \textsc{DecodeV}(\mathbf{H}_V, \mathbf{s}^V)$
3: **until** $\text{rand}([0,1]) \le \mathbf{r}_V(|\mathbf{e}_V|)$
4: **repeat**
5: $\quad \mathbf{e}_U \leftarrow \textsc{DecodeU}(\mathbf{H}_U, \varphi, \mathbf{s}^U, \mathbf{e}_V)$
6: $\quad \mathbf{e} \leftarrow \varphi(\mathbf{e}_U, \mathbf{e}_V)$
7: **until** $\text{rand}([0,1]) \le \mathbf{r}_U(|\mathbf{e}_V|, m_1(\mathbf{e}))$
8: **return** $\mathbf{e}$

Standard results on rejection sampling yield the following proposition:

Proposition 4. *Let,*

$$q_1(i) \triangleq \mathbb{P}\left(|\mathbf{e}_V| = i\right) \;;\; q_1^{\text{unif}}(i) \triangleq \mathbb{P}\left(|\mathbf{e}_V^{\text{unif}}| = i\right) \tag{14}$$

$$q_2(s,t) \triangleq \mathbb{P}\left(m_1(\mathbf{e}) = s \mid |\mathbf{e}_V| = t\right) \;;\; q_2^{\text{unif}}(s,t) \triangleq \mathbb{P}\left(m_1(\mathbf{e}^{\text{unif}}) = s \mid |\mathbf{e}_V^{\text{unif}}| = t\right) \tag{15}$$

for any $i,t \in [\![0, n/2]\!]$ and $s \in [\![0,t]\!]$. Let $\mathbf{r}_V$ and $\mathbf{r}_U$ be defined as

$$r_V(i) \triangleq \frac{1}{M_V^{rs}}\frac{q_1^{\text{unif}}(i)}{q_1(i)} \quad \textit{and} \quad r_U(s,t) \triangleq \frac{1}{M_U^{rs}(t)}\frac{q_2^{\text{unif}}(s,t)}{q_2(s,t)}$$

with

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

Then if $\textsc{DecodeV}(\cdot)$ is weightwise uniform and $\textsc{DecodeU}(\cdot)$ is m_1-uniform, the output $\mathbf{e}$ of Algorithm 3 satisfies

$$\mathbf{e} \sim \mathbf{e}^{\text{unif}}.$$

4.2 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 (the proof is given in Appendix A) 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,*

$$q_1^{\text{unif}}(i) = \frac{\binom{n/2}{i}}{\binom{n}{w} 2^{w/2}} \sum_{\substack{p=0 \\ w+p \equiv 0 \mod 2}}^{i} \binom{i}{p} \binom{n/2-i}{(w+p)/2-i} 2^{3p/2} \tag{16}$$

$$q_2^{\text{unif}}(s,t) = \begin{cases} \dfrac{\binom{t}{s}\binom{n/2-t}{\frac{w+s}{2}-t} 2^{\frac{3s}{2}}}{\sum_p \binom{t}{p}\binom{n/2-t}{\frac{w+p}{2}-t} 2^{\frac{3p}{2}}} & \text{if } w+s \equiv 0 \mod 2. \\ 0 & \text{else} \end{cases} \tag{17}$$

Algorithms DecodeV$(\cdot)$, DecodeU$(\cdot)$ are described in Algorithms 4 and 5. They use the rejection vectors given in Proposition 4 which is based on the expressions given in Proposition 5.

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

1: $\mathcal{J}, \mathcal{I} \leftarrow$ FreeSet$(\mathbf{H}_V)$
2: $\ell \hookleftarrow \mathcal{D}_V$
3: $\mathbf{x}_V \hookleftarrow \left\{ \mathbf{x} \in \mathbb{F}_3^{n/2} \mid |\mathbf{x}_{\mathcal{J}}| = \ell, \text{Supp}(\mathbf{x}) \subseteq \mathcal{I} \right\}$ ▷ $(\mathbf{x}_V)_{\mathcal{I}\setminus\mathcal{J}}$ is random
4: $\mathbf{e}_V \leftarrow$ RangeStep$(\mathbf{H}_V, \mathbf{s}^V, \mathcal{I}, \mathbf{x}_V)$
5: **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 $k-d$
1: **repeat**
2: $\quad \mathcal{J} \hookleftarrow [\![1, n]\!]$ of size $k-d$
3: **until** the rank of the columns of $\mathbf{H}$ indexed by $[\![1, n]\!]\setminus\mathcal{J}$ is $n-k$
4: **repeat**
5: $\quad \mathcal{J}' \hookleftarrow [\![1, n]\!]\setminus\mathcal{J}$ of size d
6: $\quad \mathcal{I} \leftarrow \mathcal{J} \sqcup \mathcal{J}'$
7: **until** $\mathcal{I}$ is an information set of $\langle \mathbf{H} \rangle^\perp$
8: **return** $\mathcal{J}, \mathcal{I}$

These two algorithms both use the Prange decoder in the same way as we did with the procedure described in §3.3 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 DecodeV$(\cdot)$ and 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, $k'_V \triangleq k_V - d$ and $k'_U \triangleq k_U - d$. Let X_V (resp. X_U^t) be a random variable distributed according to $\mathcal{D}_V$ (resp. $\mathcal{D}_U^t$). We have,*

$$q_1(i) = \sum_{t=0}^{i} \frac{\binom{n/2-k'_V}{i-t} 2^{i-t}}{3^{n/2-k'_V}} \mathbb{P}(X_V = t) \tag{18}$$

$$q_2(s,t) = \begin{cases} \displaystyle\sum_{\substack{t+k'_U-n/2 \leq k_{\neq 0} \leq t \\ k_0 \triangleq k'_U - k_{\neq 0}}} \frac{\binom{t-k_{\neq 0}}{s}\binom{n/2-t-k_0}{\frac{w+s}{2}-t-k_0} 2^{\frac{3s}{2}}}{\sum_p \binom{t-k_{\neq 0}}{p}\binom{n/2-t-k_0}{\frac{w+p}{2}-t-k_0} 2^{\frac{3p}{2}}} \mathbb{P}(X_U^t = k_{\neq 0}) & \text{if } w \equiv s \bmod 2. \\ 0 & \text{else} \end{cases} \tag{19}$$

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 $|\varphi(\mathbf{e}_U, \mathbf{e}_V)| = w$.

$t \leftarrow |\mathbf{e}_V|$
$k_{\neq 0} \hookleftarrow \mathcal{D}_U^t$
$k_0 \leftarrow k'_U - k_{\neq 0}$ ▷ $k'_U \triangleq k_U - d$
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 $k_{\neq 0}$
  $\mathcal{J}_2 \hookleftarrow [\![1, n]\!] \backslash \mathrm{Supp}(\mathbf{x})$ of size $k - d - k_{\neq 0}$.
  $\mathcal{J} \leftarrow \mathcal{J}_1 \sqcup \mathcal{J}_2$
until the rank of the columns of $\mathbf{H}$ indexed by $[\![1, n]\!] \backslash \mathcal{J}$ is $n - k$
repeat
  $\mathcal{J}' \hookleftarrow [\![1, n]\!] \backslash \mathcal{J}$ 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. We are going to prove that,

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*

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

where the probability is taken over the choice of matrices $\mathbf{H}_U$ *and* $\mathbf{H}_V$*.*

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. This result will be used to select the parameter d instead of the previous theorem.

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. These new decoders will 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 $\mathbf{e}_V\mathbf{H}_V^\intercal = \mathbf{s}^V$ or that the $\mathbf{e}_U$ that is output by VarDecodeU$(\cdot)$ satisfies $\mathbf{e}_U\mathbf{H}_U^\intercal = \mathbf{s}^U$. 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* $\mathbf{e}^{\text{unif}}$ *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* $I^{\text{unif}}_{\mathbf{x}_V,\ell}$ *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:*

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

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 [GM02, 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 $\mathbf{x}_U \in \mathbb{F}_3^{n/2}$,

$$p(\mathbf{x}_V) \triangleq \mathbb{P}\left(\mathbf{e}_V = \mathbf{x}_V\right) \quad ; \quad q(\mathbf{x}_V) \triangleq \mathbb{P}\left(\mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right)$$

$$p(\mathbf{x}_U|\mathbf{x}_V) \triangleq \mathbb{P}\left(\mathbf{e}_U = \mathbf{x}_U \mid \mathbf{e}_V = \mathbf{x}_V\right) \quad ; \quad q(\mathbf{x}_U|\mathbf{x}_V) \triangleq \mathbb{P}\left(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right).$$

We have,

$$\begin{aligned}
\rho\left(\mathbf{e}; \mathbf{e}^{\text{unif}}\right) &= \rho\left(\mathbf{e}_U, \mathbf{e}_V; \mathbf{e}_U^{\text{unif}}, \mathbf{e}_V^{\text{unif}}\right) \\
&= \frac{1}{2} \sum_{\mathbf{x}_V,\mathbf{x}_U} |p(\mathbf{x}_V)p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_V)q(\mathbf{x}_U|\mathbf{x}_V)| \\
&= \frac{1}{2} \sum_{\mathbf{x}_V,\mathbf{x}_U} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))p(\mathbf{x}_U|\mathbf{x}_V) + (p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V))q(\mathbf{x}_V)| \\
&\leq \frac{1}{2} \sum_{\mathbf{x}_V,\mathbf{x}_U} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))p(\mathbf{x}_U|\mathbf{x}_V)| + |(p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V)q(\mathbf{x}_V)| \\
&= \frac{1}{2} \sum_{\mathbf{x}_V} |(p(\mathbf{x}_V) - q(\mathbf{x}_V))| + \frac{1}{2} \sum_{\mathbf{x}_V,\mathbf{x}_U} |p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V)|\, q(\mathbf{x}_V) \qquad (20)
\end{aligned}$$

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:

$$\frac{1}{2}\sum_{\mathbf{x}_V} |p(\mathbf{x}_V) - q(\mathbf{x}_V)| \leq \rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) \tag{21}$$

as the internal distribution $\mathcal{D}_V$ of DecodeV$(\cdot)$ is independent of $J^{\mathbf{H}_V}$ and J^{unif}. Let us upperbound 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,

$$\mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V = \mathbf{x}_V\right) = \mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V\right) \tag{22}$$

Let us define,

$$p(\mathbf{x}_U|\mathbf{x}_V, \ell) \stackrel{\triangle}{=} \mathbb{P}(\mathbf{e}_U = \mathbf{x}_U \mid k_{\neq 0} = \ell, \mathbf{e}_V = \mathbf{x}_V) \quad ; \quad q(\mathbf{x}_U|\mathbf{x}_V, \ell) \stackrel{\triangle}{=} \mathbb{P}(\mathbf{e}_U^{\text{unif}} = \mathbf{x}_U \mid k_{\neq 0} = \ell, \mathbf{e}_V^{\text{unif}} = \mathbf{x}_V).$$

With this notation we obtain from (22)

$$p(\mathbf{x}_U|\mathbf{x}_V) - q(\mathbf{x}_U|\mathbf{x}_V) = \sum_{\ell} \left(p(\mathbf{x}_U|\mathbf{x}_V, \ell) - q(\mathbf{x}_U|\mathbf{x}_V, \ell)\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V = \mathbf{x}_V\right) \tag{23}$$

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:

$$\frac{1}{2}\sum_{x_U} |p(\mathbf{x}_U|\mathbf{x}_V, \ell) - q(\mathbf{x}_U|\mathbf{x}_V, \ell)| \leq \rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right) \tag{24}$$

Combining Equations (20), (21), (23) and (24) 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*

$$\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) = \frac{\#\{\textit{subsets of } [\![1, n/2]\!] \textit{ of size } k-d \textit{ bad for } \mathbf{H}\}}{\binom{n/2}{k-d}} \tag{25}$$

$$\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right) = \frac{N_{\mathbf{x},\ell}}{\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}} \tag{26}$$

$$\mathbb{E}\left\{\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right)\right\} \leq \frac{3^{-d}}{2} \tag{27}$$

$$\mathbb{E}\left\{\rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right)\right\} \leq \frac{3^{-d}}{2},, \tag{28}$$

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

Proof. (25) (26) 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

$$N = \sum_{i=1}^{\binom{n/2}{k-d}} X_i. \tag{29}$$

Recall now (for a proof see Lemma 8 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 (27). (28) follows from similar arguments. □

We are ready now to prove Theorem 1.

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

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

□

4.3 Instantiating the Distributions

Any choice for the distributions $\mathcal{D}_V$ and $\mathcal{D}_U^t$ in Algorithms 4 and 5 will enable uniform sampling by a proper choice of the rejection vectors $\mathbf{r}_V$ and $\mathbf{r}_U$ in Algorithm 3. We argue here, through a case study, that an appropriate choice of the distributions may considerably reduce the rejection rate. In fact, what matters is to have the smallest possible values for M_V^{rs} and $M_U^{\text{rs}}(t)$ in Proposition 4.

The first step to achieve this is to correctly align the distributions to their targets, we do that by a proper choice for the mean value or of the mode (*i.e.* maximum value) of the distributions. Next we choose a "shape" for the distributions. Here we will take (generalized and truncated) Laplace distributions with a prescribed mean and parameterize them to minimize rejection.

For typical parameters with 128 bits of classical security, we will give a case study with the above strategy, in which the total rejection rate is below 1%.

Aligning the Distributions:

1. For the distribution $\mathcal{D}_V$. The output of Algorithm 4 has an average weight $\bar{\ell}+2/3(n/2-k_V+d)$, where $\bar{\ell}$ denotes the mean of $\mathcal{D}_V$. It must be close to $\mathbb{E}(|\mathbf{e}_V^{\text{unif}}|)$. We will admit $\mathbb{E}(|\mathbf{e}_V^{\text{unif}}|) = \sum_{i=0}^{n/2} i q_V^{\text{unif}}(i) = \frac{n}{2}\left(1 - \left(1 - \frac{w}{n}\right)^2 - \frac{1}{2}\left(\frac{w}{n}\right)^2\right)$. The mean value $\bar{\ell}$ of $\mathcal{D}_V$ is chosen (close to) $(1-\alpha)(k_V - d)$ where $\alpha \in [0,1]$ is defined as follows

$$(1-\alpha)(k_V - d) = \frac{n}{2}\left(1 - \left(1 - \frac{w}{n}\right)^2 - \frac{1}{2}\left(\frac{w}{n}\right)^2\right) - \frac{2}{3}\left(\frac{n}{2} - k_V + d\right). \tag{30}$$

2. For the distribution $\mathcal{D}_U^t$, $0 \leq t \leq n/2$. Here, for every t, we want to align the functions $s \mapsto q_2(s,t)$ and $s \mapsto q_2^{\text{unif}}(s,t)$ (see Proposition 4). We get a very good estimate of the s which maximizes $q_2^{\text{unif}}(s,t)$ by solving numerically the equation $q_2^{\text{unif}}(s-1,t) = q_2^{\text{unif}}(s+1,t)$, that is

$$\frac{8\,(t-s)\,(t-s+1)\,(n-w-s+1)}{(s+1)\,s\,(w+s+1-2\,t)} = 1$$

We will denote $m_{\text{target}}^{\max}(t)$ the unique real positive root of the above polynomial equation.
We use the notations of Algorithm 5, with in addition $\mathbf{e} = \varphi(\mathbf{e}_U, \mathbf{e}_V)$. We now have to determine which value of $k_{\neq 0}$ (line 2) will be such that $q_2(s,t)$ also reaches its maximum for

$s = m^{\max}_{\text{target}}(t)$. For a given t, $q_2(s,t)$ is the probability to have $m_1(\mathbf{e}) = s$. This number counts the pairs $(i, i+n/2)$ with $i \in [\![0, n/2]\!]$ such that exactly one of $\mathbf{e}(i)$ and $\mathbf{e}(i+n/2)$ is non-zero. This may only happen when $i \in \mathrm{Supp}(\mathbf{e}_V) \setminus \mathcal{J}$, in which case $\mathbf{e}(i)$ and $\mathbf{e}(i+n/2)$ are two random distinct elements of $\mathbb{F}_3$ and this particular i is counted in $m_1(\mathbf{e})$ with probability $2/3$. Since $|\mathrm{Supp}(\mathbf{e}_V) \setminus \mathcal{J}| = t - k_{\neq 0}$, we typically have $m_1(\mathbf{e}) = \frac{2}{3}(t - k_{\neq 0})$ and the best alignment is reached when the most probable output of distribution $\mathcal{D}^t_U$ is $k_{\neq 0} = t - \frac{3}{2} m^{\max}_{\text{target}}(t)$.

Matching the "Shapes": to avoid a high rejection rate we need to choose distributions so that the tails of the emulated q_1 and q_2 are not lower than their respective targets. A bad choice in this respect could lead to values of M^{rs}_V and $M^{\text{rs}}_U(t)$ growing exponentially with the block size. We chose generalized and truncated Laplace distributions to avoid this.

Definition 5 (Generalized Truncated Discrete Laplace Distribution). *Let μ, σ, β be positive real numbers, let a and b be two integers. We say that a random variable X is distributed according to the Generalized Truncated Discrete Laplace Distribution of parameters μ, σ, β, a, b, which is denoted $X \sim \mathrm{Lap}_\beta(\mu, \sigma, a, b)$, if for all $i \in [\![a, b]\!]$,*

$$\mathbb{P}(X = i) = \frac{e^{-\left(\frac{|i-\mu|}{\sigma}\right)^\beta}}{N}$$

where N is a normalization factor.

We choose

$$\begin{cases} \mathcal{D}_V = \mathrm{Lap}_{\beta_V}(\mu_V, \sigma_V, 0, k_V - d) \\ \mathcal{D}^t_U = \mathrm{Lap}_{\beta_U(t)}(\mu_U(t), \sigma_U(t), t + k_U - d - n/2, t) \end{cases} \text{ with } \begin{cases} \mu_V \quad\;\; = (1-\alpha)(k_V - d) \\ \mu_U(t) = t - \frac{3}{2} m^{\max}_{\text{target}}(t) + \varepsilon(t) \end{cases}$$

where β_V and σ_V are selected to minimize M^{rs}_V, and $\beta_U(t)$, $\varepsilon(t)$, and $\sigma_U(t)$ are selected to minimize $M^{\text{rs}}_U(t)$. We observed heuristically that the exponents β_V and $\beta_U(t)$ are in the interval $[1, 2]$, and that the alignment offset $\varepsilon(t)$ is in the interval $[0, 2]$.

Case Study: $n = 8482$, $(k_U, k_V) = (3558, 2047)$, $w = 7980$, $\alpha = 0.5748$ and $d = 81$. With $\sigma_V = 30.31$ and $\beta_V = 1.982$, we obtain $M^{\text{rs}}_V \approx 1.000895$. With $\varepsilon = 0.29$ and $\beta_U = 1.788$ identical for all t, the optimal $\sigma_U(t)$ lies in the interval $[7.27, 11.58]$, and we obtain an average value of 1.0086 for $M^{\text{rs}}_U(t)$. The result is marginally better by selecting the best $\beta_U(t)$ and $\varepsilon(t)$ for each t. The total rejection rate is thus below 1%.

4.4 Choosing the parameters

Using the parameter α introduced in (30) in the previous subsection as

$$(1-\alpha)(k_V - d) = \frac{n}{2}\left(1 - \left(1 - \frac{w}{n}\right)^2 - \frac{1}{2}\left(\frac{w}{n}\right)^2\right) - \frac{2}{3}\left(\frac{n}{2} - k_V + d\right).$$

we may define all the system parameters depending only on α, the code rate k/n, d and the block size n

$$w = \left\lfloor n\left(1 - \alpha + \frac{1}{3}\sqrt{(3\alpha - 1)\left(3\alpha + 4\frac{k-2d}{n} - 1\right)}\right)\right\rfloor \tag{31}$$

$$k_V = d + \left\lfloor \frac{n}{2}\frac{3}{3\alpha - 1}\left(\left(1 - \frac{w}{n}\right)^2 + \frac{1}{2}\left(\frac{w}{n}\right)^2 - \frac{1}{3}\right)\right\rfloor \; ; \; k_U = d + \left\lfloor \frac{n}{2}\left(-2 + 3\frac{w}{n}\right)\right\rfloor. \tag{32}$$

5 Achieving Uniform Domain Sampling

The following definition will be useful to understand the structure of normalized generalized $(U, U+V)$-codes.

Definition 6. (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 by:*

$$n_I \stackrel{\triangle}{=} |\{1 \leq i \leq n/2 : b_i d_i = 0\}| .$$

Remark 4. n_I can be viewed as the number of positions in which a codeword of the form $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ is necessarily equal to 0: this comes from the fact that on a position where either $b_i = 0$ or $d_i = 0$, the other one is necessarily different from 0 as $a_i d_i - b_i c_i = 1$. In other words we also have

$$n_I = |\{1 \leq i \leq n/2 : b_i = 0\}| + |\{1 \leq i \leq n/2 : d_i = 0\}| .$$

We denote by $\mathbf{H}_{\text{pk}}$ the public parity-check matrix of a normalized generalized $(U, U+V)$-code as described in §2.2. It turns out that $\mathbf{H}_{\text{pk}}$ has enough randomness in it for making the syndromes associated to it indistinguishable in the strongest possible sense, i.e. statistically, from random syndromes as the following proposition shows. In other words, our scheme achieves the Domain Sampling property of Definition 1. Note that the upper-bound we give here depends on the number n_I we have just introduced.

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*

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

$$\varepsilon = \frac{3^{n-k}}{2^w \binom{n}{w}} + 3^{n/2-k_V} \sum_{j=0}^{n/2} \frac{q_1^{\text{unif}}(j)^2}{2^j \binom{n/2}{j}} + 3^{n/2-k_U} \sum_{j=0}^{n_I} \frac{\binom{n_I}{j}\binom{n-n_I}{w-j}^2}{\binom{n}{w}^2 2^j}$$

where q_1^{unif} is given in Proposition 5 in §4.

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*

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

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

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

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

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

Remark 5. 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 the following variation of the left-over hash lemma (see [BDK+11]) that is adapted to our case: here the hash function to which we apply the left-over hash lemma is defined as $h(\mathbf{e}) = \mathbf{e}\mathbf{H}_{\text{pk}}^{\intercal}$. Functions h do not form a universal family of hash functions (essentially because the distribution of the $\mathbf{H}_{\text{pk}}$'s is not the uniform distribution over $\mathbb{F}_3^{(n-k)\times n}$). However in our case we can still bound ε by a direct computation.

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

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

This lemma is proved in Appendix §C.1. In order to use this lemma to bound the statistical distance we are interested in, we used the following lemma.

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 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 this probability we will use a standard result, namely the following lemma.

Lemma 5. *Let $\mathbf{y}$ be a non-zero vector of $\mathbb{F}_3^n$ and $\mathbf{s}$ an arbitrary element in $\mathbb{F}_3^r$. We choose a matrix $\mathbf{H}$ of size $r \times n$ uniformly at random among the set of $r \times n$ ternary matrices. In this case*

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

Proof. The coefficient of $\mathbf{H}$ at row i and column j is denoted by h_{ij}, whereas the coefficients of $\mathbf{y}$ and $\mathbf{s}$ are denoted by y_i and s_i respectively. The probability we are looking for is the probability to have

$$\sum_j h_{ij} y_j = s_i \tag{33}$$

for all i in $[\![1, r]\!]$. Since $\mathbf{y}$ is non zero, it has at least one non-zero coordinate. Without loss of generality, we may assume that $y_1 = 1$. We may rewrite (33) as $h_{i1} = \sum_{j>1} h_{ij} y_j$. This event happens with probability $\frac{1}{3}$ for a given i and with probability $\frac{1}{3^r}$ on all r events simultaneously due to the independence of the h_{ij}'s. □

This leads us to distinguish between the events:

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

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

Under these events we get thanks to Lemma 5 and $k = k_U + k_V$:

$$\begin{aligned}
&\mathbb{P}_{\mathbf{H}_{\text{sk}},\mathbf{x},\mathbf{y}}\left(\mathbf{x}\mathbf{H}_{\text{sk}}^{\intercal} = \mathbf{y}\mathbf{H}_{\text{sk}}^{\intercal}\right) \\
&= \sum_{i=1}^{4} \mathbb{P}_{\mathbf{H}_{\text{sk}}}\left(\mathbf{x}\mathbf{H}_{\text{sk}}^{\intercal} = \mathbf{y}\mathbf{H}_{\text{sk}}^{\intercal} | \mathcal{E}_i\right) \mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_i\right) \\
&= \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_1\right)}{3^{n/2-k_V}} + \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_2\right)}{3^{n/2-k_U}} + \frac{\mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_3\right)}{3^{n-k}} + \mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_4\right) \\
&\leq \frac{1}{3^{n-k}}\left(1 + 3^{n/2-k_U}\mathbb{P}\left(\mathcal{E}_1\right) + 3^{n/2-k_V}\mathbb{P}\left(\mathcal{E}_2\right) + 3^{n-k}\mathbb{P}(\mathcal{E}_4)\right), \qquad (34)
\end{aligned}$$

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

$$\mathbb{P}_{\mathbf{x},\mathbf{y}}\left(\mathcal{E}_4\right) = \mathbb{P}(\mathbf{x} = \mathbf{y}) = \frac{1}{2^w\binom{n}{w}}. \qquad (35)$$

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:

$$\mathbb{P}\left(\mathcal{E}_2\right) \leq \sum_{j=0}^{n/2} \frac{q_1^{\text{unif}}(j)^2}{2^j\binom{n/2}{j}}. \qquad (36)$$

Let us now estimate the probability of $\mathcal{E}_1$ for which we derive the following upper-bound:

$$\mathbb{P}_{\mathbf{x},\mathbf{y}}(\mathcal{E}_1) \leq \mathbb{P}(\mathbf{x}_U = \mathbf{y}_U)$$

By definition of $\mathbf{x}_U$ and $\mathbf{y}_U$, the event we are looking for is $\{\mathbf{d} \odot (\mathbf{x}_1 - \mathbf{y}_1) = \mathbf{b} \odot (\mathbf{x}_2 - \mathbf{y}_2)\}$ which is the same (up to a permutation of indices of $\mathbf{x}$ and $\mathbf{y}$ and by multiplying some of their component by -1) as the case where we consider:

$$b_1 = \cdots = b_{n_I} = 0 \quad ; \quad b_{n_I+1} = \cdots = b_{n/2} = d_1 = \cdots = d_{n/2} = 1$$

where n_I is the number of blocks of type I. This gives the following probability to upper-bound

$$\mathbb{P}\left(\forall i \in [\![1, n_I]\!],\ (\mathbf{x}_1 - \mathbf{y}_1)(i) = 0, \forall i \in [\![n_I + 1, n/2]\!], (\mathbf{x}_1 - \mathbf{y}_1)(i) = (\mathbf{x}_2 - \mathbf{y}_2)(i)\right)$$

We clearly have:

$$\begin{aligned}
&\mathbb{P}(\forall i \in [\![1, n_I]\!],\ (\mathbf{x}_1 - \mathbf{y}_1)(i) = 0, \forall i \in [\![n_I + 1, n/2]\!],\ (\mathbf{x}_1 - \mathbf{y}_1)(i) = (\mathbf{x}_2 - \mathbf{y}_2)(i)) \\
&\quad \leq \sum_{\mathbf{e} \in \mathbb{F}_3^{n_I}} \mathbb{P}\left(\forall i \in [\![1, n_I]\!], \mathbf{x}_1(i) = \mathbf{e}(i)\right)^2 \\
&\quad \leq \sum_{j=0}^{n_I} \sum_{\mathbf{e}' \in \mathbb{F}_3^{n_I} : |\mathbf{e}'| = j} \mathbb{P}\left(\forall i \in [\![1, n_I]\!], \mathbf{x}_1(i) = \mathbf{e}'(i)\right)^2 \\
&\quad = \sum_{j=0}^{n_I} \sum_{\mathbf{e}' \in \mathbb{F}_3^{n_I} : |\mathbf{e}'| = j} \left(\frac{\binom{n-n_I}{w-j} 2^{w-j}}{\binom{n}{w} 2^w} \right)^2 \\
&\quad = \sum_{j=0}^{n_I} \binom{n_I}{j} 2^j \left(\frac{\binom{n-n_I}{w-j}}{\binom{n}{w} 2^j} \right)^2
\end{aligned}$$

which gives:

$$\mathbb{P}(\mathcal{E}_1) \leq \sum_{j=0}^{n_I} \binom{n_I}{j} 2^{-j} \left(\frac{\binom{n-n_I}{w-j}}{\binom{n}{w}} \right)^2 \tag{37}$$

Therefore, with Equations (34),(35), (36) and (37) we finally conclude the proof. □

6 Security Proof

6.1 Basic Tools

Basic Definitions. A *distinguisher* between two distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ over the same space $\mathcal{E}$ is a randomized algorithm which takes as input an element of $\mathcal{E}$ that follows the distribution $\mathcal{D}^0$ or $\mathcal{D}^1$ and outputs $b \in \{0, 1\}$. It is characterized by its advantage:

$$Adv^{\mathcal{D}^0, \mathcal{D}^1}(\mathcal{A}) \stackrel{\triangle}{=} \mathbb{P}_{\xi \sim \mathcal{D}^0}\left(\mathcal{A}(\xi) \text{ outputs } 1\right) - \mathbb{P}_{\xi \sim \mathcal{D}^1}\left(\mathcal{A}(\xi) \text{ outputs } 1\right).$$

Definition 7 (Computational Distance and Indistinguishability). *The computational distance between two distributions $\mathcal{D}^0$ and $\mathcal{D}^1$ in time t is:*

$$\rho_c\left(\mathcal{D}^0, \mathcal{D}^1\right)(t) \stackrel{\triangle}{=} \max_{|\mathcal{A}| \leq t} \left\{ Adv^{\mathcal{D}^0, \mathcal{D}^1}(\mathcal{A}) \right\}$$

where $|\mathcal{A}|$ denotes the running time of $\mathcal{A}$ on its inputs.

For signature schemes, one of the strongest security notion is *existential unforgeability under an adaptive chosen message attack* (EUF-CMA). In this model the adversary has access to all signatures of its choice and its goal is to produce a valid forgery. A valid forgery is a message/signature pair $(\mathbf{m}, \sigma)$ such that $\mathtt{Vrfy}^{\mathrm{pk}}(\mathbf{m}, \sigma) = 1$ whereas the signature of $\mathbf{m}$ has never been requested.

Definition 8 (EUF-CMA Security). *A forger $\mathcal{A}$ is a $(t, q_{\mathrm{hash}}, q_{\mathrm{sign}}, \varepsilon)$-adversary in* EUF-CMA *against a signature scheme $\mathcal{S}$ if after at most q_{hash} queries to the hash oracle, q_{sign} signatures queries and t working time, it outputs a valid forgery with probability at least ε. The* EUF-CMA *success probability against $\mathcal{S}$ is:*

$$Succ_{\mathcal{S}}^{\text{EUF-CMA}}(t, q_{\mathrm{hash}}, q_{\mathrm{sign}}) \stackrel{\triangle}{=} \max\left(\varepsilon | \textit{it exists a } (t, q_{\mathrm{hash}}, q_{\mathrm{sign}}, \varepsilon)\textit{-adversary}\right).$$

6.2 Code-Based Problems

We introduce the code-based problems that will be used in the security reduction.

Problem 3. [DOOM – Decoding One Out of Many] For $\mathbf{H} \in \mathbb{F}_3^{(n-k)\times n}$, $\mathbf{s}_1, \cdots, \mathbf{s}_N \in \mathbb{F}_3^{n-k}$, integer w, find $\mathbf{e} \in \mathbb{F}_3^n$ and $i \in [\![1, N]\!]$ such that $\mathbf{e}\mathbf{H}^\intercal = \mathbf{s}_i$ and $|\mathbf{e}| = w$.

We will come back to the best known algorithms to solve this problem as a function of the distance w in §7.1.

Definition 9 (One-Wayness of DOOM). *We define the success of an algorithm $\mathcal{A}$ against* DOOM *with the parameters n, k, N, w as:*

$$Succ^{n,k,N,w}_{\text{DOOM}}(\mathcal{A}) = \mathbb{P}\big(\mathcal{A}(\mathbf{H}, \mathbf{s}_1, \cdots, \mathbf{s}_N) \text{ solution of DOOM}\big)$$

where $\mathbf{H} \hookleftarrow \mathbb{F}_3^{(n-k)\times n}$, $\mathbf{s}_i \hookleftarrow \mathbb{F}_3^{n-k}$ and the probability is taken over $\mathbf{H}$, the $\mathbf{s}_i$'s and the internal coins of $\mathcal{A}$. The computational success in time t of breaking DOOM *with the parameters n, k, N, w is then defined as:*

$$Succ^{n,k,N,w}_{\text{DOOM}}(t) = \max_{|\mathcal{A}|\le t} \left\{ Succ^{n,k,N,w}_{\text{DOOM}}(\mathcal{A}) \right\}.$$

Another problem appears in the security proof: distinguish random codes from a code drawn uniformly at random in the family used for public keys in the signature scheme. In what follows $\mathcal{D}_{\text{pub}}$ denotes the distribution of public keys $\mathbf{H}_{\text{pk}}$ whereas $\mathcal{D}_{\text{rand}}$ denotes the uniform distribution over $\mathbb{F}_3^{(n-k_U-k_V)\times n}$.

6.3 EUF-CMA Security Proof

Theorem 2. (Security Reduction). *Let q_{hash} (resp. q_{sign}) be the number of queries to the hash (resp. signing) oracle. We assume that $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}})$ where λ is the security parameter of the signature scheme. We have in the random oracle model for all time t, $t_c = t + O\left(q_{\text{hash}} \cdot n^2\right)$ and ε given in Proposition 9:*

$$\begin{aligned} Succ^{\text{EUF-CMA}}_{\mathcal{S}_{\text{Wave}}}(t, q_{\text{hash}}, q_{\text{sign}}) &\le 2Succ^{n,k,q_{\text{hash}},w}_{\text{DOOM}}(t_c) + \rho_c\left(\mathcal{D}_{\text{rand}}, \mathcal{D}_{\text{pub}}\right)(t_c) \\ &+ q_{\text{sign}}\left(\mathbb{E}_{\mathbf{H}_{\text{pk}}}\left(\rho\left(\mathcal{D}_w^{\mathbf{H}_{\text{pk}}}, \mathcal{U}_w\right)\right) + \frac{\sqrt{\varepsilon}}{2} + \frac{q_{\text{hash}} + q_{\text{sign}}}{q_{\text{sign}}^2 \times 2^\lambda}\right) + \frac{1}{2}(q_{\text{hash}} + q_{\text{sign}})\sqrt{\varepsilon} + \frac{1}{2^\lambda} \end{aligned}$$

where $\mathcal{D}_w^{\mathbf{H}_{\text{pk}}}$ is the distribution sampled as follows:

- $\mathbf{s} \hookleftarrow \mathbb{F}_3^{n-k}$, $\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$, $\mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}(\mathbf{S}^{-1})^\intercal)$, *output* $(\mathbf{eP}, \mathbf{r})$.

with $D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}$ the Algorithm 3 using Algorithms 4 and 5 and $\mathcal{U}_w$ is the uniform distribution over S_w.

This theorem is proved in Appendix E.

7 Security Assumptions and Parameter Selection

Our scheme is secure under two security assumptions. One relates to the hardness decoding and the other to the indistinguishability of generalized $(U, U+V)$-codes.

7.1 Message Attack – Hardness of Decoding

Here we are interested in the hardness of the DOOM problem as stated in Problem 3 for the case $q = 3$ when the target weight w is large. This variant of the problem, including the multiple target (DOOM) aspect, was recently investigated in [BCDL19]. This work adapted to this setting the best generic decoding techniques [Dum91, Ste88, MMT11, BJMM12] which use the so-called PGE+SS framework ("Partial Gaussian Elimination and Subset Sum"). It also uses Wagner's generalized birthday algorithm [Wag02] and the representation technique [HJ10].

7.2 Key Attack – Indistinguishability of generalized $(U, U+V)$-Codes

Here we are interested in the hardness of the problem to distinguish random codes from permuted generalized normalized $(U, U+V)$-code. All the proofs of this subsection are in Appendix D.

A normalized generalized $(U, U+V)$-code where U and V are random seems very close to a random linear code. There is for instance only a very slight difference between the weight distribution of a random linear code and the weight distribution of a random normalized generalized $(U, U+V)$-code of the same length and dimension. This slight difference happens for small and large weights and is due to codewords where $\mathbf{v} = \mathbf{0}$ or $\mathbf{u} = \mathbf{0}$ which are of the form $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$ where $\mathbf{u}$ belongs to U or codewords of the form $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ where $\mathbf{v}$ belongs to V as shown by the following proposition:

Proposition 11. *Assume that we choose a normalized generalized $(U, U+V)$-code over $\mathbb{F}_3$ with a number n_I of linear combinations of type I by picking the parity-check matrices of U and V uniformly at random among the ternary matrices of size $(n/2 - k_U) \times n/2$ and $(n/2 - k_V) \times n/2$ respectively. Let $a_{(\mathbf{u},\mathbf{v})}(z)$, $a_{(\mathbf{u},\mathbf{0})}(z)$ and $a_{(\mathbf{0},\mathbf{v})}(z)$ be the expected number of codewords of weight z that are respectively in the normalized generalized $(U, U+V)$-code, of the form $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$ where $\mathbf{u}$ belongs to U and of the form $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ where $\mathbf{v}$ belongs to V. These numbers are given for even z in $[\![0, n]\!]$ by*

$$a_{(\mathbf{u},\mathbf{0})}(z) = \frac{\binom{n/2}{z/2} 2^{z/2}}{3^{n/2-k_U}} \quad ; \quad a_{(\mathbf{0},\mathbf{v})}(z) = \frac{1}{3^{n/2-k_V}} \sum_{\substack{j=0 \\ j \text{ even}}}^{z} \binom{n_I}{j} \binom{n/2-n_I}{\frac{z-j}{2}} 2^{(z+j)/2}$$

$$a_{(\mathbf{u},\mathbf{v})}(z) = a_{(\mathbf{u},\mathbf{0})}(z) + a_{(\mathbf{0},\mathbf{v})}(z) + \frac{1}{3^{n-k_U-k_V}} \left(\binom{n}{z} 2^z - \binom{n/2}{z/2} 2^{z/2} - \sum_{\substack{j=0 \\ j \text{ even}}}^{z} \binom{n_I}{j} \binom{n/2-n_I}{\frac{z-j}{2}} 2^{(z+j)/2} \right)$$

and for odd $z \in [\![0, n]\!]$ by

$$a_{(\mathbf{u},\mathbf{0})}(z) = 0 \quad ; \quad a_{(\mathbf{0},\mathbf{v})}(z) = \frac{1}{3^{n/2-k_V}} \sum_{\substack{j=0 \\ j \text{ odd}}}^{z} \binom{n_I}{j} \binom{n/2-n_I}{\frac{z-j}{2}} 2^{(z+j)/2}$$

$$a_{(\mathbf{u},\mathbf{v})}(z) = a_{(\mathbf{0},\mathbf{v})}(z) + \frac{1}{3^{n-k_U-k_V}} \left(\binom{n}{z} 2^z - \sum_{\substack{j=0 \\ j \text{ odd}}}^{z} \binom{n_I}{j} \binom{n/2-n_I}{\frac{z-j}{2}} 2^{(z+j)/2} \right)$$

On the other hand, when we choose a linear code of length n over $\mathbb{F}_3$ with a random parity-check matrix of size $(n - k_U - k_V) \times n$ chosen uniformly at random, then the expected number $a(z)$ of codewords of weight $z > 0$ is given by

$$a(z) = \frac{\binom{n}{z} 2^z}{3^{n-k_U-k_V}}.$$

We have plotted in Figure 5 the normalized logarithm of the density of codewords of the form $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$ and $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ of relative *even* weight $x \triangleq \frac{z}{n}$ against x in the case where U is of rate $\frac{k_U}{n/2} = 0.7$, V is of rate $\frac{k_V}{n/2} = 0.3$ and $\frac{n_I}{n/2} = \frac{1}{2}$. These two relative densities are defined respectively by

$$\alpha_{\mathbf{u}}(z/n) \triangleq \frac{\log_2(a_{(\mathbf{u},\mathbf{0})}(z)/a_{(\mathbf{u},\mathbf{v})}(z))}{n} \quad ; \quad \alpha_{\mathbf{v}}(z/n) \triangleq \frac{\log_2(a_{(\mathbf{0},\mathbf{v})}(z)/a_{(\mathbf{u},\mathbf{v})}(z))}{n}$$

We see that for a relative weight z/n below approximately 0.26 almost all the codewords are of the form $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$.

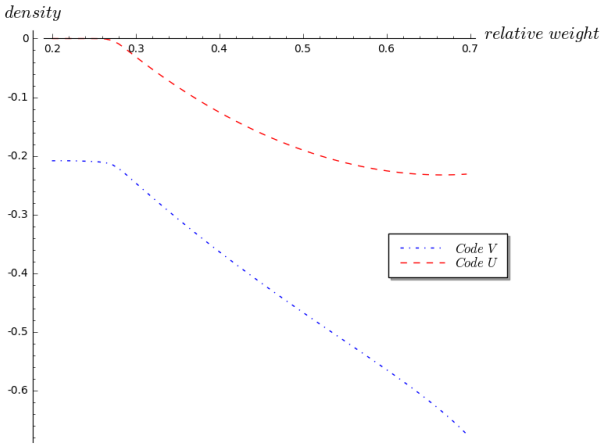


Fig. 5. $\alpha_{\mathbf{u}}(z/n)$ and $\alpha_{\mathbf{v}}(z/n)$ against $x \stackrel{\triangle}{=} \frac{z}{n}$.

Since the weight distribution is invariant by permuting the positions, this slight difference also survives in the permuted version of the normalized generalized $(U, U+V)$-code. These considerations lead to the best attack we have found for recovering the structure of a permuted normalized generalized $(U, U+V)$-code. It consists in applying known algorithms aiming at recovering low weight codewords in a linear code. We run such an algorithm until getting at some point either a permuted $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$ codeword where $\mathbf{u}$ is in U or a permuted $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ codeword where $\mathbf{v}$ belongs to V. The rationale behind this algorithm is that the density of codewords of the form $(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})$ or $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ is bigger when the weight of the codeword gets smaller.

Once we have such a codeword we can bootstrap from there very similarly to what has been done in [OT11, Subs. 4.4]. Note that this attack is actually very close in spirit to the attack that was devised on the KKS signature scheme [OT11]. In essence, the attack against the KKS scheme really amounts to recover the support of the V code. The difference with the KKS scheme is that the support of V is much bigger in our case. As explained in the conclusion of [OT11] the attack against the KKS scheme has in essence an exponential complexity. This exponent becomes really prohibitive in our case when the parameters of U and V are chosen appropriately as we will now explain. Let us first introduce the following notation that will be useful in the following.

Punctured Code. For a subset $\mathcal{I} \subset [\![1, n]\!]$ and a code $\mathcal{C}$ of length n, we denote by $\mathrm{Punc}_{\mathcal{I}}(\mathcal{C})$, the code $\mathcal{C}$ punctured in $\mathcal{I}$, namely $\{\mathbf{c}_{\bar{\mathcal{I}}} = (c_j)_{j \in [\![1,n]\!] \setminus \mathcal{I}} : \mathbf{c} \in \mathcal{C}\}$. In other words, the set of vectors obtained by deleting in the codewords of $\mathcal{C}$ the positions that belong to $\mathcal{I}$.

Recovering the U Code up to Permutation. We consider here the permuted code

$$U' \stackrel{\triangle}{=} (\mathbf{a} \odot U, \mathbf{c} \odot U)\mathbf{P} = \{(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u})\mathbf{P} : \mathbf{u} \in U\}.$$

The attack in this case consists in recovering a basis of U'. Once this is done, it is easy to recover the U code up to permutation by matching the pairs of coordinates which are either always equal or always sum to 0 in U'. The basic algorithm for recovering the code U' is given in Algorithm 6.

It uses other auxiliary functions

- $\textsc{Codewords}(\mathrm{Punc}_{\mathcal{I}}(\mathcal{C}_{\mathrm{pk}}), p)$ which computes all (or a big fraction of) codewords of weight p of the punctured public code $\mathrm{Punc}_{\mathcal{I}}(\mathcal{C}_{\mathrm{pk}})$. All modern [Dum91, FS09, MMT11, BJMM12, MO15] algorithms for decoding linear codes perform such a task in their inner loop.
- $\textsc{Complete}(\mathbf{x}, \mathcal{I}, \mathcal{C}_{\mathrm{pk}})$ which computes the codeword $\mathbf{c}$ in $\mathcal{C}_{\mathrm{pk}}$ such that its restriction outside $\mathcal{I}$ is equal to $\mathbf{x}$.
- $\textsc{CheckU}(\mathbf{x})$ which checks whether $\mathbf{x}$ belongs to U'.

Algorithm 6 ComputeU: algorithm that computes a set of independent elements in U'.

Parameters: (i) ℓ : small integer (typically $\ell \leqslant 40$),
(ii) p : very small integer (typically $1 \leqslant p \leqslant 10$).
Input: (i) $\mathcal{C}_{\text{pk}}$ the public code used for verifying signatures.
(ii) N a certain number of iterations
Output: an independent set of elements in U'

1: **function** ComputeU($\mathcal{C}_{\text{pk}}$,N)
2: **for** $i = 1, \ldots, N$ **do**
3: $B \leftarrow \emptyset$
4: Choose a set $\mathcal{I} \subset [\![1, n]\!]$ of size $n - k - \ell$ uniformly at random
5: $\mathcal{L} \leftarrow$ Codewords($\text{Punc}_{\mathcal{I}}(\mathcal{C}_{\text{pk}}), p$)
6: **for all** $\mathbf{x} \in \mathcal{L}$ **do**
7: $\mathbf{x} \leftarrow$ Complete($\mathbf{x}, \mathcal{I}, \mathcal{C}_{\text{pk}}$)
8: **if** CheckU($\mathbf{x}$) **then**
9: add $\mathbf{x}$ to B if $\mathbf{x} \notin < B >$
10: **return** B

Choosing N Appropriately. Let us first analyse how we have to choose N such that ComputeU returns $\Omega(1)$ elements. This is essentially the analysis which can be found in [OT11, §5.2].

Proposition 12. *The probability P_{succ} that one iteration of the for loop (Instruction 2) in* ComputeU *adds elements to the list B is lower-bounded by*

$$P_{succ} \geq \sum_{z=0}^{n/2} \frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z} 2^{k+\ell-2z}}{\binom{n}{k+\ell}} f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right) \tag{38}$$

where f is the function defined by $f(x) \stackrel{\triangle}{=} \max\left(x(1-x/2), 1-\frac{1}{x}\right)$. Algorithm 6 returns a non zero list with probability $\Omega(1)$ when N is chosen as $N = \Omega\left(\frac{1}{P_{succ}}\right)$.

Complexity of Recovering a Permuted Version of U. The complexity of a call to ComputeU can be estimated as follows. We denote the complexity of computing the list of codewords of weight p in a code of length $k + \ell$ and dimension k by $C_1(p, k, \ell)$. It depends on the particular algorithm used here. For more details see [Dum91, FS09, MMT11, BJMM12, MO15]. This is the complexity of the call Codewords($\text{Punc}_{\mathcal{I}}(\mathcal{C}_{\text{pk}}), p$) in Step 5 in Algorithm 6. The complexity of ComputeU and hence the complexity of recovering a permuted version of U is clearly lower bounded by $\Omega\left(\frac{C_1(p,k,\ell)}{P_{\text{succ}}}\right)$. It turns out that the whole complexity of recovering a permuted version of U is actually of this order, namely $\Theta\left(\frac{C_1(p,k,\ell)}{P_{\text{succ}}}\right)$. This can be done by a combination of two techniques

- Once a non-zero element of U' has been identified, it is much easier to find other ones. This uses one of the tricks for breaking the KKS scheme (see [OT11, Subs. 4.4]). The point is the following: if we start again the procedure ComputeU, but this time by choosing a set $\mathcal{I}$ on which we puncture the code which contains the support of the codeword that we already found, then the number N of iterations that we have to perform until finding a new element is negligible when compared to the original value of N.
- The call to CheckU can be implemented in such a way that the additional complexity coming from all the calls to this function is of the same order as the N calls to Codewords. The strategy to adopt depends on the values of the dimensions k and k_U. In certain cases, it is easy to detect such codewords since they have a typical weight that is significantly smaller than the other codewords. In more complicated cases, we might have to combine a technique checking first the weight of $\mathbf{x}$, if it is above some prescribed threshold, we decide that it is not in U', if it is below the threshold, we decide that it is a suspicious candidate and use then the

previous trick. We namely check whether the support of the codeword $\mathbf{x}$ can be used to find other suspicious candidates much more quickly than performing N calls to CheckU.

To keep the length of this paper within some reasonable limit we avoid here giving the analysis of those steps and we will just use the aforementioned lower bound on the complexity of recovering a permuted version of U.

Recovering the V Code up to a Permutation We consider here the permuted code

$$V' \stackrel{\triangle}{=} (\mathbf{b} \odot V, \mathbf{d} \odot V)\mathbf{P} = \{(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})\mathbf{P} \text{ where } \mathbf{v} \in V\}.$$

The attack in this case consists in recovering a basis of V'. Once this is achieved, the support $\mathrm{Supp}(V')$ of V' can easily be obtained. Recall that this is the set of positions for which there exists at least one codeword of V' that is non-zero in this position. This allows to easily recover the code V up to some permutation. The algorithm for recovering V' is the same as the algorithm for recovering U'. We call the associated function ComputeV though since they differ in the choice for N. The analysis is slightly different indeed.

Choosing N Appropriately. As in the previous subsection let us analyse how we have to choose N in order that ComputeV returns $\Omega(1)$ elements of V'. We have in this case the following result.

Proposition 13. *The probability P_{succ} that one iteration of the for loop (Instruction 2) in* ComputeV *adds elements to the list B is lower-bounded by*

$$P_{succ} \geq \sum_{z=0}^{\min(n-k-\ell, n-n_I)} \sum_{m=0}^{n/2-n_I} \frac{\binom{\frac{n}{2}-n_I}{m}\binom{n_I}{n-k-\ell-z}}{\binom{n}{n-k-\ell}} \max_{i=0}^{\lfloor p/2 \rfloor} f\left(\frac{\binom{n-n_I-z-2m}{p-2i}\binom{m}{i}2^{p-i}}{3^{\max(0,n-n_I-z-m-k_V)}}\right) \sum_{j=0}^{n/2-n_I-m} \binom{n/2-n_I-m}{j} 2^j \binom{n_I}{z-n+2n_I+2m+j}$$

where f is the function defined by $f(x) \stackrel{\triangle}{=} \max\left(x(1-x/2), 1-\frac{1}{x}\right)$. ComputeV *returns a non-zero list with probability $\Omega(1)$ when N is chosen as $N = \Omega\left(\frac{1}{P_{succ}}\right)$.*

Complexity of Recovering a Permuted Version of V. As for recovering the permuted U code, the complexity for recovering the permuted V is of order $\Omega\left(\frac{C_1(p,k,\ell)}{P_{\mathrm{succ}}}\right)$.

Distinguishing a Generalized $(U, U+V)$-Code It is not clear in the second case that from the single knowledge of V' and a permuted version of V we are able to find a permutation of the positions which gives to the whole code the structure of a generalized $(U, U+V)$-code. However in both cases as single successful call to ComputeV (resp. ComputeU) is really distinguishing the code from a random code of the same length and dimension. In other words, we have a distinguishing attack whose complexity is given by the following proposition

Proposition 14. *Algorithm 6 lead to a distinguishing attack whose complexity is given by*

$$\min\left(O\left(\min_{p,\ell} C_U(p,\ell)\right), O\left(\min_{p,\ell} C_V(p,\ell)\right)\right)$$

$$C_U(p,\ell) \stackrel{\triangle}{=} \frac{C_1(p,k,\ell)}{\sum_{z=0}^{n/2} \frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z}2^{k+\ell-2z}}{\binom{n}{k+\ell}} \max_{i=0}^{\lfloor p/2 \rfloor} f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right)} \tag{39}$$

$$C_V(p,\ell) \stackrel{\triangle}{=} \frac{C_1(p,k,\ell)}{\sum_{\mathcal{I}} \frac{\binom{\frac{n}{2}-n_I}{m}\binom{n_I}{n-k-\ell-z}}{\binom{n}{n-k-\ell}} \max_{i=0}^{\lfloor p/2 \rfloor} f\left(\frac{\binom{n-n_I-z-2m}{p-2i}\binom{m}{i}2^{p-i}}{3^{\max(0,n-n_I-z-m-k_V)}}\right)\binom{n/2-n_I-m}{j}2^j\binom{n_I}{z-n+2n_I+2m+j}}. \quad (40)$$

where $C_1(p,k,\ell)$ is the the complexity of a computing a constant fraction (say half of them) of the codewords of weight p in a code of length $k+\ell$ and dimension k and f is the function $f(x) \stackrel{\triangle}{=} \max\left(x(1-x/2), 1-\frac{1}{x}\right)$. The sum in the denominator of (40) is over the domain

$$\mathcal{I} \stackrel{\triangle}{=} \{(z,m,j) \mid 0 \le z \le \min(n-k-\ell, n-n_I), 0 \le m \le n/2-n_I, 0 \le j \le n/2-n_I-m\}.$$

We explain in Appendices §D.3 and §D.4 how to estimate C_U and C_V.

7.3 Parameter Selection

With proper rejection sampling, the security of Wave provably reduces to the two previous hard computational problems. The best known solvers, presented above, both have an exponential complexity. For a given set of system parameters $(n, w, k_U, k_V, k = k_U + k_V)$, their asymptotic complexities can be expressed as

- for the message attack, $2^{c_M n(1+o(1))}$ where c_M is a function of w/n and k/n
- for the key attack, $2^{c_K n(1+o(1))}$ where c_K is a function of k_U/n and k_V/n

Using the relations of §4.4, both c_M and c_K can be expressed as functions of the code rate $R = k/n$ and of the parameter α. Minimizing the public key size under the constraint $c_M(R,\alpha) = c_K(R,\alpha)$, we obtain

$$R = 0.660, \alpha = 0.574635, c_M \approx c_K \approx 0.015074.$$

For λ bits of (classical) security we get (K the key size in bits):

$$n = \frac{\lambda}{0.015074} = 66.34\,\lambda, \quad w = 0.9396\,n, \quad k_U = 0.8379\,\frac{n}{2}, \quad k_V = 0.4821\,\frac{n}{2}, \quad K = 1565.0\,\lambda^2$$

To reach 128 bits of security we obtain $n = 8492$, $w = 7980$, $k_U = 3558$, $k_V = 2047$ for a public key size of 3.2 megabytes. We also checked that the other terms in the security reduction do not interfere here. For instance, we recommend to choose the vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$ uniformly at random among the choices that give a φ that is UV-normalized, meaning that for all i in $[\![1, n/2]\!]$ we should have $a_i d_i - b_i c_i = 1$ and $a_i c_i \neq 0$. We reject choices that lead to a number n_I of V blocks of type I that are not close to their expected value $\mathbb{E}(n_I) = n/6$. By doing so we can control the parameter ε giving an upper-bound on $\mathbb{E}_{\mathbf{H}_{\text{pk}}}\left(\rho(\mathcal{D}_w^{\mathbf{H}_{\text{pk}}}, \mathcal{U})\right)$. In the case $n_I = n/6$ this upper-bound is of order $\approx 2^{-177}$.

7.4 Implementation

The scheme was implemented in C as a proof of concept[3]. For the parameters $(n, w) = (8492, 7980)$, one signature is produced in about 0.3 seconds[4]. The parameters of the generalized Laplace distributions are selected as described in §4.3. The Laplace distributions themselves are precomputed with 24 significants bits, one for V and, for U, one for each possible value of t. The corresponding rejection vectors are precomputed as well with 128 significant bits. In total, precomputed data amounts to 1.8 megabytes.

[3] `http://wave.inria.fr`

[4] using a single core of an Intel© Xeon© E3-1240 v5 clocked at 3.5GHz

8 Concluding Remarks and Further Work

We have presented Wave the first code-based "hash-and-sign" signature scheme which strictly follows the GPV strategy [GPV08]. 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 [FHK$^+$17] 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 [BCDL19] adapts the best ISD techniques for low weight [MMT11, BJMM12] and goes even further with a higher order generalized birthday algorithm [Wag02]. 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 [DST17b] 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 [DST17b]. 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 [Sen11] and the decoding techniques for linear codes have a natural extension to DOOM as noticed in [Sen11]. We also note that with our approach, where a message has many possible signatures, we avoid the tightness impossibility results given in [BJLS16] 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. This task requires a better understanding of the distributions involved. This could offer an interesting trade-off in which the designer/signer would have to precompute and store a set of distributions but in exchange would produce a signing algorithm that emulates a uniform distribution without rejection sampling.

Improving Parameters. In order to prove that the distribution of the output of the signing algorithm is almost the uniform distribution over the words of the same length and weight in a very strong sense (i.e. the statistical distance between both distributions should be negligible) we chose to degrade a little bit the signing algorithm. This was achieved by excluding d positions from the information sets. In this case, the distribution of sets that can be completed in d positions to give an information set is almost the same as the uniform distribution over the sets of such size. We use this phenomenon to upper-bound the aforementioned statistical distance (see Theorem 3 in Section 4). However, this is a very crude approach and the upper-bound we obtain in this way is extremely pessimistic. We conjecture that the statistical distance is still negligible even in the case

$d = 0$. Choosing $d = 0$ allows to reduce the block size by more than 10%. For this reason, proving such a conjecture would be an interesting task.

Acknowledgements

We are grateful to Damien Stehlé for his constructive help, in particular for clarifying the link between our definition of "preimage sampleable on average" and the GPV framework [GPV08]. We are also indebted to André Chailloux, Léo Ducas and Thomas Prest for their early interest, insightful suggestions, and unwavering support.

References

[ABB+17] Erdem Alkim, Nina Bindel, Johannes A. Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting TESLA in the quantum random oracle model. In *Post-Quantum Cryptography 2017*, volume 10346 of *LNCS*, pages 143–162, Utrecht, The Netherlands, June 2017. Springer.

[ABG+18] Nicolas Aragon, Olivier Blazy, Philippe Gaborit, Adrien Hauteville, and Gilles Zémor. Durandal: a rank metric based signature scheme. *IACR Cryptology ePrint Archive*, 2018.

[Bar97] Alexander Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity*, October 1997.

[BBC+13] Marco Baldi, Marco Bianchi, Franco Chiaraluce, Joachim Rosenthal, and Davide Schipani. Using LDGM codes and sparse syndromes to achieve digital signatures. In *Post-Quantum Cryptography 2013*, volume 7932 of *LNCS*, pages 1–15. Springer, 2013.

[BCDL19] Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weights. preprint, February 2019. arXiv:1903.07464, to appear in the proceedings of SAC 2019.

[BCS13] Daniel J. Bernstein, Tung Chou, and Peter Schwabe. Mcbits: Fast constant-time code-based cryptography. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *LNCS*, pages 250–272. Springer, 2013.

[BDK+11] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 1–20, 2011.

[BJLS16] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 273–304. Springer, 2016.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012*, LNCS. Springer, 2012.

[BM18] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography 2018*, volume 10786 of *LNCS*, pages 25–46, Fort Lauderdale, FL, USA, April 2018. Springer.

[BMS11] Paulo S.L.M Barreto, Rafael Misoczki, and Marcos A. Jr. Simplicio. One-time signature scheme from syndrome decoding over generic error-correcting codes. *Journal of Systems and Software*, 84(2):198–204, 2011.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures-how to sign with rsa and rabin. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 399–416. Springer, 1996.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174, Gold Coast, Australia, 2001. Springer.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 272–287, 2002.

[COV07] Pierre-Louis Cayrel, Ayoub Otmani, and Damien Vergnaud. On Kabatianskii-Krouk-Smeets signatures. In *Arithmetic of Finite Fields - WAIFI 2007*, volume 4547 of *LNCS*, pages 237–251, Madrid, Spain, June 21–22 2007.

[DST17a] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. A new signature scheme based on $(U|U+V)$ codes. preprint, June 2017. arXiv:1706.08065v1.

[DST17b] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. The problem with the surf scheme. preprint, November 2017. arXiv:1706.08065.

[DT17] Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding. preprint, January 2017. arXiv:1701.07416.

[DT18] Thomas Debris-Alazard and Jean-Pierre Tillich. Two attacks on rank metric code-based schemes: Ranksign and an identity-based-encryption scheme. In *Advances in Cryptology - ASIACRYPT 2018*, volume 11272 of *LNCS*, pages 62–92, Brisbane, Australia, December 2018. Springer.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

[FGO+11] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. In *Proc. IEEE Inf. Theory Workshop- ITW 2011*, pages 282–286, Paraty, Brasil, October 2011.

[FHK+17] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. First round submission to the NIST post-quantum cryptography call, November 2017.

[Fin10] Matthieu Finiasz. 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*, volume 6544 of *LNCS*, pages 159–170. Springer, 2010.

[FRX+17] Kazuhide Fukushima, Partha Sarathi Roy, Rui Xu, Shinsaku Kiyomoto, Kirill Morozov, and Tsuyoshi Takagi. RaCoSS (random code-based signature scheme). First round submission to the NIST post-quantum cryptography call, November 2017.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In M. Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, 2009.

[GM02] Shafi Goldwasser and Daniele Micciancio. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, March 2002.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[GRSZ14] Philippe Gaborit, Olivier Ruatta, Julien Schrek, and Gilles Zémor. New results for rank-based cryptography. In *Progress in Cryptology - AFRICACRYPT 2014*, volume 8469 of *LNCS*, pages 1–12, 2014.

[GS12] Philippe Gaborit and Julien Schrek. Efficient code-based one-time signature from automorphism groups with syndrome compatibility. In *Proc. IEEE Int. Symposium Inf. Theory - ISIT 2012*, pages 1982–1986, Cambridge, MA, USA, July 2012.

[GSJB14] Danilo Gligoroski, Simona Samardjiska, Håkon Jacobsen, and Sergey Bezzateev. McEliece in the world of Escher. IACR Cryptology ePrint Archive, Report2014/360, 2014. `http://eprint.iacr.org/`.

[HBPL18] Andreas Huelsing, Daniel J. Bernstein, Lorenz Panny, and Tanja Lange. Official NIST comments made for RaCoSS, 2018. Official NIST comments made for RaCoSS.

[HJ10] Nicholas Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*. Sringer, 2010.

[JJ02] Thomas Johansson and Fredrik Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE Trans. Inform. Theory*, 48(10):2669–2678, October 2002.

[KKS97] Gregory Kabatianskii, Evgenii Krouk, and Ben. J. M. Smeets. A digital signature scheme based on random error-correcting codes. In *IMA Int. Conf.*, volume 1355 of *LNCS*, pages 161–167. Springer, 1997.

[KKS05] Gregory Kabatianskii, Evgenii Krouk, and Sergei Semenov. *Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept*. John Wiley & Sons, 2005.

[LKLN17] Wijik Lee, Young-Sik Kim, Yong-Woo Lee, and Jong-Seon No. Post quantum signature scheme based on modified Reed-Muller code pqsigRM. First round submission to the NIST post-quantum cryptography call, November 2017.

[LS12] Gregory Landais and Nicolas Sendrier. Implementing CFS. In *Progress in Cryptology - INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 474–488. Springer, 2012.

[Lyu09a] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, 2009.

[Lyu09b] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 598–616. Springer, 2009.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $O(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 203–228. Springer, 2015.

[MP16] Dustin Moody and Ray A. Perlner. Vulnerabilities of "McEliece in the World of Escher". In *Post-Quantum Cryptography 2016*, LNCS. Springer, 2016.

[OT11] Ayoub Otmani and Jean-Pierre Tillich. An efficient attack on all concrete KKS proposals. In *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 98–116, 2011.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[PT16] Aurélie Phesso and Jean-Pierre Tillich. An efficient attack on a code-based signature scheme. In *Post-Quantum Cryptography 2016*, volume 9606 of *LNCS*, pages 86–103, Fukuoka, Japan, February 2016. Springer.

[S19] Personal communication with Damien Stehlé.

[Sen11] Nicolas Sendrier. Decoding one out of many. In *Post-Quantum Cryptography 2011*, volume 7071 of *LNCS*, pages 51–67, 2011.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In G. D. Cohen and J. Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *LNCS*, pages 106–113. Springer, 1988.

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In D.R. Stinson, editor, *Advances in Cryptology - CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993.

[Wag02] David Wagner. A generalized birthday problem. In Moti Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.

A Some Useful Distributions

The purpose of this section is to prove Propositions 5 and 6 which give the distributions $q_1^{\text{unif}}, q_2^{\text{unif}}, q_1$ and q_2.

A.1 Proof of Proposition 5

Let us first recall the definitions of q_1^{unif} and q_2^{unif}. We have

$$q_1^{\text{unif}}(i) = \mathbb{P}(|\mathbf{e}_V^{\text{unif}}| = i) \quad ; \quad q_2^{\text{unif}}(s,t) = \mathbb{P}(m_1(\mathbf{e}^{\text{unif}}) = s \mid |\mathbf{e}_V| = t)$$

where

- $\mathbf{e}^{\text{unif}}$ is a random vector drawn uniformly at random among the vectors of weight w in $\mathbb{F}_3^n$
- $\mathbf{e}_V^{\text{unif}} \stackrel{\triangle}{=} -\mathbf{c} \odot \mathbf{e}_1 + \mathbf{a} \odot \mathbf{e}_2$ with $\mathbf{e}_1$ and $\mathbf{e}_2$ being vectors in $\mathbb{F}_3^{n/2}$ such that $\mathbf{e}^{\text{unif}} = (\mathbf{e}_1, \mathbf{e}_2)$ and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ are vectors of $\mathbb{F}_3^{n/2}$ verifying the following equations

$$\forall i \in [\![1, n/2]\!], \quad a_i d_i - b_i c_i = 1 \quad ; \quad a_i c_i \neq 0 \tag{41}$$

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

Let us prove now Proposition 5:

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

$$q_1^{\text{unif}}(i) = \frac{\binom{n/2}{i}}{\binom{n}{w} 2^{w/2}} \sum_{\substack{p=0 \\ w+p \equiv 0 \mod 2}}^{i} \binom{i}{p} \binom{n/2 - i}{(w+p)/2 - i} 2^{3p/2} \tag{16}$$

$$q_2^{\text{unif}}(s,t) = \begin{cases} \dfrac{\binom{t}{s}\binom{n/2-t}{\frac{w+s}{2}-t} 2^{\frac{3s}{2}}}{\sum_p \binom{t}{p}\binom{n/2-t}{\frac{w+p}{2}-t} 2^{\frac{3p}{2}}} & \text{if } w + s \equiv 0 \mod 2. \\ 0 & \text{else} \end{cases} \tag{17}$$

Proof. Let us first compute the distribution q_1^{unif}. The following lemma will be useful:

Lemma 6. $|\mathbf{e}_2 - \mathbf{e}_1| \sim |\mathbf{e}_V^{\text{unif}}|$.

Proof (Proof of Lemma 6). Let $\mathbf{e}_1' \stackrel{\triangle}{=} \mathbf{c} \odot \mathbf{e}_1$, $\mathbf{e}_2' \stackrel{\triangle}{=} \mathbf{a} \odot \mathbf{e}_2$, $\mathbf{e}' \stackrel{\triangle}{=} (\mathbf{e}_1', \mathbf{e}_2')$. $\mathbf{e}'$ is clearly a random vector that is uniformly distributed over the words of weight w in $\mathbb{F}_3^n$ because all the entries of $\mathbf{a}$ and $\mathbf{c}$ are non-zero. Since $\mathbf{e}_V^{\text{unif}} = -\mathbf{c} \odot \mathbf{e}_1 + \mathbf{a} \odot \mathbf{e}_2 = \mathbf{e}_2' - \mathbf{e}_1'$ we deduce that $|\mathbf{e}_2 - \mathbf{e}_1|$ and $|\mathbf{e}_V^{\text{unif}}| = |\mathbf{e}_2' - \mathbf{e}_1'|$ have the same distribution. □

From this lemma, to compute the distribution q_1 it is enough to determine for all i in $[\![1, n/2]\!]$, $\mathbb{P}(|\mathbf{e}_2 - \mathbf{e}_1| = i)$ where $(\mathbf{e}_1, \mathbf{e}_2)$ is uniformly distributed over the words of weight w. Let us define the following quantities:

$$p \stackrel{\triangle}{=} |\{1 \leq i \leq n/2 : (\mathbf{e}_1(i), \mathbf{e}_2(i)) \in \{(1,0),(0,1),(-1,0),(0,-1)\}\}| \tag{42}$$

$$r \stackrel{\triangle}{=} |\{1 \leq i \leq n/2 : (\mathbf{e}_1(i), \mathbf{e}_2(i)) \in \{(1,-1),(-1,1)\}\}| \tag{43}$$

$$l \stackrel{\triangle}{=} |\{1 \leq i \leq n/2 : (\mathbf{e}_1(i), \mathbf{e}_2(i)) \in \{(1,1),(-1,-1)\}\}| \tag{44}$$

We have:

$$w = |\mathbf{e}| = 2l + 2r + p \quad ; \quad j = |\mathbf{e}_1 - \mathbf{e}_2| = p + r$$

We have therefore that $p \equiv w \mod 2$, $r = j - p$ and $l = (w+p)/2 - j$. By summing over all possibilities for p, it follows that the number of errors $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$ of weight w such that $|\mathbf{e}_1 - \mathbf{e}_2| = j$ is given by

$$\sum_{\substack{p=0 \\ p \equiv w \bmod 2}}^{j} \binom{n/2}{j}\binom{j}{p} 4^p 2^{j-p} \binom{n/2-j}{\frac{w+p}{2}-j} 2^{\frac{w+p}{2}-j} = \sum_{\substack{p=0 \\ p \equiv w \bmod 2}}^{j} \binom{n/2}{j}\binom{j}{p}\binom{n/2-j}{\frac{w+p}{2}-j} 2^{\frac{w+3p}{2}}$$

which concludes the computation of q_1^{unif}. Let us now compute the distribution q_2^{unif}.

Lemma 7. *Let $n'(s,t)$ be the number of words $\mathbf{e}^{\text{unif}} = (\mathbf{e}_1, \mathbf{e}_2)$ of weight w that verify $|\mathbf{e}_2 - \mathbf{e}_1| = t$ and $m_1(\mathbf{e}^{\text{unif}}) = s$. We have,*

$$n'(s,t) = \begin{cases} \binom{n/2}{t} 2^{w/2} \binom{t}{s} 2^{3s/2} \binom{n/2-t}{\frac{w+s}{2}-t} & \text{if } s \equiv w \bmod 2 \\ 0 \quad \text{else.} \end{cases}$$

Proof. We use the quantities defined in Equations (42),(43) and (44). Note that $m_1(\mathbf{e}^{\text{unif}}) = p$. For words which define $n'(s,t)$ we have $p = s$, $r = t - p = t - s$ and $l = \frac{w+p}{2} - t = \frac{w+s}{2} - t$. Moreover the constraint $p \equiv w \mod 2$ translates into $s \equiv w \mod 2$. □

This concludes the proof by noticing that

$$\mathbb{P}(m_1(\mathbf{e}^{\text{unif}}) = s \mid |\mathbf{e}_V| = t) = \frac{n'(s,t)}{\sum_p n'(p,t)}.$$

A.2 Proof of Proposition 6

Our aim here is to prove Proposition 6. It gives the weight distribution of DecodeV$(\cdot)$ as q_1 and $m_1(\cdot)$-distribution of DecodeU$(\cdot)$ as q_2. Let us recall that algorithms DecodeV$(\cdot)$ and DecodeU$(\cdot)$ are given in Subsection 4.2. We are now ready to prove:

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, $k'_V \triangleq k_V - d$ and $k'_U \triangleq k_U - d$. Let X_V (resp. X_U^t) be a random variable distributed according to $\mathcal{D}_V$ (resp. $\mathcal{D}_U^t$). We have,*

$$q_1(i) = \sum_{t=0}^{i} \frac{\binom{n/2-k'_V}{i-t} 2^{i-t}}{3^{n/2-k'_V}} \mathbb{P}(X_V = t) \tag{18}$$

$$q_2(s,t) = \begin{cases} \displaystyle\sum_{\substack{t+k'_U-n/2 \leq k_{\neq 0} \leq t \\ k_0 \triangleq k'_U - k_{\neq 0}}} \frac{\binom{t-k_{\neq 0}}{s}\binom{n/2-t-k_0}{\frac{w+s}{2}-t-k_0} 2^{\frac{3s}{2}}}{\sum_p \binom{t-k_{\neq 0}}{p}\binom{n/2-t-k_0}{\frac{w+p}{2}-t-k_0} 2^{\frac{3p}{2}}} \mathbb{P}(X_U^t = k_{\neq 0}) & \text{if } w \equiv s \bmod 2. \\ 0 & \text{else} \end{cases} \tag{19}$$

Proof. The computation of q_1 easily follows from the fact that $|\mathbf{e}_V|$ (the output of Prange Algorithm, Line 4 in Algorithm 4) can be written (Proposition 2 in Subsection 3.2) as $S + T$ where S and T are independent random variables such that S denotes the weight of a vector that is uniformly distributed over $\mathbb{F}_3^{n/2-k'_V}$ and T is distributed according to $\mathcal{D}_V$ (in the Prange algorithm used in DecodeV$(\cdot)$ we uniformly picked d symbols in the information set). To compute q_2 let us count the number $n(s,t,k_{\neq 0})$ of different $\mathbf{e}_U$ that can be output by DecodeU$(\cdot)$ for a given value of $\mathbf{e}_V$ (which is supposed to be of weight t) and $\mathcal{J}$ (included in an information set $\mathcal{I}$) that is assumed to intersect the support of $\mathbf{e}_V$ in exactly $k_{\neq 0}$ positions and that are such that $m_1(\mathbf{e}) = s$. We can partition $[\![1, n/2]\!]$ as

$$[\![1, n/2]\!] = \mathcal{J} \cup \mathcal{I}_1 \cup \mathcal{I}_2$$

where $\mathcal{I}_1$ is the set of positions that are not in $\mathcal{J}$ but in the support of $\mathbf{e}_V$, whereas $\mathcal{I}_2$ is the set of positions that are neither in $\mathcal{J}$ nor in the support of $\mathbf{e}_V$. By assumption on $\mathbf{e}_V$ we know that $|\mathcal{I}_1| = t - k_{\neq 0}$. Furthermore $|\mathcal{J}| = k_U - d$ and $\mathcal{I}_2 = n/2 - |\mathcal{J}| - |\mathcal{I}_1| = n/2 - k_U + d - (t - k_{\neq 0}) = n/2 - t - k_0$ where $k_0 \stackrel{\triangle}{=} k_U - d - k_{\neq 0}$. For $i \in \{0, 1, 2\}$ we let

$$\mathcal{J}_i \stackrel{\triangle}{=} \{i \in [\![1, n/2]\!] : |(e_i, e_{i+n/2})| = i\} \quad ; \quad j_i \stackrel{\triangle}{=} |\mathcal{J}_i|.$$

We necessarily have

$$j_1 = s \quad ; \quad n - w = j_1 + 2j_0.$$

We derive from these equalities that

$$j_0 = \frac{n - w - s}{2}$$

Now we also have

$$\mathcal{J}_1 \subseteq \mathcal{I}_1 \quad ; \quad \mathcal{J}_0 \subseteq \mathcal{I}_2.$$

We can choose the $j_1 = s$ positions of $\mathcal{J}_1$ as we wish among the $t - k_{\neq 0}$ positions of $\mathcal{I}_1$. Similarly we may choose the $j_0 = \frac{n-w-s}{2}$ positions of $\mathcal{J}_0$ as we wish among the $n/2 - t - k_0$ positions of $\mathcal{I}_2$. Vector $\mathbf{e}_U$ is necessarily fixed over all positions in $\mathcal{J}$ by choice of the Prange algorithm, it is also necessarily fixed in the positions $\mathcal{I}_1 \setminus \mathcal{J}_1$ and $\mathcal{J}_0$. For positions i in $\mathcal{J}_1 \cup (\mathcal{I}_2 \setminus \mathcal{J}_0)$ there are two possibilities for the value $\mathbf{e}_U(i)$. This implies that

$$\begin{aligned} n(s, t, k_{\neq 0}) &= \binom{t - k_{\neq 0}}{s} \binom{n/2 - t - k_0}{\frac{n-w-s}{2}} 2^s 2^{n/2 - t - k_0 - \frac{n-w-s}{2}} \\ &= \binom{t - k_{\neq 0}}{s} \binom{n/2 - t - k_0}{\frac{n-w-s}{2}} 2^{\frac{3s}{2} + \frac{w}{2} - t - k_0}. \end{aligned}$$

We therefore have

$$\begin{aligned} \mathbb{P}(m_1(\mathbf{e}) = s \mid |\mathbf{e}_V| = t, \mathcal{J} \cap \mathrm{Supp}(\mathbf{e}_V) = k_{\neq 0}) &= \frac{n(s, t, k_{\neq 0})}{\sum_p n(s, t, p)} \\ &= \frac{\binom{t-k_{\neq 0}}{s}\binom{n/2-t-k_0}{\frac{n-w-s}{2}} 2^{\frac{3s}{2} + \frac{w}{2} - t - k_0}}{\sum_p \binom{t-k_{\neq 0}}{p}\binom{n/2-t-k_0}{\frac{n-w-p}{2}} 2^{\frac{3p}{2} + \frac{w}{2} - t - k_0}} \\ &= \frac{\binom{t-k_{\neq 0}}{s}\binom{n/2-t-k_0}{\frac{n-w-s}{2}} 2^{\frac{3s}{2}}}{\sum_p \binom{t-k_{\neq 0}}{p}\binom{n/2-t-k_0}{\frac{n-w-p}{2}} 2^{\frac{3p}{2}}}. \end{aligned}$$

This concludes the proof by summing over all possibilities for $k_{\neq 0}$.

B A refinement of Theorem 1

The following theorem strengthens significantly Theorem 1.

Theorem 3. *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*

$$\mathbb{P}\left(\rho(\mathbf{e}, \mathbf{e}^{\mathrm{unif}}) > \frac{(n/2+1)(n/2 - k_U + d + 1) + 1}{3^d}\right) \leq \frac{2}{\binom{n/2}{k-d}} \left(3^d + 2 \cdot 3^{2d + \gamma n/2}\right) + 3^d \sum_{t,\ell} \frac{2 + 4n3^{n\gamma_0/2}}{\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}(\alpha_{t,\ell} - 1)^2}$$

where the probability is taken over the choice of matrices $\mathbf{H}_V$ *and* $\mathbf{H}_U$ *with,*

$$\gamma \triangleq \min_{x>0}\left((1-R_V+\delta)\log_3\left(\frac{1+3x}{x}\right)+(R_V-\delta)\log_3(1+x)\right)-1+R_V$$

$$\gamma_1(\pi) \triangleq \inf_{x>0} \pi\log_3(1+3x)+(\tau-\pi)\log_3(1+x)-(\tau-\lambda)\log_3(x)$$

$$\gamma_2(\pi) \triangleq \inf_{x>0} (1-R+\delta-\pi)\log_3(1+3x)+(R-\delta+\pi-\tau)\log_3(1+x)-(1-R+\delta-\tau+\lambda)\log_3(x)$$

$$\gamma_0 \triangleq R-1+\sup_{\pi}\left\{\gamma_1(\pi)+\gamma_2(\pi)+(1-R+\delta)h_3\left(\frac{\pi}{1-R+\delta}\right)+(R-\delta)h_3\left(\frac{\tau-\pi}{R-\delta}\right)\right\}.$$

$$\alpha_{t,\ell} \triangleq \frac{2}{\mathbb{P}\left(k_{\neq 0}=\ell \mid |\mathbf{e}_V|=t\right)\mathbb{P}(|\mathbf{e}_V^{\text{unif}}|=t)}$$

where,

$$\delta=\frac{d}{n/2},\quad R_V \triangleq \frac{k_V}{n/2},\quad R_U \triangleq \frac{k_U}{n/2},\quad \tau \triangleq \frac{t}{n/2},\quad \lambda \triangleq \frac{\ell}{n/2}$$

and,

$$h_3(x) \triangleq -(1-x)\log_3(1-x)-x\log_3\left(\frac{x}{2}\right).$$

Remark 6. For the set of parameters of §7.3 (with $d=81$), we have $\mathbb{P}\left(\rho(\mathbf{e},\mathbf{e}^{\text{unif}})>\frac{1}{2^{106}}\right)<2^{-600}$.

The quantities

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

are functions of $\mathbf{H}_V$ and $\mathbf{H}_U$. We are going to show that their probabilities over $\mathbf{H}_V$ and $\mathbf{H}_U$ to be greater than $1/3^d$ is negligible. We will first need the following lemma .

Lemma 8. *Let* d *and* m *be two positive integers with* $d<m$ *and let* $\mathbf{M}$ *be a matrix chosen uniformly at random in* $\mathbb{F}_3^{(m-d)\times m}$*. The probability that* $\mathbf{M}$ *is of rank* $<m-d$ *is upper-bounded by* $\frac{1}{2\cdot 3^d}$*.*

Proof. Let $\mathbf{M}_1,\ldots,\mathbf{M}_{m-d}$ be the rows of $\mathbf{M}$. Let V_i be the vector space spanned by $\mathbf{M}_1,\ldots,\mathbf{M}_i$. If $\mathbf{M}$ is not of full rank then necessarily for at least one $i\in[\![1,m-d]\!]$ we have $\dim V_i=\dim V_{i-1}=i-1$ where $V_{-1}\triangleq\{0\}$.The probability P that $\mathbf{M}$ is not of full rank is therefore upper-bounded by

$$\begin{aligned}
P &\leq \sum_{i=1}^{m-d}\mathbb{P}(\dim V_i=\dim V_{i-1}=i-1)\\
&\leq \sum_{i=1}^{m-d}\mathbb{P}(\dim V_i=i-1|\dim V_{i-1}=i-1)\\
&= \sum_{i=1}^{m-d}\frac{1}{3^{m+1-i}}\\
&\leq \frac{1}{2\cdot 3^d}.
\end{aligned}$$

□

The following lemmas will be useful too.

Lemma 9. *Let* X *and* Y *be two Bernoulli variables that are independent conditioned on an event* E*. Let* $\varepsilon \triangleq \mathbb{P}(\overline{E})$*. Then*

$$\mathbb{E}(XY)-\mathbb{E}(X)\mathbb{E}(Y)\leq 2\varepsilon.$$

Proof. We have

$$\begin{aligned}\mathbb{E}(XY) &= \mathbb{P}(X=1,Y=1|E)\mathbb{P}(E) + \mathbb{P}(X=1,Y=1|\overline{E})\mathbb{P}(\overline{E}) \\ &\leq \mathbb{P}(X=1|E)\mathbb{P}(Y=1|E)(1-\varepsilon)+\epsilon.\end{aligned}$$

On the other hand

$$\mathbb{E}(X)\mathbb{E}(Y) \geq \mathbb{P}(X=1|E)\mathbb{P}(Y=1|E)(1-\varepsilon)^2.$$

Using both bounds yields

$$\begin{aligned}\mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y) &\leq P(X=1|E)\mathbb{P}(Y=1|E)(1-\varepsilon-(1-\varepsilon)^2)+\varepsilon \\ &\leq 2\varepsilon.\end{aligned}$$

□

Lemma 10. *Let $s=\sigma n$, $t=\tau n$ and $w=\omega n$ be three positive integers such that $w \leq \min(s,t)$. We have*

$$\sum_{i=0}^{w} \binom{s}{i}\binom{t}{w-i}3^i \leq 3^{\gamma n}$$

where

$$\gamma \stackrel{\triangle}{=} \inf_{x>0} \sigma \log_3(1+3x) + \tau \log_3(1+x) - \omega \log_3(x).$$

Proof. Let

$$\begin{aligned}a(x) &\stackrel{\triangle}{=} \sum_{j=0}^{s}\binom{s}{j}(3x)^j \\ &= (1+3x)^s \\ b(x) &\stackrel{\triangle}{=} \sum_{j=0}^{t}\binom{t}{j}x^j \\ &= (1+x)^t \\ c(x) &\stackrel{\triangle}{=} a(x)b(x)\end{aligned}$$

We also define the coeffients c_k by $c(x)=\sum_k c_k t^k$. Notice that

$$\begin{aligned}\sum_{i=0}^{w} \binom{s}{i}\binom{t}{w-i}3^i &= c_w \\ &\leq \inf_{x>0} \frac{c(x)}{x^w} \\ &= \inf_{x>0} \frac{a(x)b(x)}{x^w} \\ &= \inf_{x>0} \frac{(1+3x)^s(1+x)^t}{x^w} \\ &= \inf_{x>0} 3^{(\sigma \log_3(1+3x)+\tau\log_3(1+x)-\omega\log_3(x))n} \\ &= 3^{\gamma n}.\end{aligned}$$

□

Lemma 11. *Let* $\mathbf{H}$ *be a matrix chosen uniformly at random in* $\mathbb{F}_3^{(n/2-k)\times n/2}$ *and let* d *be an integer in the range* $[\![1,k]\!]$*. We define* $R \stackrel{\triangle}{=} k/(n/2)$ *and* $\delta = d/(n/2)$*. Let* J^{unif} *be uniformly distributed over the subsets of* $[\![1,n/2]\!]$ *of size* $k_V - d$ *whereas* $J^{\mathbf{H}}$ *is uniformly distributed over the same subsets that are good for* $\mathbf{H}$*. We have*

$$\mathbb{P}\left(\rho(J^{\text{unif}}; J^{\mathbf{H}}) > \frac{1}{3^d}\right) \leq \frac{2}{\binom{n/2}{k-d}}\left(3^d + 2\cdot 3^{2d+\gamma n/2}\right)$$

where

$$\gamma \stackrel{\triangle}{=} \min_{x>0}\left((1-R+\delta)\log_3\left(\frac{1+3x}{x}\right) + (R-\delta)\log_3(1+x)\right) - 1 + R$$

Proof. Recall 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 N be the number of subsets of $[\![1,n/2]\!]$ of size $k-d$ that are bad for $\mathbf{H}$. By using the previous remark, we obtain

$$\rho(J^{\text{unif}}; J^{\mathbf{H}}) = \frac{N}{\binom{n/2}{k-d}}. \tag{45}$$

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

$$N = \sum_{i=1}^{\binom{n/2}{k-d}} X_i. \tag{46}$$

We have by using Bienaymé-Tchebychev's inequality, that for any positive integer t:

$$\begin{aligned}\mathbb{P}(N > \mathbb{E}(N) + t) \leq & \frac{\mathbf{Var}(N)}{t^2} \\ = & \frac{\sum_i \mathbf{Var}(X_i) + \sum_{i\neq j}\mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j)}{t^2} \\ \leq & \frac{\mathbb{E}(N)}{t^2} + \frac{1}{t^2}\left(\sum_{i\neq j}\mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j)\right)\end{aligned} \tag{47}$$

where we use in the last line that $\mathbf{Var}(X_i) \leq \mathbb{E}(X_i^2)$ and $\mathbb{E}(X_i^2) = \mathbb{E}(X_i)$. Let us now upper-bound the second term of the inequality. We first define for any $i \neq j$ the intersection of the complementary of the sets indexed by i and j as $\mathcal{E}_{i,j}$.

By definition of a bad set, if $\mathcal{E}_{i,j} = \emptyset$ then $X_i = 1$ and $X_j = 1$ are independent events and $\mathbb{E}(X_iX_j) = \mathbb{E}(X_i)\mathbb{E}(X_j)$. Otherwise, let $e_{i,j} \stackrel{\triangle}{=} |\mathcal{E}_{i,j}| > 0$. Observe that X_i and X_j are independent conditioned on the event that $\mathbf{H}_{\mathcal{E}_{i,j}}$ is of full rank. We can apply Lemma 9 and obtain for $e_{i,j} \geq 1$

$$\mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) \leq \frac{1}{3^{n/2-k-e_{i,j}}} \tag{48}$$

Let us make the following computations by using (48):

$$\begin{aligned}\sum_{i\neq j}\mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) = & \sum_i \sum_{e=1}^{n/2-k+d}\sum_{j:e_{i,j}=e}\mathbb{E}(X_iX_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) \\ \leq & \sum_i \sum_{e=1}^{n/2-k+d}\sum_{j:e_{i,j}=e}\frac{1}{3^{n/2-k-e_{i,j}}} \\ \leq & \frac{1}{3^{n/2-k}}\binom{n/2}{k-d}\sum_{e=0}^{n/2-k+d} 3^e\binom{n/2-k+d}{e}\binom{k-d}{n/2-k+d-e}\end{aligned} \tag{49}$$

We finish the proof by using Lemma 10 with the sum that appears in the the last term and obtain

$$\begin{aligned}\mathbb{P}(N > \mathbb{E}(N) + t) &\leq \frac{\mathbb{E}(N)}{t^2} + \frac{1}{t^2}\binom{n/2}{k-d}3^{\gamma n/2} \\ &\leq \frac{\binom{n/2}{k-d}}{2\cdot 3^d t^2} + \frac{1}{t^2}\binom{n/2}{k-d}3^{\gamma n/2}\end{aligned}$$

where in the last inequality we used that $\mathbb{E}(N) \leq \frac{\binom{n/2}{k-d}}{2\cdot 3^d}$ which is obtained thanks to Lemma 8. Therefore, by choosing $t = \frac{\binom{n/2}{k-d}}{2\cdot 3^d}$,

$$\mathbb{P}\left(N > \mathbb{E}(N) + \frac{\binom{n/2}{k-d}}{2\cdot 3^d}\right) \leq \frac{1}{\binom{n/2}{k-d}}\left(2\cdot 3^d + 4\cdot 3^{2d+\gamma n/2}\right)$$

But now as $\mathbb{E}(N) \leq \frac{\binom{n/2}{k-d}}{2\cdot 3^d}$,

$$\mathbb{P}\left(N > \frac{\binom{n/2}{k-d}}{3^d}\right) \leq \frac{2}{\binom{n/2}{k-d}}\left(3^d + 2\cdot 3^{2d+\gamma n/2}\right)$$

from which we easily conclude the proof by using Equation (45). □

Lemma 12. *Let* $\mathbf{H}$ *be a matrix chosen uniformly at random in* $\mathbb{F}_3^{(n/2-k)\times n/2}$. *Let* $t \in [\![0, n/2]\!]$, $\ell \in [\![0, n/2]\!]$. *Let* $R \triangleq \frac{2k}{n}$, $\lambda \triangleq \frac{2\ell}{n}$, $t \triangleq \frac{2t}{n}$ *and*

$$\begin{aligned}\gamma_1(\pi) &\triangleq \inf_{x>0} \pi\log_3(1+3x) + (\tau-\pi)\log_3(1+x) - (\tau-\lambda)\log_3(x) \\ \gamma_2(\pi) &\triangleq \inf_{x>0} (1-R+\delta-\pi)\log_3(1+3x) + (R-\delta+\pi-\tau)\log_3(1+x) - (1-R+\delta-\tau+\lambda)\log_3(x) \\ \gamma_0 &\triangleq R - 1 + \sup_{\pi} \gamma_1(\pi) + \gamma_2(\pi) + (1-R+\delta)h_3\left(\frac{\pi}{1-R+\delta}\right) + (R-\delta)h_3\left(\frac{\tau-\pi}{R-\delta}\right). \\ \Delta &\triangleq \frac{\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}}{2\cdot 3^d}(\alpha-1).\end{aligned}$$

where α *is an arbitrary constant satisfying* $\alpha > 1$. *We have,*

$$\mathbb{P}\left(\frac{1}{\binom{n/2}{t}}\sum_{\mathbf{x}\in\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho(I^{\text{unif}}_{\mathbf{x},\ell}; I^{\mathbf{H}}_{\mathbf{x},\ell}) > \frac{\alpha}{2\cdot 3^d}\right) \leq \frac{1}{(\alpha-1)\Delta} + \frac{1}{\Delta^2} n\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}3^{n\gamma_0/2}.$$

where we used the same notation as in Proposition 7.

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

$$\rho(I^{\text{unif}}_{\mathbf{x},\ell}; I^{\mathbf{H}}_{\mathbf{x},\ell}) = \frac{N_{\mathbf{x},\ell}}{\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}}. \tag{50}$$

Let us index these subsets by $1, \ldots, \binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}$ and let $X_{\mathbf{x},\ell}(i)$ be the indicator of the event "the subset of index i is bad". We have

$$N_{\mathbf{x},\ell} = \sum_{i=1}^{\binom{|\mathbf{x}|}{\ell}\binom{n/2-|\mathbf{x}|}{k-d-\ell}} X_{\mathbf{x},\ell}(i). \tag{51}$$

Let

$$N \stackrel{\triangle}{=} \sum_{\mathbf{x}\in\{0,1\}^{n/2}:|\mathbf{x}|=t} N_{\mathbf{x},\ell} \tag{52}$$

We have,

$$\sum_{\mathbf{x},\in\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho(I^{\text{unif}}_{\mathbf{x},\ell}; I^{\mathbf{H}}_{\mathbf{x},\ell}) = N. \tag{53}$$

We have by using Bienaymé-Tchebychev's inequality, that for any positive integer Δ:

$$\begin{aligned}
\mathbb{P}(N > \mathbb{E}(N) + \Delta) &\leq \frac{\mathbf{Var}(N)}{\Delta^2} \\
&= \frac{\sum_{\mathbf{x},i} \mathbf{Var}(X_{\mathbf{x},\ell}(i)) + \sum_{(\mathbf{x},i)\neq(\mathbf{y},j)} \left(\mathbb{E}\left(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},\ell}(j)\right) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},\ell}(j))\right)}{\Delta^2} \\
&\leq \frac{\mathbb{E}(N)}{\Delta^2} + \frac{1}{\Delta^2}\left(\sum_{(\mathbf{x},i)\neq(\mathbf{y},j)} \left(\mathbb{E}\left(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},m}(j)\right) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},m}(j))\right)\right)
\end{aligned} \tag{54}$$

where we use in the last line that $\mathbf{Var}(X_{\mathbf{x},\ell}(i)) \leq \mathbb{E}(X_{\mathbf{x},\ell}(i)^2)$, $\mathbb{E}(X_{\mathbf{x},\ell}(i)^2) = \mathbb{E}(X_{\mathbf{x},\ell}(i))$.

Let us now upper-bound the second term of the inequality. We first define for any $(\mathbf{x}, i)$ and $(\mathbf{y}, j)$ the intersection of the complementary of the sets indexed by i and j for $(\mathbf{x}, i)$ and $(\mathbf{y}, j)$ as $\mathcal{E}(\mathbf{x}, i; \mathbf{y}, j)$. Let $e(\mathbf{x}, i; \mathbf{y}, m) \stackrel{\triangle}{=} |\mathcal{E}(\mathbf{x}, i; \mathbf{y}, j)|$ and we suppose that $e(\mathbf{x}, i; \mathbf{y}, j) > 0$. By using Lemma 9 we obtain:

$$\mathbb{E}(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},m}(j)) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},m}(j)) \leq \frac{1}{3^{|n/2-k-e(\mathbf{x},i;\mathbf{y},j)|}}. \tag{55}$$

When $e(\mathbf{x}, i; \mathbf{y}, j) = 0$, $X_{\mathbf{x},\ell}(i)$ and $X_{\mathbf{y},m}(j)$ are independent and we have in this case $\mathbb{E}(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},m}(j)) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},m}(j)) = 0$. This implies

$$\begin{aligned}
&\sum_{(\mathbf{x},i)\neq(\mathbf{y},j)} \left(\mathbb{E}(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},\ell}(j)) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},\ell}(j))\right) \\
&\quad\leq \sum_{(\mathbf{x},i)} \sum_{e=1}^{n/2-k+d} \sum_{(\mathbf{y},j):e(\mathbf{x},i;\mathbf{y},j)=e} \left(\mathbb{E}(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},m}(j)) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},m}(j))\right) \\
&\quad\leq \sum_{(\mathbf{x},i)} \sum_{e=1}^{n/2-k+d} \sum_{(\mathbf{y},j):e(\mathbf{x},i;\mathbf{y},j)=e} \frac{1}{3^{|n/2-k-e|}} \quad \text{(By using Eq.(55))}
\end{aligned}$$

Our aim now is to compute the following quantity:

$$S(\mathbf{x}, i, \mathbf{y}) \stackrel{\triangle}{=} \sum_{e=1}^{n/2-k+d} \sum_{j:e(\mathbf{x},i;\mathbf{y},j)=e} \frac{1}{3^{|n/2-k-e|}} \tag{56}$$

Let us denote by $\mathcal{E}_i$ (resp. $\mathcal{F}_j$) the complementary of the set indexed by i (resp. j). Let,

$$p \stackrel{\triangle}{=} |\text{Supp}(\mathbf{y}) \cap \mathcal{E}_i| \,.$$

It will be helpful to partition the support $[\![1, n/2]\!]$ as

$$[\![1, n/2]\!] = (\text{Supp}(\mathbf{y}) \cap \mathcal{E}_i) \cup (\overline{\text{Supp}(\mathbf{y})} \cap \mathcal{E}_i) \cup (\text{Supp}(\mathbf{y}) \cap \overline{\mathcal{E}_i}) \cup (\overline{\text{Supp}(\mathbf{y})} \cap \overline{\mathcal{E}_i})$$

Here,

$$|\mathcal{F}_j| = |\mathcal{E}_i| = n/2 - k + d \quad ; \quad |\overline{\mathcal{E}}_i| = k - d$$

By definition we have $|\operatorname{Supp}(\mathbf{y})| = t$. We also have

$$\begin{aligned}\left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{E}_i\right| &= |\mathcal{E}_i| - |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i| \\ &= n/2 - k + d - p \qquad (57)\\ \left|\operatorname{Supp}(\mathbf{y}) \cap \overline{\mathcal{E}_i}\right| &= |\operatorname{Supp}(\mathbf{y})| - |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i| \\ &= t - p \qquad (58)\\ \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \overline{\mathcal{E}_i}\right| &= \left|\overline{\operatorname{Supp}(\mathbf{y})}\right| - \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{E}_i\right| \\ &= n/2 - t - (n/2 - k + d - p) \\ &= k - d + p - t. \qquad (59)\end{aligned}$$

We bring in now

$$f \stackrel{\triangle}{=} |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i \cap \mathcal{F}_j| \qquad (60)$$

$$g \stackrel{\triangle}{=} \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{E}_i \cap \mathcal{F}_j\right|. \qquad (61)$$

Observe that we have

$$e = |\mathcal{E}_i \cap \mathcal{F}_j| = |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i \cap \mathcal{F}_j| + \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{E}_i \cap \mathcal{F}_j\right| = f + g. \qquad (62)$$

and that

$$|\operatorname{Supp}(\mathbf{y}) \cap \mathcal{F}_j| = |\operatorname{Supp}(\mathbf{y})| - \left|\operatorname{Supp}(\mathbf{y}) \cap \overline{\mathcal{F}_j}\right| = t - \ell. \qquad (63)$$

Let us compute the cardinalities of $\mathcal{F}_j$ intersected with the sets of the partition. We already know two of them, let us compute the two remaining ones

$$\begin{aligned}\left|\operatorname{Supp}(\mathbf{y}) \cap \overline{\mathcal{E}}_i \cap \mathcal{F}_j\right| &= |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{F}_j| - |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i \cap \mathcal{F}_j| \\ &= t - \ell - f \qquad (64)\\ \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{F}_j \cap \overline{\mathcal{E}}_i\right| &= \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{F}_j\right| - \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{F}_j \cap \mathcal{E}_i\right| \\ &= |\mathcal{F}_j| - |\operatorname{Supp}(\mathbf{y}) \cap \mathcal{F}_j| - \left|\overline{\operatorname{Supp}(\mathbf{y})} \cap \mathcal{F}_j \cap \mathcal{E}_i\right| \\ &= n/2 - k + d - (t - \ell) - g \\ &= n/2 - k + d - t + \ell - g. \qquad (65)\end{aligned}$$

Therefore, $S(\mathbf{x}, i, \mathbf{y})$ of Equation (56) is given by summing over all possible f and g as:

$$\begin{aligned}S(\mathbf{x}, i, \mathbf{y}) &= \sum_{f,g} \binom{p}{f} \binom{n/2 - k + d - p}{g} \binom{t - p}{t - \ell - f} \binom{k - d + p - t)}{n/2 - k + d - t + \ell - g} \frac{1}{3^{|n/2 - k - f - g|}} \\ &\leq \sum_{f,g} \binom{p}{f} \binom{n/2 - k + d - p}{g} \binom{t - p}{t - \ell - f} \binom{k - d + p - t)}{n/2 - k + d - t + \ell - g} \frac{1}{3^{n/2 - k - f - g}} \\ &= \frac{1}{3^{n/2-k}} \sum_{f} \binom{p}{f} \binom{t - p}{t - \ell - f} 3^f \sum_{g} \binom{n/2 - k + d - p}{g} \binom{k - d + p - t}{n/2 - k + d - t + \ell - g} 3^g \end{aligned} \qquad (66)$$

We now use Lemma 10 to bound (66) as

$$S(\mathbf{x}, i, \mathbf{y}) \leq 3^{(R-1)n/2} 3^{\gamma_1(\pi)n/2} 3^{\gamma_2(\pi)n/2}$$

where $\pi \stackrel{\triangle}{=} \frac{2p}{n}$. Now, the number of binary vectors $\mathbf{y}$ of weight t such that $|\operatorname{Supp}(\mathbf{y}) \cap \mathcal{E}_i| = p$ is given by:

$$\binom{n/2 - k + d}{p} 2^p \binom{k - d}{t - p} 2^{t-p} \leq 3^{\left[(1-R+\delta)h_3\left(\frac{\pi}{1-R+\delta}\right) + (R-\delta)h_3\left(\frac{\tau-\pi}{R-\delta}\right)\right]n/2} \qquad (67)$$

We deduce from this that

$$\sum_{(\mathbf{x},i)\neq(\mathbf{y},j)} \mathbb{E}(X_{\mathbf{x},\ell}(i)X_{\mathbf{y},\ell}(j)) - \mathbb{E}(X_{\mathbf{x},\ell}(i))\mathbb{E}(X_{\mathbf{y},\ell}(j)) \leq n\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}3^{n\gamma_0/2}. \tag{68}$$

Plugging this upper-bound into (54) yields

$$\mathbb{P}(N > \mathbb{E}(N) + \Delta) \leq \frac{\mathbb{E}(N)}{\Delta^2} + \frac{1}{\Delta^2} n\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}3^{n\gamma_0/2}$$

We readily observe that $\mathbb{E}(N) \leq \frac{\Delta}{\alpha-1}$ and that

$$\mathbb{P}\left(\frac{1}{\binom{n/2}{t}} \sum_{\mathbf{x}\in\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho(I^{\text{unif}}_{\mathbf{x},\ell}; I^{\mathbf{H}}_{\mathbf{x},\ell}) > \frac{1}{3^d}\right) \leq \mathbb{P}(N > \mathbb{E}(N) + \Delta) \leq \frac{1}{(\alpha-1)\Delta} + \frac{1}{\Delta^2} n\binom{n/2}{t}\binom{t}{\ell}\binom{n/2-t}{k-d-\ell}3^{n\gamma_0/2}.$$

□

We are now ready to prove Theorem 3.

Proof (Theorem 3). By Proposition 7,

$$\begin{aligned}\rho\left(\mathbf{e};\mathbf{e}^{\text{unif}}\right) &\leq \rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) + \sum_{\mathbf{x}_V\in\mathbb{F}_3^{n/2},\ell} \rho\left(I^{\mathbf{H}_U}_{\mathbf{x}_V,\ell}; I^{\text{unif}}_{\mathbf{x}_V,\ell}\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid \mathbf{e}_V = \mathbf{x}_V\right) \mathbb{P}\left(\mathbf{e}^{\text{unif}}_V = \mathbf{x}_V\right) \\ &= \rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) + \sum_{t,\ell} \frac{1}{\binom{n/2}{t}} \sum_{\mathbf{x}\in\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho\left(I^{\mathbf{H}_U}_{\mathbf{x},\ell}; I^{\text{unif}}_{\mathbf{x},\ell}\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid |\mathbf{e}^{\text{unif}}_V| = t\right) \mathbb{P}\left(|\mathbf{e}^{\text{unif}}_V| = t\right),\end{aligned}$$

where we used the fact that $\rho\left(I^{\mathbf{H}_U}_{\mathbf{x},\ell}; I^{\text{unif}}_{\mathbf{x},\ell}\right)$ is constant on all $\mathbf{x}$ that have the same support to reduce the sum of the possible $\mathbf{x}$ from $\mathbb{F}_3^{n/2}$ to $\{0,1\}^{n/2}$. Therefore,

$$\begin{aligned}\mathbb{P}\left(\rho\left(\mathbf{e};\mathbf{e}^{\text{unif}}\right) > \frac{(n/2+1)(n/2-k_U+d+1)+1}{3^d}\right) &\leq \mathbb{P}\left(\rho\left(J^{\mathbf{H}_V}; J^{\text{unif}}\right) > \frac{1}{3^d}\right) \\ &+ \sum_{t,\ell} \mathbb{P}\left(\frac{1}{\binom{n/2}{t}} \sum_{\mathbf{x}\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho\left(I^{\mathbf{H}_U}_{\mathbf{x},\ell}; I^{\text{unif}}_{\mathbf{x},\ell}\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid |\mathbf{e}_V| = t\right) \mathbb{P}\left(|\mathbf{e}^{\text{unif}}_V| = t\right) > \frac{1}{3^d}\right)\end{aligned}$$

We used the union-bound here and the fact that t ranges over $[\![0, n/2]\!]$ whereas ℓ ranges over $[\![t + k_U - d - n/2, t]\!]$. We observe now that

$$\begin{aligned}&\mathbb{P}\left(\frac{1}{\binom{n/2}{t}} \sum_{\mathbf{x}\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho\left(I^{\mathbf{H}_U}_{\mathbf{x},\ell}; I^{\text{unif}}_{\mathbf{x},\ell}\right) \mathbb{P}\left(k_{\neq 0} = \ell \mid |\mathbf{e}_V| = t\right) \mathbb{P}\left(|\mathbf{e}^{\text{unif}}_V| = t\right) > \frac{1}{3^d}\right) \\ &\leq \mathbb{P}\left(\frac{1}{\binom{n/2}{t}} \sum_{\mathbf{x}\{0,1\}^{n/2}:|\mathbf{x}|=t} \rho\left(I^{\mathbf{H}_U}_{\mathbf{x},\ell}; I^{\text{unif}}_{\mathbf{x},\ell}\right) > \frac{1}{3^d \mathbb{P}\left(k_{\neq 0} = \ell \mid |\mathbf{e}_V| = t\right) \mathbb{P}\left(|\mathbf{e}^{\text{unif}}_V| = t\right)}\right)\end{aligned} \tag{69}$$

Let us define,

$$\alpha \stackrel{\triangle}{=} \frac{2}{\mathbb{P}\left(k_{\neq 0} = \ell \mid |\mathbf{e}_V| = t\right) \mathbb{P}\left(|\mathbf{e}_V| = t\right)} \geq 2 \tag{70}$$

To conclude the proof it enough to apply Lemma 12 with α defined in (70) with each term of (69) as long as $\alpha\frac{1}{2\cdot 3^d} < 1$ otherwise we can directly upper-bound the probability by 0. □

C Proof of Proposition 9

C.1 Proof of the variation of the left-over hash lemma

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

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

Proof. Let $q_{h,f}$ be the probability distribution of the discrete random variable $(h_0, h_0(e))$ where h_0 is drawn uniformly at random in $\mathcal{H}$ and e drawn uniformly at random in E (i.e. $q_{h,f} = \mathbb{P}_{h_0,e}(h_0 = h, h_0(e) = f)$). By definition of the statistical distance we have

$$\begin{aligned}
\mathbb{E}_h\left\{\rho(\mathcal{D}(h),\mathcal{U})\right\} &= \sum_{h\in\mathcal{H}} \frac{1}{|\mathcal{H}|}\rho(\mathcal{D}(h),\mathcal{U}) \\
&= \sum_{h\in\mathcal{H}} \frac{1}{2|\mathcal{H}|} \sum_{f\in F} \left|\mathbb{P}_e(h(e) = f) - \frac{1}{|F|}\right| \\
&= \frac{1}{2} \sum_{(h,f)\in\mathcal{H}\times F} \left|\mathbb{P}_{h_0,e}(h_0 = h, h_0(e) = f) - \frac{1}{|\mathcal{H}|\cdot|F|}\right| \\
&= \frac{1}{2} \sum_{(h,f)\in\mathcal{H}\times F} \left|q_{h,f} - \frac{1}{|\mathcal{H}|\cdot|F|}\right|. \qquad (71)
\end{aligned}$$

Using the Cauchy-Schwarz inequality, we obtain

$$\sum_{(h,f)\in\mathcal{H}\times F} \left|q_{h,f} - \frac{1}{|\mathcal{H}|\cdot|F|}\right| \leq \sqrt{\sum_{(h,f)\in\mathcal{H}\times F} \left(q_{h,f} - \frac{1}{|\mathcal{H}|\cdot|F|}\right)^2} \cdot \sqrt{|\mathcal{H}|\cdot|F|}. \qquad (72)$$

Let us observe now that

$$\begin{aligned}
\sum_{(h,f)\in\mathcal{H}\times F} \left(q_{h,f} - \frac{1}{|\mathcal{H}|\cdot|F|}\right)^2 &= \sum_{h,f} \left(q_{h,f}^2 - 2\frac{q_{h,f}}{|\mathcal{H}|\cdot|F|} + \frac{1}{|\mathcal{H}|^2\cdot|F|^2}\right) \\
&= \sum_{h,f} q_{h,f}^2 - 2\frac{\sum_{h,f} q_{h,f}}{|\mathcal{H}|\cdot|F|} + \frac{1}{|\mathcal{H}|\cdot|F|} \\
&= \sum_{h,f} q_{h,f}^2 - \frac{1}{|\mathcal{H}|\cdot|F|}. \qquad (73)
\end{aligned}$$

Consider for $i \in \{0,1\}$ independent random variables h_i and e_i that are drawn uniformly at random in $\mathcal{H}$ and E respectively. We continue this computation by noticing now that

$$\begin{aligned}
\sum_{h,f} q_{h,f}^2 &= \sum_{h,f} \mathbb{P}_{h_0,e_0}(h_0 = h, h_0(e_0) = f)\mathbb{P}_{h_1,e_1}(h_1 = h, h_1(e_1) = f) \\
&= \mathbb{P}_{h_0,h_1,e_0,e_1}\left(h_0 = h_1, h_0(e_0) = h_1(e_1)\right) \\
&= \frac{\mathbb{P}_{h_0,e_0,e_1}\left(h_0(e_0) = h_0(e_1)\right)}{|\mathcal{H}|} \\
&= \frac{1+\varepsilon}{|\mathcal{H}|\cdot|F|}. \qquad (74)
\end{aligned}$$

By substituting for $\sum_{h,f} q_{h,f}^2$ the expression obtained in (74) into (73) and then back into (72) we finally obtain

$$\sum_{(h,f)\in\mathcal{H}\times F}\left|q_{h,f}-\frac{1}{|\mathcal{H}|\cdot|F|}\right|\leq\sqrt{\frac{1+\varepsilon}{|\mathcal{H}|\cdot|F|}-\frac{1}{|\mathcal{H}|\cdot|F|}}\sqrt{|\mathcal{H}|\cdot|F|}=\sqrt{\frac{\varepsilon}{|\mathcal{H}|\cdot|F|}}\sqrt{|\mathcal{H}|\cdot|F|}=\sqrt{\varepsilon}.$$

This finishes the proof of our lemma.

Lemmas 4 and 3 imply directly Proposition 9 as shown in the following proof.

Proof (Proposition 9). Indeed we let in Lemma 3, $E\stackrel{\triangle}{=}\mathbb{F}_3^n$, $F\stackrel{\triangle}{=}\mathbb{F}_3^{n-k}$ and $\mathcal{H}$ be the set of functions associated to the 4-tuples $(\mathbf{H}_U,\mathbf{H}_V,\mathbf{S},\mathbf{P})$ used to generate a public parity-check matrix $\mathbf{H}_{\rm pk}$. These functions h are given by $h(\mathbf{e})=\mathbf{e}\mathbf{H}_{\rm pk}^{\intercal}$. Lemma 4 gives an upper-bound for the ε term in Lemma 3 and this finishes the proof of Proposition 9.

D Distinguishing a Permuted Normalized Generalized $(U, U+V)$-Code

D.1 Proof of Proposition 11

Our aim here is to prove,

Proposition 11. *Assume that we choose a normalized generalized $(U, U+V)$-code over $\mathbb{F}_3$ with a number n_I of linear combinations of type I by picking the parity-check matrices of U and V uniformly at random among the ternary matrices of size $(n/2-k_U)\times n/2$ and $(n/2-k_V)\times n/2$ respectively. Let $a_{(\mathbf{u},\mathbf{v})}(z)$, $a_{(\mathbf{u},\mathbf{0})}(z)$ and $a_{(\mathbf{0},\mathbf{v})}(z)$ be the expected number of codewords of weight z that are respectively in the normalized generalized $(U, U+V)$-code, of the form $(\mathbf{a}\odot\mathbf{u},\mathbf{c}\odot\mathbf{u})$ where $\mathbf{u}$ belongs to U and of the form $(\mathbf{b}\odot\mathbf{v},\mathbf{d}\odot\mathbf{v})$ where $\mathbf{v}$ belongs to V. These numbers are given for even z in $[\![0,n]\!]$ by*

$$a_{(\mathbf{u},\mathbf{0})}(z)=\frac{\binom{n/2}{z/2}2^{z/2}}{3^{n/2-k_U}}\quad;\quad a_{(\mathbf{0},\mathbf{v})}(z)=\frac{1}{3^{n/2-k_V}}\sum_{\substack{j=0\\ j\ even}}^{z}\binom{n_I}{j}\binom{n/2-n_I}{\frac{z-j}{2}}2^{(z+j)/2}$$

$$a_{(\mathbf{u},\mathbf{v})}(z)=a_{(\mathbf{u},\mathbf{0})}(z)+a_{(\mathbf{0},\mathbf{v})}(z)+\frac{1}{3^{n-k_U-k_V}}\left(\binom{n}{z}2^z-\binom{n/2}{z/2}2^{z/2}-\sum_{\substack{j=0\\ j\ even}}^{z}\binom{n_I}{j}\binom{n/2-n_I}{\frac{z-j}{2}}2^{(z+j)/2}\right)$$

and for odd $z\in[\![0,n]\!]$ by

$$a_{(\mathbf{u},\mathbf{0})}(z)=0\quad;\quad a_{(\mathbf{0},\mathbf{v})}(z)=\frac{1}{3^{n/2-k_V}}\sum_{\substack{j=0\\ j\ odd}}^{z}\binom{n_I}{j}\binom{n/2-n_I}{\frac{z-j}{2}}2^{(z+j)/2}$$

$$a_{(\mathbf{u},\mathbf{v})}(z)=a_{(\mathbf{0},\mathbf{v})}(z)+\frac{1}{3^{n-k_U-k_V}}\left(\binom{n}{z}2^z-\sum_{\substack{j=0\\ j\ odd}}^{z}\binom{n_I}{j}\binom{n/2-n_I}{\frac{z-j}{2}}2^{(z+j)/2}\right)$$

On the other hand, when we choose a linear code of length n over $\mathbb{F}_3$ with a random parity-check matrix of size $(n-k_U-k_V)\times n$ chosen uniformly at random, then the expected number $a(z)$ of codewords of weight $z>0$ is given by

$$a(z)=\frac{\binom{n}{z}2^z}{3^{n-k_U-k_V}}.$$

Proof. Lemma 5 in §5 will be useful for the proof. The last part of Proposition 11 is a direct application of this lemma. We namely have,

Proposition 15. *Let $a(z)$ be the expected number of codewords of weight z in a ternary linear code $\mathcal{C}$ of length n whose parity-check matrix is chosen $\mathbf{H}$ uniformly at random among all binary matrices of size $r \times n$. We have*

$$a(z) = \frac{\binom{n}{z}}{3^r}.$$

We are ready now to prove Proposition 11 concerning the expected weight distribution of a random generalized normalized $(U, U+V)$-code, namely a code $(\mathbf{a} \odot U + \mathbf{b} \odot V, \mathbf{c} \odot U + \mathbf{d} \odot V)$ that we will denote by $\mathcal{C}$.

Weight distributions of $(\mathbf{a} \odot U, \mathbf{c} \odot U) \stackrel{\triangle}{=} \{(\mathbf{a} \odot \mathbf{u}, \mathbf{c} \odot \mathbf{u}) : \mathbf{u} \in U\}$ and $(\mathbf{b} \odot V, \mathbf{d} \odot V) \stackrel{\triangle}{=} \{(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v}) : \mathbf{v} \in V\}$. Let us recall from the definition of normalized generalized codes that $a_i c_i \neq 0$ for all $i \in [\![1, n/2]\!]$ and therefore it follows directly from Proposition 15 since $a_{(\mathbf{u},\mathbf{0})}(z) = 0$ for odd and $a_{(\mathbf{u},\mathbf{0})}(z)$ is equal to the expected number of codewords of weight $z/2$ in a random linear code of length $n/2$ with a parity-check matrix of size $(n/2 - k_U) \times n/2$ when z is even. On the other hand, the weight distribution of $(\mathbf{b} \odot \mathbf{v}, \mathbf{d} \odot \mathbf{v})$ for $\mathbf{v} \in V$ is little more sophisticate. It depends of the number n_I (see Definition 6) when either $b_i = 0$ or $d_i = 0$, the other one is necessarily different from 0. In this way, $a_{(\mathbf{0},\mathbf{v})}(z)$ is equal to the expected number of weight $j + \frac{z-j}{2}$ for all j in $[\![1, n_I]\!]$ in a random linear code of length $n/2$ where j positions correspond to the n_I positions which gives the number of block of type I and $\frac{z-j}{2}$ for the others as there are involved in components which count twice in the weight. Furthermore this code has a parity-check matrix of size $(n/2 - k_V) \times n/2$ which easily gives from Proposition 15 the expected result for $a_{(\mathbf{0},\mathbf{v})}$.

Weight distributions of $\mathcal{C}$. The normalized generalized $(U, U+V)$-code is chosen randomly by picking up a parity-check matrix $\mathbf{H}_U$ of U (resp. $\mathbf{H}_V$ of V) uniformly at random among the set of $(n/2 - k_U) \times n/2$ (resp. $(n/2 - k_V) \times n/2$) ternary matrices. Let $Z \stackrel{\triangle}{=} \sum_{\mathbf{x} \in \mathbb{F}_3^n : |\mathbf{x}| = z} Z_{\mathbf{x}}$ where $Z_{\mathbf{x}}$ is the indicator function of "$\mathbf{x} \in \mathcal{C}$". Therefore,

$$\begin{aligned} a_{(\mathbf{u},\mathbf{v})}(z) &= \mathbb{E}(Z) \\ &= \sum_{\mathbf{x} \in \mathbb{F}_3^n : |\mathbf{x}| = z} \mathbb{P}(\mathbf{x} \in \mathcal{C}) \end{aligned} \tag{75}$$

Therefore, by Proposition 3 we get: $\mathbf{x} \in \mathcal{C} \iff \mathbf{x}_U \mathbf{H}_U^\intercal = \mathbf{0}$ and $\mathbf{x}_V \mathbf{H}_V^\intercal = \mathbf{0}$ which lead to three disjoint cases to (we use in each case Lemma 5):

Case 1: $\mathbf{x}_U = \mathbf{0}$ and $\mathbf{x}_V \neq \mathbf{0}$,

$$\mathbb{P}(\mathbf{x} \in \mathcal{C}) = \mathbb{P}(\mathbf{x}_V \mathbf{H}_V^\intercal = \mathbf{0}) = \frac{1}{3^{n/2 - k_V}}$$

Case 2: $\mathbf{x}_U \neq \mathbf{0}$ and $\mathbf{x}_V = \mathbf{0}$,

$$\mathbb{P}(\mathbf{x} \in \mathcal{C}) = \mathbb{P}(\mathbf{x}_U \mathbf{H}_U^\intercal = \mathbf{0}) = \frac{1}{3^{n/2 - k_U}}$$

Case 3: $\mathbf{x}_U \neq \mathbf{0}$ and $\mathbf{x}_V \neq \mathbf{0}$,

$$\mathbb{P}(\mathbf{x} \in \mathcal{C}) = \mathbb{P}(\mathbf{x}_V \mathbf{H}_V^\intercal = \mathbf{0}, \mathbf{x}_U \mathbf{H}_U^\intercal = \mathbf{0}) = \frac{1}{3^{n/2 - k_U}} \frac{1}{3^{n/2 - k_V}}$$

By substituting $\mathbb{P}(\mathbf{x} \in \mathcal{C})$ in (75) and using definition of number of blocks of type I we conclude the proof. □

D.2 Proof of Propositions 12 and 13

Our aim is to prove the following proposition. It gives the expected number of iteration of Algorithm 6 to output a non zero list with probability $\Omega(1)$.

Proposition 12. *The probability P_{succ} that one iteration of the for loop (Instruction 2) in* ComputeU *adds elements to the list B is lower-bounded by*

$$P_{succ} \geq \sum_{z=0}^{n/2} \frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z}2^{k+\ell-2z}}{\binom{n}{k+\ell}} f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right) \tag{38}$$

where f is the function defined by $f(x) \stackrel{\triangle}{=} \max\left(x(1-x/2), 1-\frac{1}{x}\right)$. Algorithm 6 returns a non zero list with probability $\Omega(1)$ when N is chosen as $N = \Omega\left(\frac{1}{P_{succ}}\right)$.

Proof. It will be helpful to recall [OT11, Lemma 3]

Lemma 13. *Choose a random code $\mathcal{C}_{rand}$ of length n from a parity-check matrix of size $r \times n$ chosen uniformly at random in $\mathbb{F}_3^{r\times n}$. Let X be some subset of $\mathbb{F}_3^n$ of size m. We have*

$$\mathbb{P}(X \cap \mathcal{C}_{rand} \neq \emptyset) \geq f\left(\frac{m}{3^r}\right).$$

We say that two positions i and j are matched (for U') if and only if there exists $\lambda \in \{\pm 1\}$ such that $c_i = \lambda c_j$ for every $\mathbf{c} \in U'$. From the fact that we only consider normalized generalized $(U, U+V)$-codes, there are $n/2$ pairs of matched positions. Z will now be defined by the number of matched pairs that are included in $[\![1, n]\!] \setminus \mathcal{I}$ where $\mathcal{I}$ is the random set of size $n-k-\ell$ which is drawn in Instruction 4 of Algorithm 6. We compute the probability of success by conditioning on the values taken by Z:

$$P_{\text{succ}} = \sum_{z=0}^{n/2} \mathbb{P}(Z=z)\mathbb{P}\left(\exists \mathbf{x} \in U' : |\mathbf{x}_{\bar{\mathcal{I}}}| = p \mid Z = z\right) \tag{76}$$

where $\bar{\mathcal{I}} \stackrel{\triangle}{=} [\![1, n]\!] \setminus \mathcal{I}$. Notice that we can partition $\bar{\mathcal{I}}$ as $\bar{\mathcal{I}} = \mathcal{J}_1 \cup \mathcal{J}_2$ where $\mathcal{J}_2$ consists in the union of the matched pairs in $\bar{\mathcal{I}}$. Note that $|\mathcal{J}_2| = 2z$. We may further partition $\mathcal{J}_2$ as $\mathcal{J}_2 = \mathcal{J}_{21} \cup \mathcal{J}_{22}$ where the elements of a matched pair are divided into the two sets. In other words, neither $\mathcal{J}_{21}$ nor $\mathcal{J}_{22}$ contains a matched pair. We are going to consider the codes

$$U" \stackrel{\triangle}{=} \underset{\mathcal{I}}{\text{Punc}}(U') \quad ; \quad U''' \stackrel{\triangle}{=} \underset{\mathcal{I}\cup\mathcal{J}_{22}}{\text{Punc}}(U')$$

The last code is of length $n-(n-k-\ell+z) = k+\ell-z$ as $|\mathcal{J}_{22}| = z$ and $|\mathcal{I}| = n-k-\ell$. The point of defining the first code is that

$$\mathbb{P}\left(\exists \mathbf{x} \in U' : |\mathbf{x}_{\bar{\mathcal{I}}}| = p \mid Z = z\right)$$

is equal to the probability that $U"$ contains a codeword of weight p. The problem is that we can not apply Lemma 13 to it due to the matched positions it contains (the code is not random). This is precisely the point of defining U'''. In this case, we can consider that it is a random code whose parity-check matrix is chosen uniformly at random among the set of matrices of size $\max(0, k+\ell-z-k_U) \times (k+\ell-z)$. We can therefore apply Lemma 13 to it. We have to be careful about the words of weight p in $U"$ though, since they do not have the same probability of occurring in $U"$ due to the possible presence of matched pairs in the support. This is why we introduce for i in $[\![0, \lfloor p/2 \rfloor]\!]$ the sets X_i defined as follows

$$X_i \stackrel{\triangle}{=} \{\mathbf{x} = (x_i)_{i\in\bar{\mathcal{I}}\setminus\mathcal{J}_{22}} \in \mathbb{F}_3^{k+\ell-z} : |\mathbf{x}_{\mathcal{J}_1}| = p-2i,\ |\mathbf{x}_{\mathcal{J}_{21}}| = i\}$$

A codeword of weight p in U" corresponds to some word in one of the X_i's by puncturing it in $\mathcal{J}_{22}$. We obviously have the lower bound

$$\mathbb{P}\left\{\exists \mathbf{x} \in U' : |\mathbf{x}_{\bar{\mathcal{I}}}| = p \mid Z = z\right\} \geq \max_{i=0}^{\lfloor p/2 \rfloor} \left\{\mathbb{P}(X_i \cap U''' \neq \emptyset)\right\} \tag{77}$$

By using Lemma 13 we have

$$\mathbb{P}(X_i \cap U''' \neq \emptyset) \geq f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right). \tag{78}$$

On the other hand, we may notice that

$$\mathbb{P}(Z = z) = \frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z}2^{k+\ell-2z}}{\binom{n}{k+\ell}}.$$

Thanks to these considerations we conclude the proof. □

Proposition 13. *The probability P_{succ} that one iteration of the for loop (Instruction 2) in* ComputeV *adds elements to the list B is lower-bounded by*

$$P_{succ} \geq \sum_{z=0}^{\min(n-k-\ell,n-n_I)} \sum_{m=0}^{n/2-n_I} \frac{\binom{\frac{n}{2}-n_I}{m}\binom{n_I}{n-k-\ell-z}}{\binom{n}{n-k-\ell}} \max_{i=0}^{\lfloor p/2 \rfloor} f\left(\frac{\binom{n-n_I-z-2m}{p-2i}\binom{m}{i}2^{p-i}}{3^{\max(0,n-n_I-z-m-k_V)}}\right) \sum_{j=0}^{n/2-n_I-m} \binom{n/2-n_I-m}{j} 2^j \binom{n_I}{z-n+2n_I+2m+j}$$

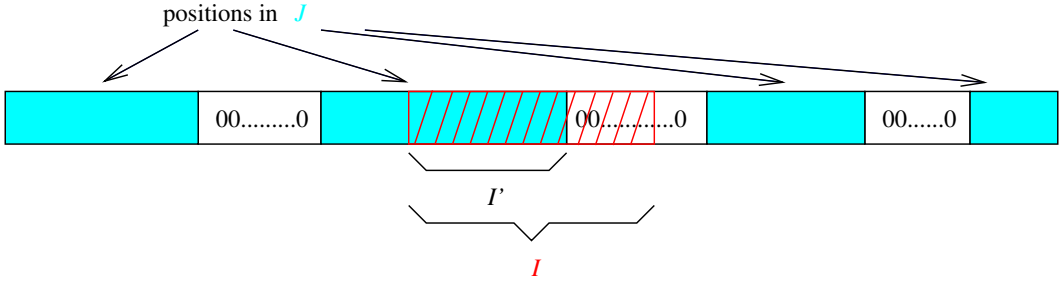
where f is the function defined by $f(x) \triangleq \max\left(x(1-x/2), 1-\frac{1}{x}\right)$. ComputeV *returns a non-zero list with probability $\Omega(1)$ when N is chosen as $N = \Omega\left(\frac{1}{P_{succ}}\right)$.*

Proof. We have $\frac{n}{2} - n_I$ pairs of matched positions i and j (it exists $\lambda \in \{\pm 1\}$ such that $c_i = \lambda c_j$ for every $\mathbf{c} \in V'$). Let us define the following set: $\mathcal{J}$ is the set of positions that are of the images of the permutation $\mathbf{P}$ of the positions $1 \leq i \leq n/2$ such that $b_i \neq 0$ and the images of positions $n/2 + j$ with $0 \leq j \leq n/2$ such that $d_j \neq 0$.

Remark 7. From Definition 6 and Remark 4 in §5 it follows that $|\mathcal{J}| = n - n_I$.

Let us now bring in the following random variables $\mathcal{I}' \triangleq \mathcal{I} \cap \mathcal{J}$, $Z \triangleq |\mathcal{I}'|$ and M be the number of matched pairs which are included in $\mathcal{J} \setminus \mathcal{I}'$. $\mathcal{J} \setminus \mathcal{I}'$ represents the set of positions that are not necessarily equal to 0 in the punctured code $\mathrm{Punc}_{\mathcal{I}}(V')$ (see Figure 6). ComputeV outputs at

Fig. 6. A figure representing $\mathcal{J}$, $\mathcal{I}$ and $\mathcal{I}'$ and the form of a codeword in V'.



least one element of V' if there is an element of weight p in $\mathrm{Punc}_{\mathcal{I}'}(V')$. Therefore the probability of success P_{succ} is given by

$$P_{\mathrm{succ}} = \sum_{z=0}^{\min(n-k-\ell,n-n_I)} \sum_{m=0}^{n/2-n_I} \mathbb{P}\left(\exists \mathbf{x} \in V' : |\mathbf{x}_{\mathcal{J}'}| = p \mid Z = z, M = m\right) \mathbb{P}(Z = z, M = m) \quad (79)$$

where

$$\mathcal{J}' \stackrel{\triangle}{=} \mathcal{J} \setminus \mathcal{I}'.$$

Notice that we can partition $\mathcal{J}'$ as $\mathcal{J}' = \mathcal{J}_1 \cup \mathcal{J}_2$ where $\mathcal{J}_2$ consists in the union of the matched pairs in $\mathcal{J}'$. Note that $|\mathcal{J}_2| = 2m$. We may further partition $\mathcal{J}_2$ as $\mathcal{J}_2 = \mathcal{J}_{21} \cup \mathcal{J}_{22}$ where the elements of a matched pair are divided in two sets. In other words, neither $\mathcal{J}_{21}$ nor $\mathcal{J}_{22}$ contains a matched pair. We are going to consider the following codes

$$V" \stackrel{\triangle}{=} \underset{\mathcal{I} \cup \bar{\mathcal{J}}}{\mathrm{Punc}}(V') \quad ; \quad V''' \stackrel{\triangle}{=} \underset{\mathcal{I} \cup \bar{\mathcal{J}} \cup \mathcal{J}_{22}}{\mathrm{Punc}}(V').$$

$V"$ is of length $n - n_I - z$, whereas the last code is of length $n - n_I - z - m$. The point of defining the first code is that

$$\mathbb{P}\left(\exists \mathbf{x} \in V' : |\mathbf{x}_{\mathcal{J}'}| = p \mid Z = z\right)$$

is equal to the probability that $V"$ contains a codeword of weight p. The problem is that we can not apply Lemma 13 to it due to the matched positions it contains. This is precisely the point of defining V'''. In this case, we can consider that it is a random code whose parity-check matrix is chosen uniformly at random among the set of matrices of size $\max(0, n - n_I - z - m - k_V) \times (n_V - z - m)$. We can therefore apply Lemma 13 to it. We have to be careful about the words of weight p in $V"$ though, since they do not have the same probability of occurring in $V"$ due to the possible presence of matched pairs in the support. This is why we introduce for i in $[\![0, \lfloor p/2 \rfloor]\!]$ the sets X_i defined as follows

$$X_i \stackrel{\triangle}{=} \{\mathbf{x} = (x_i)_{i \in \mathcal{J}' \setminus \mathcal{J}_{22}} \in \mathbb{F}_3^{n-n_I-z-m} : |\mathbf{x}_{\mathcal{J}_1}| = p - 2i,\ |\mathbf{x}_{\mathcal{J}_{21}}| = i\}$$

A codeword of weight p in $V"$ corresponds to some word in one of the X_i's by puncturing it in $\mathcal{J}_{22}$. We obviously have the lower bound

$$\mathbb{P}\left\{\exists \mathbf{x} \in V' : |\mathbf{x}_{\bar{\mathcal{I}}}| = p \mid Z = z, M = m\right\} \geq \max_{i=0}^{\lfloor p/2 \rfloor} \left\{\mathbb{P}(X_i \cap V''' \neq \emptyset)\right\} \quad (80)$$

By using Lemma 13 we have

$$\mathbb{P}(X_i \cap V''' \neq \emptyset) \geq f\left(\frac{\binom{n-n_I-z-2m}{p-2i}\binom{m}{i}2^{p-i}}{3^{\max(0,n-n_I-z-m-k_V)}}\right). \quad (81)$$

On the other hand, we have

$$\mathbb{P}(Z = z, M = m) = \frac{\binom{\frac{n}{2}-n_I}{m}\binom{n_I}{n-k-\ell-z}}{\binom{n}{n-k-\ell}} \sum_{j=0}^{n/2-n_I-m} \binom{n/2-n_I-m}{j} 2^j \binom{n_I}{z-n+2n_I+2m+j}$$

Thanks to these considerations we conclude the proof. □

D.3 Effective Estimate of the Security Exponent for the Recovery of U

Non Asymptotic Setting. Given k, k_U, we want to estimate $\min_{p,\ell} \mathrm{WF}_{p,\ell}$ where

$$\begin{aligned}
\mathrm{WF}_{p,\ell} = \ C_U(p,\ell) \ &= C_{p,\ell}/P_{p,\ell} \\
C_{p,\ell} = C_1(p,k,\ell) &= \max\left(L_{p,\ell}, L_{p,\ell}^2 3^{-\ell}\right) \text{ with } L_{p,\ell} = \sqrt{\tbinom{k+\ell}{p}2^p} \\
P_{p,\ell} = \quad P_{\mathrm{succ}} \quad &= \sum_{z=0}^{n/2} \left(\frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z}2^{k+\ell-2z}}{\binom{n}{k+\ell}} \max_{0 \leq i \leq p/2} f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right)\right)
\end{aligned}$$

with $f(x) = \max(1-1/x, x-x^2/2)$. We may simplify the function $f()$ which is equal up to a small constant factor (smaller than 3) to $\min(1,x)$. We will now assume $f(x) = \min(1,x)$. We write

$$P_{p,\ell} = \sum_{z=0}^{n/2} G_\ell(z) F_{p,\ell}(z),$$

with

$$\begin{aligned}
G_\ell(z) &= \frac{\binom{n/2}{z}\binom{n/2-z}{k+\ell-2z}2^{k+\ell-2z}}{\binom{n}{k+\ell}},\\
F_{p,\ell}(z) &= \max_{0\le i\le p/2} f\left(\frac{\binom{k+\ell-2z}{p-2i}\binom{z}{i}2^{p-i}}{3^{\max(0,k+\ell-z-k_U)}}\right) = \min\left(1, \frac{\max\limits_{0\le i\le p/2}\phi_{p,\ell}(z,i)}{3^{k+\ell-z-k_U}}\right),\\
\phi_{p,\ell}(z,i) &= \tbinom{k+\ell-2z}{p-2i}\tbinom{z}{i}2^{p-i}
\end{aligned}$$

(the max in the denominator of $F_{p,\ell}$ can be removed because $\phi_{p,\ell} \ge 1$).

Asymptotic Setting. We are interested by the asymptotic behavior of the above quantities when n goes to infinity. For the sake of simplicity, we will use the same notations, but all integers parameters k, k_U, p, ℓ, z, i are replaced by their relative values, the letter $x \in \{k, k_U, p, \ell, z, i\}$ now stands for x/n, and instead of an integer it is a real number.

The functions $C_{p,\ell}, L_{p,\ell}, P_{p,\ell}, G_\ell, F_{p,\ell}, \phi_{p,\ell}$ now stand for for their relative asymptotic exponent, that is any X above now stands for $\lim_{n\to\infty} \frac{1}{n}\log_2 X$.

We rewrite

$$\begin{aligned}
\mathrm{WF}_{p,\ell} &= C_{p,\ell} - P_{p,\ell}\\
C_{p,\ell} &= \max\left(L_{p,\ell}, 2L_{p,\ell} - \ell\log_2 3\right) \text{ with } L_{p,\ell} = \frac{k+\ell}{2} h_3\left(\frac{p}{k+\ell}\right)\\
G_\ell(z) &= \frac{1}{2}h_2(2z) + \left(\frac{1}{2}-z\right) h_3\left(\frac{k+\ell-2z}{\frac{1}{2}-z}\right) - h_2(k+\ell)\\
F_{p,\ell}(z) &= \min\left(0, \tilde{F}_{p,\ell}(z)\right)\\
\tilde{F}_{p,\ell}(z) &= \max_{0\le i\le p/2} \phi_{p,\ell}(z,i) - (k+\ell-z-k_U)\log_2 3\\
\phi_{p,\ell}(z,i) &= (k+\ell-2z)h_3\left(\frac{p-2i}{k+\ell-2z}\right) + w h_3\left(\frac{i}{z}\right)
\end{aligned}$$

where $h_q(x) = -x\log_2(x/(q-1)) - (1-x)\log_2(1-x)$ is the q-ary entropy function. The sum in the denominator of $P_{p,\ell}$ will be replaced by a maximum over z

$$P_{p,\ell} = \max_{0\le z\le 1/2} (G_\ell(z) + F_{p,\ell}(z)) \tag{82}$$

To determine which value of z dominates in the above maximum, we need to study the variations of $z \mapsto G_\ell(z)$ and $z \to F_{p,\ell}(z)$. But before that we need to study the variation of $i \mapsto \phi_{p,\ell}(z,i)$ to determine the dominant term in $\max_{0\le i\le p/2}\phi_{p,\ell}(z,i)$.

– The partial derivative of $\phi_{p,\ell}(z,i)$ with respect to i is

$$\frac{\partial\phi_{p,\ell}}{\partial i}(z,i) = \log_2 \frac{(p-2i)^2(z-i)}{2i(k+\ell-2z-p+2i)^2}$$

It follows that the value of i which maximizes $\phi_{p,\ell}(z,i)$ is the solution of a polynomial equation of degree 3.

$$Q(i) = 2i(k+\ell-2z-p+2i)^2 - (p-2i)^2(z-i) \tag{83}$$

An easy analysis shows that Q admits a unique real root in the interval $[0, p/2]$. We denote it $i_0(z)$. We have

$$\tilde{F}_{p,\ell}(z) = \phi_{p,\ell}(z, i_0(z)) - (k + \ell - z - k_U)\log_2 3$$

- The variations of $z \mapsto \tilde{F}_{p,\ell}(z)$ are dominated by the term $z\log_2 3$ and $\tilde{F}_{p,\ell}(z)$ is an increasing function of z. We denote z_1 the (unique) root of $\tilde{F}_{p,\ell}(z)$ in the range $]k + \ell - 1/2, (k + \ell)/2[$. The function $F_{p,\ell}(z)$ is increasing (almost linearly) for $z \in]k + \ell - 1/2, z_1]$ and is null for $z \in [z_1, (k + \ell)/2[$.
- The derivative of $z \to \tilde{F}_{p,\ell}(z)$ is equal to

$$\begin{aligned}\frac{d\tilde{F}_{p,\ell}}{dz}(z) &= \frac{di_0}{dz}(z)\frac{\partial\phi_{p,\ell}}{\partial i}(z, i_0(z)) + \frac{\partial\phi_{p,\ell}}{\partial z}(z, i_0(z)) + \log_2 3 \\ &= \frac{\partial\phi_{p,\ell}}{\partial z}(z, i_0(z)) + \log_2 3 = \log_2 \frac{3z(k + \ell - 2z - p + 2i_0(z))^2}{(z - i_0(z))(k + \ell - 2z)^2}.\end{aligned}$$

- The derivative of $z \to G_\ell(z)$ is equal to

$$\frac{dG_\ell}{dz}(z) = \log_2 \frac{(k + \ell - 2z)^2}{2z(1 - 2k - 2\ell + 2z)}$$

and is null for $z_0 = (k + \ell)^2/2$. The function $z \mapsto G_\ell(z)$ is increasing for $z \in [k + \ell - 1/2, z_0]$, decreasing for $z \in [z_0, (k + \ell)/2]$, and $G_\ell(z_0) = 0$.
- The derivative of $z \to G_\ell(z) + F_{p,\ell}(z)$ is equal to

$$P'_{p,\ell}(z) = \frac{dG_\ell}{dz}(z) + \frac{d\tilde{F}_{p,\ell}}{dz}(z) = \log_2 \frac{3(k + \ell - 2z - p + 2i_0(z))^2}{2(z - i_0(z))(1 - 2k - 2\ell + 2z)}. \tag{84}$$

There exists a unique $z \in]k + \ell - 1/2, (k + \ell)/2[$ which cancels the above derivative we denote it z_2.

For a given pair (p, ℓ),

- Compute z_0, if $F_{p,\ell}(z_0) = 0$ then $P_{p,\ell} = 0$ and $\mathrm{WF}_{p,\ell} = C_{p,\ell}$.
- Compute z_1, z_2, and $z = \min(z_1, z_2)$

$$\mathrm{WF}_{p,\ell} = C_{p,\ell} - G_\ell(z) - F_{p,\ell}(z)$$

Proposition 16. *For any (k, k_U, p, ℓ) let $z_0 = (k+\ell)^2/2$ and let z_1 and z_2 denote respectively the roots of $z \mapsto \tilde{F}_{p,\ell}(z)$ and $z \mapsto P'_{p,\ell}(z)$ for z in $]k + \ell - 1/2, (k + \ell)/2[$. We have*

$$W_{p,\ell} = C_{p,\ell} - G_\ell(z) - F_{p,\ell}(z), \text{ where } z = \max(z_0, \min(z_1, z_2)).$$

Further Simplifications.

- We have a very good approximation of $i_0(z)$ with

$$i_0(z) \approx \frac{p}{2}\frac{pw}{pw + (k + \ell - 2z)^2}.$$

The above assumes that $Q(i)$, given in (83), is close to affine when $i \in [0, p/2]$. It is true enough in practice.
- ***Get rid of parameter p.*** We have

$$C_{p,\ell} = \max\left(L_{p,\ell}, 2L_{p,\ell} - \ell\log_2 3\right)$$

In the max above, *and for the optimal values of the parameters p and ℓ*, the two terms are always equal. This gives us and additional identity

$$h_3\left(\frac{p}{k + \ell}\right) = \frac{2\ell\log_2 3}{k + \ell}$$

which allows us to express the optimal value of p as function of ℓ.

Application to Wave. For Wave $k_U = 0.8379\,n/2$ and $k = 0.660\,n$. In relative value $k_U = 0.41895$ and $k = 0.660$. The minimal value for $W_{p,\ell}$ is reached for $(p, \ell) = (0.0003463, 0.001458)$ and the dominant term in (82) corresponds to $z = 0.24002$. Finally

$$\frac{1}{n}\log_2 \min_{p,\ell} C_U(p,\ell) = 0.015074.$$

Application to Wave Dual Code. The above analysis must also be applied the dual code. In that case, we replace k by $n-k$ and k_U by $n/2 - k_V$ (in the dual U is replaced by $V^\perp$ and V by $U^\perp$). We repeat the analysis with $k_U = 0.24107$ and $k = 0.340$. The minimal value for $W_{p,\ell}$ is reached for $(p, \ell) = (0.00005596, 0.0002650)$ and the dominant term in (82) corresponds to $z = 0.08084$. Finally

$$\frac{1}{n}\log_2 \min_{p,\ell} C_{V^\perp}(p,\ell) = 0.015222.$$

D.4 Security Exponent for the Recovery of V

For the Wave parameters the cost $C_V(p,\ell)$ for recovering V is much larger than the cost $C_U(p,\ell)$ for recovering U. The same holds for $U^\perp$ versus $V^\perp$. Finally, for Wave parameters, the smallest of all is $C_U(p,\ell)$ and it will be used for selecting the parameters.

E Proofs for §6

E.1 Basic Tools

Basic results on the statistical distance. We will need here a few straightforward facts about the statistical distance that we recall here.

Proposition 17. *For i in $[\![1,3]\!]$, let X_i and Y_i be discrete random variables such that the range $\mathcal{A}_i$ of X_i coincides with the range of Y_i. For $a_i \in \mathcal{A}_i$ and $i \in [\![2,3]\!]$ we let $p(.|a_2)$ be the conditional distribution of X_1 given that $X_2 = a_2$, whereas $p(.|a_2, a_3)$ stands for the conditional distribution of X_1 given that $X_2 = a_2$ and $X_3 = a_3$. Similarly $q(.|a_2)$ stands for the conditional distribution of Y_1 given that $Y_2 = a_2$ whereas $q(.|a_2, a_3)$ stands for the conditional distribution of Y_1 given that $Y_2 = a_2$ and $Y_3 = a_3$. We also assume that for all $a_i \in \mathcal{A}_i$ and $i \in [\![2,3]\!]$, we have $\mathbb{P}(X_3 = a_3|X_2 = a_2) = \mathbb{P}(Y_3 = a_3|Y_2 = a_2)$. In such a case for all a_2 in $\mathcal{A}_2$ we have*

$$\rho\left(p(.|a_2), q(.|a_2)\right) \leq \sup_{a_3 \in \mathcal{A}_3} \rho\left(p(.|a_2, a_3), q(.|a_2, a_3)\right).$$

Proof. We will overload the notation p and q by writing for a_3 in $\mathcal{A}_3$

$$p(a_3|a_2) \stackrel{\triangle}{=} \mathbb{P}(X_3 = a_3|X_2 = a_2)$$
$$q(a_3|a_2) \stackrel{\triangle}{=} \mathbb{P}(Y_3 = a_3|Y_2 = a_2)$$

$$\begin{aligned}
\rho(p(.|a_2), q(.|a_2)) &= \frac{1}{2}\sum_{a_1 \in \mathcal{A}_1} |p(a_1|a_2) - q(a_1|a_2)| \\
&= \frac{1}{2}\sum_{a_1 \in \mathcal{A}_1} \left| \sum_{a_3 \in \mathcal{A}_3} p(a_1|a_2, a_3)p(a_3|a_2) - q(a_1|a_2, a_3)q(a_3|a_2) \right| \\
&\leq \frac{1}{2}\sum_{a_3 \in \mathcal{A}_3} p(a_3|a_2) \sum_{a_1 \in \mathcal{A}_1} |p(a_1|a_2, a_3) - q(a_1|a_2, a_3)| \\
&= \sum_{a_3 \in \mathcal{A}_3} \rho\left(p(.|a_2, a_3), q(.|a_2, a_3)\right) p(a_3|a_2) \\
&\leq \sup_{a_3 \in \mathcal{A}_3} \rho\left(p(.|a_2, a_3), q(.|a_2, a_3)\right).
\end{aligned}$$

The following proposition will be helpful to bound the statistical distance between n-tuples of random variables.

Proposition 18. *Let X_i and Y_i with $i \in \{1,2\}$ be discrete random variables where the range $\mathcal{A}_i$ of X_i coincides with the range of Y_i. For a_1 in $\mathcal{A}_1$ we let $p(.|a_1)$ be the conditional distribution of X_2 given that $X_1 = a_1$ whereas $q(.|a_1)$ is the conditional distribution of Y_2 given that $Y_1 = a_1$. We have*

$$\rho\left(X_1X_2, Y_1Y_2\right) \leq \sup_{a_1 \in \mathcal{A}_1} \rho\left(p(.|a_1), q(.|a_1)\right) + \rho\left(X_1, Y_1\right).$$

Proof.

$$\begin{aligned}
\rho\left(X_1X_2, Y_1Y_2\right) &= \frac{1}{2} \sum_{a_1,a_2} \left|\mathbb{P}(X_1 = a_1, X_2 = a_2) - \mathbb{P}(Y_1 = a_1, Y_2 = a_2)\right| \\
&= \frac{1}{2} \sum_{a_1,a_2} \left|\mathbb{P}(X_2 = a_2|X_1 = a_1)\mathbb{P}(X_1 = a_1) - \mathbb{P}(Y_2 = a_2|Y_1 = a_1)\mathbb{P}(Y_1 = a_1)\right|
\end{aligned}$$

To simplify notation we overload the meaning of p and q with the following notation

$$\begin{aligned}
p(a_1) &\triangleq \mathbb{P}(X_1 = a_1) \\
q(a_1) &\triangleq \mathbb{P}(Y_1 = a).
\end{aligned}$$

With this notation at hand we obtain

$$\begin{aligned}
\rho\left(X_1X_2, Y_1Y_2\right) &= \frac{1}{2} \sum_{a_1,a_2} \left|p(a_2|a_1)p(a_1) - q(a_2|a_1)q(a_1)\right| \\
&= \frac{1}{2} \sum_{a_1,a_2} \left|p(a_2|a_1)p(a_1) - q(a_2|a_1)p(a_1) + q(a_2|a_1)p(a_1) - q(a_2|a_1)q(a_1)\right| \\
&\leq \frac{1}{2} \sum_{a_1,a_2} \left|p(a_2|a_1) - q(a_2|a_1)\right| p(a_1) + \frac{1}{2} \sum_{a_1,a_2} \left|p(a_1) - q(a_1)\right| q(a_2|a_1) \\
&\leq \sup_{a_1} \rho\left(p(.|a_1), q(.|a_1)\right) + \frac{1}{2} \sum_{a_1} \left|p(a_1) - q(a_1)\right| \underbrace{\sum_{a_2 \in \mathcal{A}_2} q(a_2|a_1)}_{=1} \\
&= \sup_{a_1 \in \mathcal{A}_1} \rho\left(p(.|a_1), q(.|a_1)\right) + \rho\left(X_1, Y_1\right).
\end{aligned}$$

The Game Associated to Our Code-Based Signature Scheme. In our case, the security of the signature scheme is defined as a game with an adversary that has access to hash and sign oracles. It will be helpful here to be more formal and to define more precisely the games we will consider. They are games between two players, an *adversary* and a *challenger*. In a game G, the challenger executes three kind of procedures:

- an initialization procedure `Initialize` which is called once at the beginning of the game.
- oracle procedures which can be requested at the will of the adversary. In our case, there will be two, `Hash` and `Sign`. The adversary $\mathcal{A}$ which is an algorithm may call `Hash` at most q_{hash} times and `Sign` at most q_{sign} times.
- a final procedure `Finalize` which is executed once $\mathcal{A}$ has terminated. The output of $\mathcal{A}$ is given as input to this procedure.

The output of the game G, which is denoted $G(\mathcal{A})$, is the output of the finalization procedure (which is a bit $b \in \{0,1\}$). The game G with $\mathcal{A}$ is said to be successful if $G(\mathcal{A}) = 1$. The standard approach for obtaining a security proof in a certain model is to construct a sequence of games such that the success of the first game with an adversary $\mathcal{A}$ is exactly the success against the model of

security, the difference of the probability of success between two consecutive games is negligible until the final game where the probability of success is the probability for $\mathcal{A}$ to break one of the problems which is supposed to be hard. In this way, no adversary can break the claim of security with non-negligible success unless it breaks one of the problems that are supposed to be hard.

In the following, $\mathcal{S}_{\text{Wave}}$ will denote the signature scheme defined with the Wave-PSF family.

Definition 10 (challenger procedures in the EUF-CMA Game). *The challenger procedures for the* EUF-CMA *Game corresponding to* $\mathcal{S}_{\text{Wave}}$ *are defined as:*

$\underline{\texttt{proc Initialize}(\lambda)}$	$\underline{\texttt{proc Hash}(\mathbf{m},\mathbf{r})}$	$\underline{\texttt{proc Sign}(\mathbf{m})}$	$\underline{\texttt{proc Finalize}(\mathbf{m},\mathbf{e},\mathbf{r})}$
$(\text{pk},\text{sk}) \leftarrow \texttt{Gen}(1^\lambda)$	$\texttt{return Hash}(\mathbf{m},\mathbf{r})$	$\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$	$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m},\mathbf{r})$
$\mathbf{H}_{\text{pk}} \leftarrow \text{pk}$		$\mathbf{s} \leftarrow \texttt{Hash}(\mathbf{m},\mathbf{r})$	$\texttt{return}$
$(\varphi,\mathbf{H}_U,\mathbf{H}_V,\mathbf{S},\mathbf{P}) \leftarrow \text{sk}$		$\mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}(\mathbf{S}^{-1})^\intercal)$	$\mathbf{eH}_{\text{pk}}^\intercal = \mathbf{s} \wedge \lvert\mathbf{e}\rvert = w$
$\texttt{return } \mathbf{H}_{\text{pk}}$		$\texttt{return } (\mathbf{eP},\mathbf{r})$	

E.2 The Proof

We can now prove the following theorem

Theorem 2. (Security Reduction). *Let* q_{hash} *(resp.* q_{sign}*) be the number of queries to the hash (resp. signing) oracle. We assume that* $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}})$ *where* λ *is the security parameter of the signature scheme. We have in the random oracle model for all time* t*,* $t_c = t + O\left(q_{\text{hash}} \cdot n^2\right)$ *and* ε *given in Proposition 9:*

$$Succ^{\text{EUF-CMA}}_{\mathcal{S}_{\text{Wave}}}(t, q_{\text{hash}}, q_{\text{sign}}) \leq 2Succ^{n,k,q_{\text{hash}},w}_{\text{DOOM}}(t_c) + \rho_c\left(\mathcal{D}_{\text{rand}}, \mathcal{D}_{\text{pub}}\right)(t_c)$$
$$+ q_{\text{sign}}\left(\mathbb{E}_{\mathbf{H}_{\text{pk}}}\left(\rho\left(\mathcal{D}_w^{\mathbf{H}_{\text{pk}}}, \mathcal{U}_w\right)\right) + \frac{\sqrt{\varepsilon}}{2} + \frac{q_{\text{hash}} + q_{\text{sign}}}{q_{\text{sign}}^2 \times 2^\lambda}\right) + \frac{1}{2}(q_{\text{hash}} + q_{\text{sign}})\sqrt{\varepsilon} + \frac{1}{2^\lambda}$$

where $\mathcal{D}_w^{\mathbf{H}_{\text{pk}}}$ *is the distribution sampled as follows:*

– $\mathbf{s} \hookleftarrow \mathbb{F}_3^{n-k}$, $\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$, $\mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}(\mathbf{S}^{-1})^\intercal)$, *output* $(\mathbf{eP},\mathbf{r})$.

with $D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}$ *the Algorithm 3 using Algorithms 4 and 5 and* $\mathcal{U}_w$ *is the uniform distribution over* S_w.

Proof. Let $\mathcal{A}$ be a $(t, q_{\text{sign}}, q_{\text{hash}}, \varepsilon)$-adversary in the EUF-CMA model against $\mathcal{S}_{\text{Wave}}$ and let $(\mathbf{H}_0, \mathbf{s}_1, \cdots, \mathbf{s}_{q_{\text{hash}}})$ be drawn uniformly at random among all instances of DOOM for parameters n, k, q_{hash}, w. We stress here that syndromes $\mathbf{s}_j$ are random and independent vectors of $\mathbb{F}_3^{n-k}$.

Game 0 is the EUF-CMA game for $\mathcal{S}_{\text{Wave}}$.

Game 1 is identical to Game 0 unless the following failure event F occurs: there is a collision in a signature query (*i.e.* two signatures queries for a same message $\mathbf{m}$ lead to the same salt $\mathbf{r}$). By using the difference lemma (see for instance [Sho04, Lemma 1]) we get:

$$\mathbb{P}(S_0) \leq \mathbb{P}(S_1) + \mathbb{P}(F).$$

Here (and also in what follows) $\mathbb{P}(S_i)$ denotes the probability of success for $\mathcal{A}$ of game G_i. The following lemma shows that in our case as $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}})$, the probability of the event F is negligible.

Lemma 14. *For* $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}})$ *we have:* $\mathbb{P}(F) \leq \frac{1}{2^\lambda}$.

Proof. Let us begin by the following lemma.

Lemma 15. *The probability of no collisions after drawing independently t elements among n is bounded by t^2/n.*

Proof. Let us consider for $1 \leq i < j \leq t$ the indicator function $X_{i,j}$ of the event "there is a collision between ith and jth drawn". The probability that we are looking to bound is then given by:

$$\mathbb{P}\left(\bigcup_{1\leq i<j\leq t} X_{i,j} = 1\right)$$

But classically we have,

$$\mathbb{P}\left(\bigcup_{1\leq i<j\leq t} X_{i,j} = 1\right) \leq \sum_{1\leq i<j\leq t} \mathbb{P}\left(X_{i,j} = 1\right).$$

In our case we make t drawing independently in a set of size n, thus:

$$\mathbb{P}(X_{i,j} = 1) = \frac{1}{n}$$

which concludes the proof. □

In our case, the probability of the event F is bounded by the previous inequality with $t = q_{\text{sign}}$ and $n = 2^{\lambda_0}$. In this way, with $\lambda_0 = \lambda + 2\log_2 q_{\text{sign}}$, we get

$$\mathbb{P}\left(F\right) \leq \frac{q_{\text{sign}}^2}{2^{\lambda_0}} = \frac{1}{2^{\lambda_0 - 2\log_2(q_{\text{sign}})}} = \frac{1}{2^{\lambda}}$$

which concludes the proof. □

Game 2 is modified from Game 1 by replacing the procedures `Initialize`, `Hash` and `Sign` as follows (the modifications are in red):

`proc Initialize`	`proc Hash`$(\mathbf{m}, \mathbf{r})$	`proc Sign`$(\mathbf{m})$
$(\text{pk}, \text{sk}) \leftarrow$ `Gen`(1^λ)	if $L_{\mathbf{m}}$ undefined	if $L_{\mathbf{m}}$ undefined
$\mathbf{H}_{\text{pk}} \leftarrow \text{pk}$	$L_{\mathbf{m}} \leftarrow q_{\text{sign}}$ random elements in $\mathbb{F}_2^{\lambda_0}$	$L_{\mathbf{m}} \leftarrow q_{\text{sign}}$ random elements in $\mathbb{F}_2^{\lambda_0}$
$(\varphi, \mathbf{H}_U, \mathbf{H}_V, \mathbf{S}, \mathbf{P}) \leftarrow \text{sk}$	if $\mathbf{r} \in L_{\mathbf{m}}$	$\mathbf{r} \leftarrow L_{\mathbf{m}}$.`next`()
$j \leftarrow 0$	$\mathbf{e}_{\mathbf{m},\mathbf{r}} \hookleftarrow S_w$	$\mathbf{s} \leftarrow$ `Hash`$(\mathbf{m}, \mathbf{r})$
return $\mathbf{H}_{\text{pk}}$	return $\mathbf{e}_{\mathbf{m},\mathbf{r}}\mathbf{H}_{\text{pk}}^\intercal$	$\mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}(\mathbf{S}^{-1})^\intercal)$
	else	return $(\mathbf{eP}, \mathbf{r})$
	$j \leftarrow j + 1$	
	return $\mathbf{s}_j$	

Here the call $L_{\mathbf{m}}$.`next`() returns elements of $L_{\mathbf{m}}$ sequentially. The list is large enough to satisfy all queries. The `Hash` procedure now creates the list $L_{\mathbf{m}}$ if needed, then, if $\mathbf{r} \in L_{\mathbf{m}}$ it returns $\mathbf{e}_{\mathbf{m},\mathbf{r}}\mathbf{H}_{\text{pk}}^\intercal$ with $\mathbf{e}_{\mathbf{m},\mathbf{r}} \hookleftarrow S_w$. Although we do not use it in this game, we remark that $(\mathbf{e}_{\mathbf{m},\mathbf{r}}, \mathbf{r})$ is a valid signature for $\mathbf{m}$. The error value is stored. If $\mathbf{r} \notin L_{\mathbf{m}}$ it outputs one of $\mathbf{s}_j$ of the instance $(\mathbf{H}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{q_{\text{hash}}})$ of the DOOM problem. The `Sign` procedure is unchanged, except for $\mathbf{r}$ which is now taken in $L_{\mathbf{m}}$.

This game can be related to the previous one through the following lemma.

Lemma 16.

$$\mathbb{P}(S_1) \leq \mathbb{P}(S_2) + \frac{q_{\text{hash}}}{2}\sqrt{\varepsilon} \text{ where } \varepsilon \text{ is given in Proposition 9.}$$

Proof. The behavior of Games 1 and 2 only really differ in the calls to `Hash`. In Game 1, a random value X_i uniformly distributed in $\mathbb{F}_3^{n-k}$ is output at the i-th call of `Hash`. In Game 2, if `Hash` is queried with a pair $(\mathbf{m}, \mathbf{r})$ such that $\mathbf{r} \in L_{\mathbf{m}}$, it outputs $Y_i = \mathbf{e}\mathbf{H}_{\text{pk}}^{\intercal}$ where $\mathbf{e}$ has been chosen uniformly at random in S_w. We have

$$\begin{aligned}\mathbb{P}(S_1) - \mathbb{P}(S_2) &= \sum_{\mathbf{H}} \mathbb{P}(\mathbf{H}_{\text{pk}} = \mathbf{H}) \left[\mathbb{P}(S_1 | \mathbf{H}_{\text{pk}} = \mathbf{H}) - \mathbb{P}(S_2 | \mathbf{H}_{\text{pk}} = \mathbf{H})\right] \\ &\leq \mathbb{E}_{\mathbf{H}_{\text{pk}}} \left\{\rho\left(X_1 \cdots X_{q_{\text{hash}}}, Y_1 \cdots Y_{q_{\text{hash}}}\right)\right\}. \end{aligned} \tag{85}$$

Notice that a direct application of Proposition 9 yields

$$\mathbb{E}_{\mathbf{H}_{\text{pk}}} \left\{\rho\left(X_1, Y_1\right)\right\} \leq \frac{\sqrt{\varepsilon}}{2}. \tag{86}$$

We can use now Proposition 18 to bound $\rho\left(X_1 X_2, Y_1, Y_2\right)$. We use the notation p and q to denote by $p(.|x_1)$ the conditional distribution of X_2 given that $X_1 = x_1$ whereas $q(.|x_1)$ denotes the conditional distribution of Y_2 given that $Y_1 = x_1$):

$$\rho\left(X_1 X_2, Y_1 Y_2\right) = \sup_{x_1 \in \mathbb{F}_3^{n-k}} \rho\left(p(.|x_1), q(.|x_1)\right) + \rho\left(X_1, Y_1\right)$$

By using Proposition 9 and (86) we deduce

$$\begin{aligned}\mathbb{E}_{\mathbf{H}_{\text{pk}}} \left\{\rho\left(X_1 X_2, Y_1 Y_2\right)\right\} &\leq \frac{\sqrt{\varepsilon}}{2} + \frac{\sqrt{\varepsilon}}{2} \\ &= \sqrt{\varepsilon}. \end{aligned}$$

An easy induction concludes the proof. □

Game 3 differs from Game 2 by changing in `proc Sign` calls "$\mathbf{e} \leftarrow D_{\varphi, \mathbf{H}_U, \mathbf{H}_V}(\mathbf{s}(\mathbf{S}^{-1})^{\intercal})$" by "$\mathbf{e} \leftarrow \mathbf{e}_{\mathbf{m},\mathbf{r}}$" and "return $(\mathbf{e}\mathbf{P}, \mathbf{r})$" by "return $(\mathbf{e}, \mathbf{r})$". Any signature $(\mathbf{e}, \mathbf{r})$ produced by `proc Sign` is valid. We will prove that:

Lemma 17.

$$\mathbb{P}\left(S_2\right) \leq \mathbb{P}\left(S_3\right) + q_{\text{sign}} \left(\mathbb{E}_{\mathbf{H}_{\text{pk}}} \left(\rho\left(\mathcal{U}_w, \mathcal{D}_w^{\mathbf{H}_{\text{pk}}}\right)\right) + \frac{\sqrt{\varepsilon}}{2} + \frac{q_{\text{hash}} + q_{\text{sign}}}{2^{\lambda_0}}\right).$$

where ε is given in Proposition 9.

Proof. Both games differ in the output of `Sign`. Let $X_1, \cdots, X_{q_{\text{sign}}}$ be the outputs of `Sign` in Game 2, whereas $Y_1, \cdots, Y_{q_{\text{sign}}}$ are the outputs of `Sign` in Game 3. We have

$$\mathbb{P}\left(S_2\right) \leq \mathbb{P}\left(S_3\right) + \rho\left(X_1 \cdots X_{q_{\text{sign}}}, Y_1 \cdots Y_{q_{\text{sign}}}\right).$$

We will bound this statistical distance by using recursively Proposition 18 and bound each term by the following lemma

Lemma 18. *We denote $X_i(\mathbf{m})$ the output of i-th call to* `proc Sign` *in Game 2 if the signing procedure is queried with message $\mathbf{m}$, whereas $Y_i(\mathbf{m})$ denotes the corresponding output for Game 3. Then, for all message $\mathbf{m}$,*

$$\rho\left(X_i(\mathbf{m}), Y_i(\mathbf{m})\right) \leq \mathbb{E}_{\mathbf{H}_{\text{pk}}} \left\{\rho\left(\mathcal{U}_w, \mathcal{D}_w^{\mathbf{H}_{\text{pk}}}\right)\right\} + \varepsilon_{\mathbf{H}_{\text{pk}}} + \frac{q_{\text{hash}} + q_{\text{sign}}}{2^{\lambda_0}}$$

where $\varepsilon_{\mathbf{H}_{\text{pk}}}$ is defined as

$$\varepsilon_{\mathbf{H}_{\text{pk}}} \stackrel{\triangle}{=} \rho\left(\mathbf{e}\mathbf{H}_{\text{pk}}^{\intercal}, \mathbf{s}\right)$$

with $\mathbf{s} \hookleftarrow \mathbb{F}_3^{n-k}$, $\mathbf{e} \hookleftarrow \mathcal{U}_w$ and $\mathcal{U}_w$ be the uniform distribution over words of $\mathbb{F}_3^n$ of weight w.

Proof. Recall that $X_i(\mathbf{m})$ and $Y_i(\mathbf{m})$ are obtained as follows

- $X_i(\mathbf{m}) : \mathbf{r} \leftarrow L_{\mathbf{m}}.\texttt{next}(), \mathbf{s} \leftarrow \mathrm{Hash}(\mathbf{m}, \mathbf{r}), \mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}\left(\mathbf{S}^{-1}\right)^{\intercal}), X_i(\mathbf{m}) \leftarrow (\mathbf{eP}, \mathbf{r})$.
- $\mathcal{D}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}) : \mathbf{r} \leftarrow L_{\mathbf{m}}.\texttt{next}(), \mathbf{e} \leftarrow \mathbf{e}_{\mathbf{m},\mathbf{r}}$, output $(\mathbf{e}, \mathbf{r})$.

Let $(\mathbf{m}_1, \mathbf{r}_1), \ldots, (\mathbf{m}_t, \mathbf{r}_t)$ the queries made to `proc Hash`, including those done by the signing oracle so far and let $R \stackrel{\triangle}{=} \{\mathbf{r}_i : 1 \leq i \leq t\}$. We have $t \leq q \stackrel{\triangle}{=} q_{\mathrm{hash}} + q_{\mathrm{sign}}$. We also define $\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ and $\mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ the distributions $\mathcal{D}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{D}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ conditioned on $\mathbf{r} \notin R$. Since $\mathbf{r} \notin R$ happens with probability at least $1 - \frac{q}{2^{\lambda_0}}$ as $\mathbf{r} \in \{0,1\}^{\lambda_0}$. We have:

$$\rho(\mathcal{D}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{D}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) \leq \rho(\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) + \frac{q}{2^{\lambda_0}}. \tag{87}$$

We consider the intermediate distribution $\mathcal{D}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}$ that can be sampled as follows:

- $\mathcal{D}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}) : \mathbf{r} \leftarrow L_{\mathbf{m}}.\texttt{next}(), \mathbf{s} \hookleftarrow \mathbb{F}_3^{n-k}, \mathbf{e} \leftarrow D_{\varphi,\mathbf{H}_U,\mathbf{H}_V}(\mathbf{s}\left(\mathbf{S}^{-1}\right)^{\intercal})$, output $(\mathbf{eP}, \mathbf{r})$.

and $\mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ the distribution $\mathcal{D}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ conditioned on $\mathbf{r} \notin R$. In $\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$, since $\mathbf{r} \notin R$, the call to $\mathrm{Hash}(\mathbf{m}, \mathbf{r})$ is new and outputs $\mathbf{s}$ which is $\varepsilon_{\mathbf{H}_{\mathrm{pk}}}$ close to uniform. Therefore, we have:

$$\rho(\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) \leq \varepsilon_{\mathbf{H}_{\mathrm{pk}}}. \tag{88}$$

Now, let's compare $\mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ and $\mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$. Distribution $\mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ outputs a random $\mathbf{r} \notin R$ and $\mathbf{e}$ according to distribution $\mathcal{D}_w^{\mathbf{H}_{\mathrm{pk}}}$. Distribution $\mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})$ outputs a random $\mathbf{r} \notin R$ and $\mathbf{e}$ according to distribution $\mathcal{U}_w$ hence:

$$\rho(\mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) \leq \rho(\mathcal{D}_w^{\mathbf{H}_{\mathrm{pk}}}, \mathcal{U}_w). \tag{89}$$

Putting Equations (87),(88) and (89) together, we get:

$$\begin{aligned}
\rho(\mathcal{D}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{D}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) &\leq \rho(\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) + \frac{q}{2^{\lambda_0}} \\
&\leq \rho(\mathcal{E}_2^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) + \rho(\mathcal{E}_{2.5}^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m}), \mathcal{E}_3^{\mathbf{H}_{\mathrm{pk}}}(\mathbf{m})) + \frac{q}{2^{\lambda_0}} \\
&\leq \varepsilon_{\mathbf{H}_{\mathrm{pk}}} + \rho(\mathcal{D}_w^{\mathbf{H}_{\mathrm{pk}}}, \mathcal{U}_w) + \frac{q}{2^{\lambda_0}}.
\end{aligned}$$

which concludes the proof of the lemma. □
□

Game 4 is the game where we replace the public matrix $\mathbf{H}_{\mathrm{pk}}$ by $\mathbf{H}_0$. In this way we will force the adversary to build a solution of the DOOM problem. Here if a difference is detected between games it gives a distinguisher between distributions $\mathcal{D}_{\mathrm{rand}}$ and $\mathcal{D}_{\mathrm{pub}}$:

$$\mathbb{P}(S_3) \leq \mathbb{P}(S_4) + \rho_c(\mathcal{D}_{\mathrm{pub}}, \mathcal{D}_{\mathrm{rand}})(t_c).$$

We show in appendix how to emulate the lists $L_{\mathbf{m}}$ in such a way that list operations cost, including its construction, is at most linear in the security parameter λ. Since $\lambda \leq n$, it follows that the cost to a call to `proc Hash` cannot exceed $O(n^2)$ and the running time of the challenger is $t_c = t + O\left(q_{\mathrm{hash}} \cdot n^2\right)$.

Game 5 differs in the finalize procedure.

`proc Finalize`$(\mathbf{m}, \mathbf{e}, \mathbf{r})$
$\mathbf{s} \leftarrow$ `Hash`$(\mathbf{m}, \mathbf{r})$
$b \leftarrow \mathbf{eH}_{\mathrm{pk}}^{\intercal} = \mathbf{s} \wedge \lvert\mathbf{e}\rvert = w$
`return` $b \wedge \mathbf{r} \notin L_{\mathbf{m}}$

We assume the forger outputs a valid signature $(\mathbf{e}, \mathbf{r})$ for the message $\mathbf{m}$. The probability of success of Game 5 is the probability of the event "$S_4 \wedge (\mathbf{r} \notin L_{\mathbf{m}})$".

If the forgery is valid, the message $\mathbf{m}$ has never been queried by `Sign`, and the adversary never had access to any element of the list $L_{\mathbf{m}}$. This way, the two events are independent and we get:

$$\mathbb{P}(S_5) = (1 - 2^{-\lambda_0})^{q_{\text{sign}}} \mathbb{P}(S_4).$$

As we assumed $\lambda_0 = \lambda + 2\log_2(q_{\text{sign}}) \geq \log_2(q_{\text{sign}}^2)$, we have:

$$\left(1 - 2^{-\lambda_0}\right)^{q_{\text{sign}}} \geq \left(1 - \frac{1}{q_{\text{sign}}^2}\right)^{q_{\text{sign}}} \geq \frac{1}{2}.$$

Therefore

$$\mathbb{P}(S_5) \geq \frac{1}{2}\mathbb{P}(S_4). \tag{90}$$

The probability $\mathbb{P}(S_5)$ is then exactly the probability for $\mathcal{A}$ to output $\mathbf{e}_j \in S_w$ such that $\mathbf{e}_j \mathbf{H}_0{}^{\intercal} = \mathbf{s}_j$ for some j which gives

$$\mathbb{P}(S_5) \leq \ Succ^{n,k,q_{\text{hash}},w}_{\text{DOOM}}(t_c). \tag{91}$$

This concludes the proof of Theorem 2 by combining this together with all the bounds obtained for each of the previous games. □

E.3 List Emulation

In the security proof, we need to build lists of indices (salts) in $\mathbb{F}_3^{\lambda_0}$. Those lists have size q_{sign}, the maximum number of signature queries allowed to the adversary, a number which is possibly very large. For each message $\mathbf{m}$ which is either hashed or signed in the game we need to be able to

- create a list $L_{\mathbf{m}}$ of q_{sign} random elements of $\mathbb{F}_3^{\lambda_0}$, when calling the constructor `new list()`;
- pick an element in $L_{\mathbf{m}}$, using the method $L_{\mathbf{m}}$`.next()`, this element can be picked only once;
- decide whether or not a given salt $\mathbf{r}$ is in $L_{\mathbf{m}}$, when calling $L_{\mathbf{m}}$`.contains(`$\mathbf{r}$`)`.

The straightforward manner to achieve this is to draw q_{sign} random numbers when the list is constructed, this has to be done once for each different message $\mathbf{m}$ used in the game. This may result in a quadratic cost $q_{\text{hash}} q_{\text{sign}}$ just to build the lists. Once the lists are constructed, and assuming they are stored in a proper data structure (a heap for instance) picking an element or testing membership has a cost at most $O(\log q_{\text{sign}})$, that is at most linear in the security parameter λ.

class list	method list.contains($\mathbf{r}$)
elt, index	return $\mathbf{r} \in \{\texttt{elt}[i], 1 \leq i \leq q_{\text{sign}}\}$
list()	
`index` $\leftarrow 0$	method list.next()
for $i = 1, \ldots, q_{\text{sign}}$	`index` $\leftarrow$ `index` $+ 1$
`elt`$[i] \leftarrow$ `randint`(2^{λ_0})	return `elt[index]`

Fig. 7. Standard implementation of the list operations.

Note that in our game we condition on the event that *all elements of* $L_{\mathbf{m}}$ *are different*. This implies that now $L_{\mathbf{m}}$ is obtained by choosing among the subsets of size q_{sign} of $\mathbb{F}_3^{\lambda_0}$ uniformly at random. We wish to emulate the list operations and never construct them explicitly such that the probabilistic model for $L_{\mathbf{m}}$`.next()` and $L_{\mathbf{m}}$`.contains(`$\mathbf{r}$`)` stays the same as above (but again conditioned on the event that all elements of $L_{\mathbf{m}}$ are different). For this purpose, we want to ensure that at any time we call either $L_{\mathbf{m}}$`.contains(`$\mathbf{r}$`)` or $L_{\mathbf{m}}$`.next()` we have

$$\mathbb{P}(L_{\mathbf{m}}\texttt{.contains}(\mathbf{r}) = \texttt{true}) = \mathbb{P}(\mathbf{r} \in L_{\mathbf{m}} | \mathcal{Q}) \tag{92}$$

$$\mathbb{P}(\mathbf{r} = L_{\mathbf{m}}\texttt{.next}()) = p(\mathbf{r} | \mathcal{Q}) \tag{93}$$

for every $\mathbf{r} \in \mathbb{F}_3^{\lambda_0}$. Here $\mathcal{Q}$ represents the queries to $\mathbf{r}$ made so far and whether or not these $\mathbf{r}$'s belong to $L_{\mathbf{m}}$. Queries to $\mathbf{r}$ can be made through two different calls. The first one is a call of the form $\texttt{Sign}(\mathbf{m})$ when it chooses $\mathbf{r}$ during the random assignment $\mathbf{r} \hookleftarrow \{0,1\}^{\lambda_0}$. This results in a call to $\texttt{Hash}(\mathbf{m}, \mathbf{r})$ which queries itself whether $\mathbf{r}$ belongs to $L_{\mathbf{m}}$ or not through the call $L_{\mathbf{m}}.\texttt{contains}(\mathbf{r})$. The answer is necessarily positive in this case. The second way to query $\mathbf{r}$ is by calling $\texttt{Hash}(\mathbf{m}, \mathbf{r})$ directly. In this case, both answers `true` and `false` are possible. $p(\mathbf{r}|\mathcal{Q})$ represents the probability distribution of $L_{\mathbf{m}}.\texttt{next}()$ that we have in the above implementation of the list operations given the previous queries $\mathcal{Q}$.

A convenient way to represent $\mathcal{Q}$ is through three lists S, H_{true} and H_{false}. S is the list of $\mathbf{r}$'s that have been queried through a call $\texttt{Sign}(\mathbf{m})$. They belong necessarily to $L_{\mathbf{m}}$. H_{true} is the set of $\mathbf{r}$'s that have not been queried so far through a call to $\texttt{Sign}(\mathbf{m})$ but have been queried through a direct call $\texttt{Hash}(\mathbf{m}, \mathbf{r})$ and for which $L_{\mathbf{m}}.\texttt{contains}(\mathbf{r})$ returned `true`. H_{false} is the list of $\mathbf{r}$'s that have been queried by a call of the form $\texttt{Hash}(\mathbf{m}, \mathbf{r})$ and $L_{\mathbf{m}}.\texttt{contains}(\mathbf{r})$ returned `false`.

We clearly have

$$\mathbb{P}(\mathbf{r} \in L_{\mathbf{m}}|\mathcal{Q}) = 0 \text{ if } \mathbf{r} \in H_{\text{false}} \tag{94}$$

$$\mathbb{P}(\mathbf{r} \in L_{\mathbf{m}}|\mathcal{Q}) = 1 \text{ if } \mathbf{r} \in S \cup H_{\text{true}} \tag{95}$$

$$\mathbb{P}(\mathbf{r} \in L_{\mathbf{m}}|\mathcal{Q}) = \frac{q_{\text{sign}} - |H_{\text{true}}| - |S|}{2^{\lambda_0} - |H_{\text{true}}| - |S| - |H_{\text{false}}|} \text{ else.} \tag{96}$$

To compute the probability distribution $p(\mathbf{r}|\mathcal{Q})$ it is helpful to notice that

$$\mathbb{P}(L_{\mathbf{m}}.\texttt{next}() \text{ outputs an element of } H_{\text{true}}) = \frac{|H_{\text{true}}|}{q_{\text{sign}} - |S|}. \tag{97}$$

This can be used to derive $p(\mathbf{r}|\mathcal{Q})$ as follows

$$p(\mathbf{r}|\mathcal{Q}) = 0 \text{ if } \mathbf{r} \in H_{\text{false}} \cup S \tag{98}$$

$$p(\mathbf{r}|\mathcal{Q}) = \frac{1}{q_{\text{sign}} - S} \text{ if } \mathbf{r} \in H_{\text{true}} \tag{99}$$

$$p(\mathbf{r}|\mathcal{Q}) = \frac{q_{\text{sign}} - |S| - |H_{\text{true}}|}{(q_{\text{sign}} - S)(2^{\lambda_0} - |H_{\text{true}}| - |S| - |H_{\text{false}}|)} \text{ else.} \tag{100}$$

(98) is obvious. (99) follows from that all elements of H_{true} have the same probability to be chosen as return value for $L_{\mathbf{m}}.\texttt{next}()$ and (97). (100) follows by a similar reasoning by arguing (i) that all the elements of $\mathbb{F}_3^{\lambda_0} \setminus (S \cup H_{\text{true}} \cup H_{\text{false}})$ have the same probability to be chosen as return value for $L_{\mathbf{m}}.\texttt{next}()$, (ii) the probability that $L_{\mathbf{m}}.\texttt{next}()$ outputs an element of $\mathbb{F}_3^{\lambda_0} \setminus (S \cup H_{\text{true}} \cup H_{\text{false}})$ is the probability that it does not output an element of H_{true} which is $1 - \frac{|H_{\text{true}}|}{q_{\text{sign}} - |S|} = \frac{q_{\text{sign}} - |S| - |H_{\text{true}}|}{q_{\text{sign}} - |S|}$.

Figure 8 explains how we perform the emulation of the list operations so that they perform similarly to genuine list operations as specified above. The idea is to create and to operate explicitly on the lists S, H_{true} and H_{false} described earlier. We have chosen there

$$\beta = \frac{q_{\text{sign}} - |H_{\text{true}}| - |S|}{2^{\lambda_0} - |H_{\text{true}}| - |S| - |H_{\text{false}}|} \text{ and } \gamma = \frac{|H_{\text{true}}|}{q_{\text{sign}} - |S|}.$$

we also assume that when we call `randomPop()` on a list it outputs an element of the list uniformly at random and removes this element from it. The method `push` adds an element in a list. The procedure `rand`() picks a real number between 0 and 1 uniformly at random.

The correctness of this emulation follows directly from the calculations given above. For instance the correctness of the call $L_{\mathbf{m}}.\texttt{next}()$ follows from the fact that with probability $\frac{|H_{\text{true}}|}{q_{\text{sign}} - |S|} = \gamma$ it outputs an element of H_{true} chosen uniformly at random (see (97)). In such a case the corresponding element has to be moved from H_{true} to S (since it has been queried now through a call to $\texttt{Sign}(\mathbf{m})$). The correctness of $L_{\mathbf{m}}.\texttt{contains}(\mathbf{r})$ is a direct consequence of the formulas for $\mathbb{P}(\mathbf{r} \in L_{\mathbf{m}}|\mathcal{Q})$ given in (94), (95) and (96). All `push`, `pop`, membership testing above can be implemented in time proportional to λ_0.

class list	method list.contains($\mathbf{r}$)	method list.next()
H_{true}, H_{false}, S	if $\mathbf{r} \notin H_{\text{true}} \cup H_{\text{false}} \cup S$	if $\texttt{rand}() \leq \gamma$
list()	if $\text{rand}() \leq \beta$	$\mathbf{r} \leftarrow H_{\text{true}}.\texttt{randomPop}()$
$H_{\text{true}} \leftarrow \emptyset$	$H_{\text{true}}.\texttt{push}(\mathbf{r})$	else
$H_{\text{false}} \leftarrow \emptyset$	else	$\mathbf{r} \hookleftarrow \mathbb{F}_3^{\lambda_0} \setminus (H_{\text{true}} \cup S \cup H_{\text{false}})$
$S \leftarrow \emptyset$	$H_{\text{false}}.\texttt{push}(\mathbf{r})$	$S.\texttt{push}(\mathbf{r})$
	return $\mathbf{r} \in H_{\text{true}} \cup S$	return $\mathbf{r}$

Fig. 8. Emulation of the list operations.