Public Key Compression for Constrained Linear Signature Schemes

Ward Beullens, Bart Preneel, and Alan Szepieniec(✉)

imec-COSIC KU Leuven, Leuven, Belgium
{ward.beullens,bart.preneel,alan.szepieniec}@esat.kuleuven.be

Abstract. We formalize the notion of a *constrained linear trapdoor* as an abstract strategy for the generation of signature schemes, concrete instantiations of which can be found in MQ-based, code-based, and lattice-based cryptography. Moreover, we revisit and expand on a transformation by Szepieniec *et al.* [39] to shrink the public key at the cost of a larger signature while reducing their combined size. This transformation can be used in a way that is provably secure in the random oracle model, and in a more aggressive variant whose security remained unproven. In this paper we show that this transformation applies to any constrained linear trapdoor signature scheme, and prove the security of the first mode in the quantum random oracle model. Moreover, we identify a property of constrained linear trapdoors that is sufficient (and necessary) for the more aggressive variant to be secure in the quantum random oracle model. We apply the transformation to an MQ-based scheme, a code-based scheme and a lattice-based scheme targeting 128-bits of post quantum security, and we show that in some cases the combined size of a signature and a public key can be reduced by more than a factor 300.

Keywords: Digital signatures · Post-quantum
Quantum random oracle model · Key size reduction

1 Introduction

Trapdoor functions are an important tool in public key cryptography due to the computational asymmetry they bring about. On the one hand, the function is a proper cryptographic one-way function to anyone who is ignorant of the secret trapdoor information; but on the other hand, anyone who does know this trapdoor information can use it to find inverse images quickly.

The case of *surjective* trapdoor functions is especially interesting for generating *digital signature schemes*. A cryptographic hash function maps a message of any size to a random point in the trapdoor function's output space. An inverse of this point under the trapdoor function, or *signature*, testifies to the involvement of the trapdoor information, or *secret key*, in its generation. This testimony ensures the target property of *non-repudiation of origin*: the secret key holder cannot deny generating the signature at a later date.

Since their inception in the seminal paper by Diffie and Hellman [10], various digital signature schemes have been deployed whose security is based on the hardness of integer factorization [35] and the discrete logarithm problem [30, 36]. However, the advent of quantum computers threatens the security of these signature schemes because both hard problems are solved efficiently by Shor's quantum algorithm [37]. This ultimatum drives the need to design, develop and deploy so-called *post-quantum* cryptosystems, *i.e.*, cryptography that can be run on classical hardware but promises to resist attacks by quantum computers.

Even though the RSA trapdoor is broken by quantum computers, the hash-and-sign construction that RSA signatures are based on seems to survive the transition to post-quantum cryptography. To achieve post-quantum secure signature schemes it suffices to exchange the underlying trapdoor for one that has the desired security against quantum adversaries. There is no shortage of trapdoor-based signature schemes based on the MQ problem [11,21,34], coding theory [8,9], or lattices [3,15,27].

Unfortunately, the public keys in these schemes are prohibitively large, measurable in hundreds of kilobytes if not megabytes. In contrast, post-quantum signature schemes derived from zero-knowledge proofs require only a one-way function whose selection can be random or might as well be determined by a short seed and an implicit pseudorandom generator. Signature schemes based on zero-knowledge proofs tend to exchange tiny public keys for prohibitively large signatures [7,18,23,38], and moreover require complicated and expansive non-interactivity transforms to retain security against quantum attackers [40]. Although provable security in the case of hash-based signature schemes is much more straightforward, this family of constructions follows the same pattern: tiny public keys but huge signatures [4,5].

Szepieniec, Beullens and Preneel offer an alternative to the dilemma between large public keys or large signatures [39], motivated by the desire to minimize the combined size of public key and signature. This minimization is particularly important in the context of public key infrastructure (PKI) where a chain of signatures and public keys is transmitted in order to authenticate a message with respect to a pre-shared root public key. The construction of Szepieniec *et al.* applies specifically to MQ trapdoors and relies on the observation that verifying a couple of random linear combinations of the public key's polynomial equations can be as good as verifying all of them. The coefficients of this linear combination are determined as a function of the produced signature, and the combination itself is transmitted along with this signature in addition to information authenticating its link to the public key. This transformation reduces the size of public key plus that of the signature by roughly a factor three whilst provably retaining security in the random oracle model; and by a much larger factor at the expense of a heuristic security argument.

This article expands on the paper of Szepieniec *et al.* in several ways. We observe that this transformation also applies to other post-quantum trapdoor signature schemes, most notably code-based and lattice-based trapdoors. From a general perspective, these three hard problems are variations on a common theme, which we call *constrained linear signature schemes*. This commonality allows a generic presentation of the transformation. The security proofs of Szepieniec *et al.* only work in the classical random oracle model. However, security proofs that purport to defend against quantum adversaries should additionally hold in the *quantum random oracle model*, which our proof does. Moreover, we identify a necessary and sufficient security property, called (σ, r)-hash-and-sign-security ((σ, r)-HSS), that a constrained linear signature scheme must have in order for the more aggressive parameter choices of Szepieniec *et al.* to be provably secure. This leads to an improved understanding of the security of instantiations of this construction, which includes the DualModeMS submission of Faugère *et al.* [12] to the NIST PQC standardization project [29]. To showcase the key size improvements that can be achieved with the transformation, we apply the transformation to a lattice-based, code-based and multivariate constrained linear signature scheme with parameters targeting 128 bits of security against quantum computers.

2 Preliminaries

Random Oracle Model. We use a hash function in our construction. For the purpose of proving security we model it by a *random oracle*, which is a random function $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ with a fixed output length, typically equal to the security parameter. If necessary, the random oracle's output space can be lifted to any finite set X. We use subscripts to differentiate the random oracles associated with different output spaces. A security proof relying on the modelling of hash function as random oracles is said to hold in the *random oracle model*. When quantum adversaries are considered, the security proofs should allow for superposition queries to the random oracle [6]; a security proof with this property is said to hold in the *quantum random oracle model*.

Trapdoor Functions. A trapdoor function is a function that can be efficiently computed in one direction, but for which it is hard to compute preimages *unless* by someone who knows a secret piece of information called the *trapdoor*. We associate three algorithms to a trapdoor function family:

- $\mathsf{GenTrapdoor}$ takes a security parameter as input and outputs a trapdoor function f and a trapdoor t.
- $\mathsf{Evaluate}$ takes a description of the trapdoor function f and an argument x as input, and returns the evaluation of f at x. In the rest of the paper, we simply write this as $f(x)$.
- Invert takes the function f, the trapdoor t and an image y as input, and outputs a value x such that $f(x) = y$.

Signature Scheme. A public key signature scheme is defined as a triple of polynomial-time algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$. The probabilistic key generation algorithm takes the security level κ (in unary notation) and produces a secret and public key: $\mathsf{KeyGen}(1^\kappa) = (sk, pk)$; the signature generation algorithm produces a signature: $s = \mathsf{Sign}(\mathsf{sk}, m) \in \{0,1\}^*$. The verification algorithm takes the public key, the message and the signature and decides if the signature is valid: $\mathsf{Verify}(pk, m, s) \in \{0,1\}$; we refer to these outputs as "reject" and "accept", respectively. The signature scheme is *correct* if signing a message with the secret key produces a valid signature under the matching public key:

$$(\mathsf{KeyGen}(1^\kappa) \Rightarrow (sk, pk)) \quad \Longrightarrow \quad \forall m \in \{0,1\}^* \,.\, \mathsf{Verify}\left(pk, m, \mathsf{Sign}(sk, m)\right) = 1.$$

Here and elsewhere we use $\Rightarrow$ to denote the event of the probabilistic algorithm on the left hand producing the output on the right hand, and $\Longrightarrow$ to denote logical implication.

Security is defined with respect to the Existential Unforgeability under Chosen Message Attack (EUF-CMA) game of Goldwasser *et al.* [17]. The adversary A is allowed to make a polynomial number of queries $m_i, i \in \{1, \ldots, q\}, q \leq \kappa^c$ for some c, which the challenger signs using the secret key and sends back: $s_i \leftarrow \text{Sign}(\mathsf{sk}, m_i)$. At the end of the game, the adversary must produce a pair of values (m', s') where m' was not queried before: $m' \notin \{m_i\}_{i=1}^q$. The adversary wins if $\mathsf{Verify}(pk, m', s') = 1$. In the game below, the Iverson brackets $[\![\cdot]\!]$ return 0 if the expression is False or 1 if it is True.

```
Game EUF-CMA
1: $sk, pk \leftarrow \mathsf{KeyGen}(1^\kappa)$
2: $\mathcal{M} \leftarrow \varnothing$
3: define $\mathsf{S}(m)$ as
4:   $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
5:   return $\mathsf{Sign}(sk, m)$
6: end definition
7: $(m, s) \leftarrow \mathsf{A}^{\mathsf{S}}(pk)$
8: return $[\![\mathsf{Verify}(pk, m, s) = \mathsf{True} \wedge m \notin \mathcal{M}]\!]$
```

We define the insecurity function $\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{scheme}}(Q_{\mathsf{S}}; t)$ as the maximum winning probability across all quantum adversaries that run in time t and that make at most Q_{S} signature queries.

Hash-and-Sign Signature Schemes. Given a trapdoor function family and a hash function H that hashes arbitrary messages to elements in the range of the trapdoor functions we can use the hash-and-sign construction to build a (not necessarily secure) signature scheme. The key generation algorithm simply calls the $\mathsf{GenTrapdoor}$ function to get (f, t). The public key is then the description of f, and the trapdoor t is the private key. To sign a message m, the signer uses his trapdoor t to produce a preimage s for $\mathsf{H}(m)$. This preimage is the signature for m. Lastly, to verify the validity of a signature the verifier computes $\mathsf{H}(m)$, uses the public key to evaluate f at s and checks if $f(s) = \mathsf{H}(m)$.

Merkle Tree. A Merkle tree [26] is a balanced binary tree whose root authenticates a list of data items which are contained in the leaves. Every non-leaf node, including the root, has a value equal to the hash of the concatenation of its two children. A leaf can be proven to be a member of the tree by tracing a path from the leaf to the root and listing all siblings of nodes on that path: every step can be verified by computing one hash. We associate three algorithms with a Merkle tree:

– CalculateMerkleRoot takes a list of leaf items, computes the entire Merkle tree, and returns its root.
– OpenMerklePath takes a list of leaf nodes and an index, and outputs its authentication path: the list of all siblings of nodes on the path from the indicated leaf node to the root.
– VerifyMerklePath takes an index, a leaf node, a Merkle path, and a root, and decides whether the leaf node is a member of the tree with the given root.

3 Trapdoor-Based Signature Schemes

3.1 MQ Trapdoors

Multivariate quadratic (MQ) trapdoor functions date back to the C^* scheme of Matsumoto and Imai [25], which has since given rise to a number of viable candidates including HFE_v^- [32], UOV [21] and Rainbow [11]. The idea is to compose a special quadratic map $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ with two linear transforms, $T \in \mathsf{GL}_m(\mathbb{F}_q)$ and $S \in \mathsf{GL}_n(\mathbb{F}_q)$ to obtain the public key $\mathcal{P} = T \circ \mathcal{F} \circ S$. A vector $\mathbf{s} \in \mathbb{F}_q^n$ that represents an assignment to the variables, is a valid signature for the document $d \in \{0,1\}^*$ whenever

$$\mathcal{P}(\mathbf{s}) = \mathsf{H}(d). \tag{1}$$

In order to find $\mathbf{s}$, the signer computes $\mathbf{z} = \mathsf{H}(d)$, $\mathbf{y} = T^{-1}\mathbf{z}$, uses the special structure of $\mathcal{F}$ to sample an inverse $\mathbf{x}$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$, and then computes $\mathbf{s} = S^{-1}\mathbf{x}$.

We focus on the Rainbow submission to the NIST PQC project [29], where the parameter set $(q = 256, v = 68, o_1 = 36, o_2 = 36)$ is proposed. In this case, $n = v + o_1 + o_2 = 140$ and $m = o_1 + o_2 = 72$. While the proposal does not employ Petzoldt's compression trick [33] we note that it is possible in principle, in which case $v(v+1)/2 + vo_1$ columns of the public Macaulay matrix are set as the output of a PRG expanding a seed of 32 bytes.[1] Allocating five bits per field element, we obtain signatures of 140 bytes and public keys of 356.9 kB. Without Petzoldt's compression trick the public key is 694.0 kB.

3.2 Code-Based Trapdoors

The first code-based signature scheme was proposed by Courtois, Finiasz and Sendrier (CFS) [8]; it relies on the difficulty of finding a low Hamming weight

[1] In fact, Petzoldt manages to fix more elements of the public key's Macaulay matrix, but as these elements are not arranged into columns they are incompatible with our compression technique.

word associated with a given syndrome. The public key in such a signature scheme is a parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$. A signature $(\mathbf{s}, i) \in \mathbb{F}_2^{1\times n} \times \mathbb{Z}$ on a document $d \in \{0,1\}^*$ consists of an error vector and an index; it is valid when the error vector has Hamming weight at most t and syndrome equal to the hash of the document concatenated with the index i. The index i can be thought of as selecting a different hash function. Formulaically:

$$H\mathbf{s}^{\mathsf{T}} = \mathsf{H}(d\|i) \quad \text{and} \quad \mathsf{HW}(\mathbf{s}) \leq t. \tag{2}$$

By our calculations, a 128-bit post-quantum security level is achieved with the parameter set $m = 26$, $t = 15$ and thus $n = 2^m = 2^{26}$ and $n - k = tm = 390$. At this point the public key is 3.05 GB but the signatures are 390 bits. We refer to Appendix A for a derivation of these parameters. We choose not to consider the question whether the cryptosystem is practically usable with these parameters and instead focus on the obtained compression factor. The CFS scheme is used as a generic stand-in for code-based signature schemes using the hash-and-sign paradigm and relying on the hardness of syndrome decoding.

3.3 Lattice-Based Trapdoors

A first trapdoor-based signature schemes from lattices was proposed by Goldreich, Goldwasser and Halevi (GGH) at Crypto'97 [16]. The signatures of this scheme leak information about the private key, and the scheme was broken by Nguyen and Regev [31]. Gentry et al. [15] showed how to sample signatures that do not leak information and constructed a provably secure signature scheme. Later improvements by Alwen and Peikert [3] and by Micciancio and Peikert [27] make the scheme more efficient. The main idea is the same in all schemes: the public key is a matrix $A \in \mathbb{F}_q^{n\times m}$ with large coefficients but such that there exists another matrix $S \in \mathbb{Z}^{m\times m}$ with small coefficients with $AS = 0 \bmod q$. In order to generate a signature for a document $d \in \{0,1\}^*$, the signer uses the secret key S to obtain a small-coefficient vector $\mathbf{z} \in \mathbb{Z}^m$. It is a valid signature whenever

$$A\mathbf{z} = \mathsf{H}(d) \bmod q \quad \text{and} \quad \|\mathbf{z}\|_2 \leq \beta, \tag{3}$$

for some length bound $\beta \in \mathbb{R}_{>0}$.

Using the methodology of [28], and the estimator for the concrete hardness of the SIS problem of Albrecht et al. [1], we choose parameters for the scheme of [27] that achieves 128 bits of security. This results in the parameters $n = 321, q = 2^{26} - 5, m = 16692$ and $\beta = 112296$, a public key of $n \times m \times 26$ bits = 16.6 MB, and signatures of $\lceil \log_2(\beta) \rceil \times m$ bits = 34.6 KB. We chose q to be prime as this is required for our security proof to work. The first half of the matrix A can be chosen randomly, so we can fix this part with a PRG to cut the size of the public key in half.

3.4 A Unifying View

The above three signature schemes can be thought of as variations on a common theme. These schemes are all hash-and-sign signature schemes with a linear trapdoor function $f : \mathbb{F}_q^\ell \to \mathbb{F}_q^k$, but with f restricted to a domain defined by a nonlinear constraint function $\mathsf{nc} : \mathbb{F}_q^\ell \to \{\mathsf{True}, \mathsf{False}\}$. We call these trapdoor functions **constrained linear trapdoor functions**, and if they are used in a hash-and-sign construction, we call the resulting signature scheme a **constrained linear signature scheme**.

For all the constrained linear signature schemes the public key is a matrix $M \in \mathbb{F}_q^{k\times\ell}$ with $k < \ell$ which represents the trapdoor function f and a signature is represented by a vector $\mathbf{s} \in \mathbb{F}_q^\ell$. A signature is valid if $M\mathbf{s}$ is equal to a target $\mathbf{t} \in \mathbb{F}_q^k$, which is the evaluation of a hash function at a document, and if the vector $\mathbf{s}$ also satisfies the constraint nc. Symbolically:

$$\mathsf{Verify}(sk, m, \mathbf{s}) = 1 \quad \iff \quad M\mathbf{s} = \mathbf{t} = \mathsf{H}(m) \wedge \mathsf{nc}(\mathbf{s}) = \mathsf{True}.$$

In the case of lattice-based trapdoors, the signature is valid only if $\mathbf{s}$ is a short vector. In the case of code-based trapdoors, it is valid only if the Hamming weight of $\mathbf{s}$ is low. And in the case of MQ trapdoors, the matrix M is the coefficient matrix (or Macaulay matrix) of the quadratic polynomial map $\mathcal{P}$ and the signature $\mathbf{s}$ must be factorizable as a vector of products of n variables: $\mathbf{s}^\mathsf{T} = (x_1^2, x_1x_2, \ldots, x_n^2)$. Formally, we capture this difference between MQ, code-based, and lattice-based trapdoors with the nonlinear constraint nc, namely by defining for

- code-based trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \textsc{hw}(\mathbf{s}) \le t$;
- lattice-based trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \|\mathbf{s}\|_2 \le \beta$;
- MQ trapdoors: $\mathsf{nc}(\mathbf{s}) = \mathsf{True} \Leftrightarrow \exists\, x_1, \ldots, x_n \in \mathbb{F}_q \,.\, \mathbf{s}^\mathsf{T} = (x_1^2, x_1x_2, \ldots, x_n^2)$.

3.5 Additional Security Properties

We say that a surjective trapdoor function f is one-way (OW) if it is hard to find a preimage for a randomly chosen output, and we say that f is hash-and-sign secure (HSS) if using the trapdoor function f in the hash-and-sign construction leads to a signature scheme that is EUF-CMA secure. If f is a constrained linear trapdoor function we can define stronger versions of the OW and HSS security properties that will be useful for the security analysis of the transformation (Fig. 1).

(σ, r)-One-Wayness. For any two non-negative integers $\sigma > r$ we define (σ, r)-one-wayness and (σ, r)-hash-and-sign security. To break (σ, r)-one-wayness, an adversary has to find σ preimages $\mathbf{x}_1, \ldots, \mathbf{x}_\sigma \in \mathbb{F}_q^\ell$ for σ vectors $\mathbf{y}_1, \ldots, \mathbf{y}_\sigma \in \mathbb{F}_q^k$. However, the adversary is allowed to make mistakes in each of the σ preimages it produces, as long as the errors $f(\mathbf{x}_i) - \mathbf{y_i}$ are contained in a vector space of dimension r. The $(1, 0)$-one-wayness property is identical to the one-wayness property, because the adversary only needs to find a preimage for one target and it is not allowed to make any mistakes.

The (σ, r)-OW property is a generalization of the AMQ problem introduced in [39]; an MQ trapdoor $\mathcal{P}$ is (σ, r)-one-way precisely if the Approximate MQ problem with σ targets and rank r is hard for the map $\mathcal{P}$.

(σ, r)-Hash-and-Sign Security. We also define a (σ, r)-variant of the HSS property. The security game behind this property is similar to the EUF-CMA game of the hash-and-sign signature scheme induced by f. To break this property, an adversary has to come up with a message m and σ 'signatures' $\mathbf{s}_1, \cdots, \mathbf{s}_\sigma$ such that the errors $f(\mathbf{s}_i) - \mathsf{H}(m||i)$ are contained in a a subspace of dimension r. The adversary can query a signing oracle S any (polynomially bounded) number of times. When given a message m', this signing oracle uses the trapdoor to produce preimages for $\mathsf{H}(m'||1), \cdots, \mathsf{H}(m'||\sigma)$ and returns these σ preimages. The adversary loses the game if it returns a message m for which it has queried the signing oracle, as is the case for the familiar EUF-CMA game.

We define the insecurity function $\mathsf{InSec}_f^{(\sigma,r)-\mathsf{HSS}}(Q_\mathsf{S}, Q_\mathsf{H}; t)$ as the maximal winning probability of an adversary that plays the (σ, r)-HSS game of f, that makes Q_S queries to the signing oracle, Q_H queries to the random oracle and that runs in time t. The $(1, 0)$-HSS property is equivalent to the HSS property.

Remark 1. If f is a *collision-resistant preimage-sampleable trapdoor function* (as is the case for some lattice-based trapdoor functions), the one-wayness of f can be reduced tightly to its hash-and-sign security and so OW and HSS are equivalent [15, Proposition 6.1]. Under the same assumption on f, the security proof of [15] can be modified to prove that (σ, r)-OW and (σ, r)-HSS are equivalent for all $\sigma > r \geq 0$ (Fig. 2).

4 Construction

4.1 Description

This section describes the transform of Szepieniec *et al.* but adapted to apply generically to constrained linear signature schemes. The parameters for the transformation are:

- $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, the constrained linear signature scheme to start from. We denote the hash function used in the verification algorithm by H_1 and the nonlinear constraint by nc.
- τ, the number of leaves in the Merkle tree.
- e, the extension degree of $\mathbb{F}_{q^e}$, which is the field over which the error-correcting code is defined. This value is constrained by $q^e \geq \tau$.
- ϑ, the number of Merkle paths that are opened with each new signature.
- σ, the number of signatures of the original signature scheme that is included in each signature of the new scheme.
- H_2, a hash function that outputs a α-by-k matrix over $\mathbb{F}_q$.
- H_3, a hash function that outputs a set of ϑ numbers between 1 and τ.
- H_4, a hash function used for building a Merkle tree.

The transformation outputs a new signature scheme ($\mathsf{NEW.KeyGen}$, $\mathsf{NEW.Sign}$, $\mathsf{NEW.Verify}$) with a smaller public key but larger signatures.

```
Game (σ, r)-OW
1: $(f, t) \leftarrow \mathsf{GenTrapdoor}(1^\kappa)$
2: $\mathbf{y}_1, \ldots, \mathbf{y}_\sigma \xleftarrow{\$} \mathbb{F}_q^k$
3: $\mathbf{x}_1, \ldots, \mathbf{x}_\sigma \leftarrow \mathsf{A}(f, \mathbf{y}_1, \ldots, \mathbf{y}_\sigma)$
4: return $[\![\dim(\langle f(\mathbf{x}_i) - \mathbf{y}_i \rangle_i) \le r]\!]$
```

```
Game (σ, r)-HSS
1: $(f, t) \leftarrow \mathsf{GenTrapdoor}(1^\kappa)$
2: $\mathcal{M} \leftarrow \varnothing$
3: define $\mathsf{S}(m)$ as
4:     $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$
5:     for $i$ from 1 to $\sigma$ do
6:         $\mathbf{s}_i \leftarrow \mathsf{Invert}(f, t, \mathsf{H}(m||i))$
7:     end for
8:     return $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$
9: end definition
10: $m, \mathbf{s}_1, \ldots, \mathbf{s}_\sigma \leftarrow \mathsf{A}^{\mathsf{H}, \mathsf{S}(\cdot)}(f)$
11: $d = \dim(\langle f(\mathbf{s}_i) - \mathsf{H}(m||i) \rangle_i)$
12: return $[\![(d \le r) \wedge (m \notin \mathcal{M})]\!]$
```

Fig. 1. The security game of the $(\sigma, r) - \mathsf{OW}$ property (left) and of the $(\sigma, r) - \mathsf{HSS}$ property (right).

$$\begin{tikzcd}
\text{OW} & (\sigma, r)\text{-OW} \arrow[l, Leftarrow] \\
\text{HSS} \arrow[u, Rightarrow] & (\sigma, r)\text{-HSS} \arrow[l, Leftarrow] \arrow[u, Rightarrow]
\end{tikzcd}$$

Fig. 2. Security properties of constrained linear trapdoor functions, and implications between them.

Random Linear Combinations. A signature of the new signature scheme consists of σ signatures of the original signature scheme, along with some information to verify them. The ith signature is obtained by using the signature generation algorithm of the original contrained-linear signature scheme to sign $d\|i$. It is not necessary to communicate the entire public key $M \in \mathbb{F}_q^{k \times \ell}$. Rather, it suffices to transmit a few random linear combinations of its rows. Therefore, part of the new signature consists of a matrix T that is equal to RM, where R is drawn uniformly at random from the space of $\alpha \times k$ matrices. Instead of checking whether $M\mathbf{s}_i = \mathsf{H}_1(d\|i)$, the verifier can now check wheter $T\mathbf{s}_i = R\mathsf{H}_1(d\|i)$. Obviously, if all signatures are valid, then the latter equations will also be satisfied for any matrix R. Conversely, if at least one signature is invalid, *i.e.*, $M\mathbf{s}_i \neq \mathsf{H}_1(d\|i)$ for some i, then the probability that $RM\mathbf{s} = R\mathsf{H}_1(d\|i)$ is at most $q^{-\alpha}$. By choosing α large enough, the probability of accepting an invalid signature can be made arbitrarily small.

Determining *R*. In order for the above argument to work, R must be chosen independently from $\mathbf{s} = \mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma$. Therefore, we determine R with a hash function as $R = \mathsf{H}_2(d\|\mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma)$ to ensure that a forger cannot use knowledge about R in his choice of the $\mathbf{s}_i$.

Verifying *T*. An attacker can present the verifier with a signature containing a matrix T which is totally unrelated to the matrix M. How can the verifier be sure that the matrix T that is included in the signature, is really equal to RM with $R = \mathsf{H}_2(d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma)$? We solve this problem with a probabilistic test based on an $\mathbb{F}_q$-linear error correcting code. This is a code whose alphabet consists of the elements of a finite field $\mathbb{F}_q$, with the property that any $\mathbb{F}_q$-linear combination of codewords is again a codeword. We work with Reed-Solomon Codes[2] over $\mathbb{F}_{q^e}$ with message length $L = \lceil \ell/e \rceil$ (we pack e elements of $\mathbb{F}_q$ into each symbol), codeword length τ and minimal codeword distance $D = \tau - L$. We use $\mathsf{Enc} : \mathbb{F}_{q^e}^{a\times L} \rightarrow \mathbb{F}_{q^e}^{a\times\tau}$ to denote the operation of encoding the rows of a matrix.

In the key generation phase, we compute $E = \mathsf{Enc}(M)$. Then we commit to this matrix E by building a Merkle tree whose leaves contain the columns of E, which are denoted by e_i for $i \in \{1, \ldots, \tau\}$. The new public key is the root of this tree. If $T = RM$, then by $\mathbb{F}_q$-linearity of the error correcting code, we have that $\mathsf{Enc}(T)$ is equal to $R\mathsf{Enc}(M) = RE$. Conversely, if $T \neq RM$, then $\mathsf{Enc}(T)$ and RE differ in at least one row. These rows are different codewords, so they differ in at least D of the τ symbols. To verify that $T = RM$, we now select ϑ columns $e_{b_1}, \cdots, e_{b_\vartheta}$ of E with the hash function H_3 and we check whether the b_i-th column of T agrees with Re_{b_i} for all i in $1, \cdots, \vartheta$. If T is not equal to RM, this will go undetected with a probability of at most $(\frac{L}{\tau})^\vartheta$.

Pseudocode. Algorithms 1, 2 and 3 present pseudocode for the new signature scheme ($\mathsf{NEW.KeyGen}$, $\mathsf{NEW.Sign}$, $\mathsf{NEW.Verify}$) obtained from transforming the old constrained-linear signature scheme (KeyGen, Sign, Verify).

```
Algorithm NEW.KeyGen

input: $1^\kappa$ — security level (in unary)
       random coins
output: $root$ — A public key
        $(sk, M)$ — A corresponding secret key

1: $(sk, M) \leftarrow \mathsf{KeyGen}(1^\kappa)$
2: $E \leftarrow \mathsf{Enc}(M)$                                   ▷ Encode $M$ row by row.
3: $root \leftarrow \mathsf{CalculateMerkleRoot}(e_1, \cdots, e_\tau)$   ▷ Build tree on columns of $E$
4: return $(root, (sk, M))$
```

Algorithm 1. The key generation algorithm

[2] While the original description of the transformation used MAC-polynomials, we think it is better to describe the same transformation it in the language of Reed-Solomon error correcting codes.

Key and Signature Sizes. For a post-quantum security level of κ bits, the new public key is 2κ bits in size, as it represents the Merkle root. The new signature consists of σ old signatures, α linear combinations of the rows of M (each one of which consists of ℓ field elements of size $\lceil \log_2 q \rceil$ bits), ϑ columns of $\mathsf{Enc}(M)$ (each one of which consists of k field elements of $e \times \lceil \log_2 q \rceil$ bits), and ϑ Merkle paths of consisting of $\log_2 \tau$ hash images of 2κ bits each. Put all together, we have

$$|\mathsf{NEW}.\mathit{signature}| = \sigma|\mathsf{OLD}.\mathit{signature}| + (\alpha\ell + \vartheta k e) \times \lceil \log_2 q \rceil + 2\vartheta\kappa \times \log_2 \tau. \quad (4)$$

The old signatures can be represented as ℓ field elements but in some cases a more concise encoding is possible. For instance, CFS signatures require only the positions of the 1-bits, and MQ signatures require only an assignment to the variables from which the vector of quadratic monomials can be derived.

4.2 Security

Before we present the security statement and its proof, we need to introduce a pair of security games that will be important for our security analysis. In particular, we need hash functions that are one-way and second-preimage resistant, in both cases with respect to multiple targets. Both games are formalized with respect to a hash function H that is randomly selected from a hash function family $\mathcal{H}$. We follow the formalisms of Hülsing *et al.* [20].

```
Algorithm NEW.Sign

input: $d$ — A document to sign
       $(sk, M)$ — A private key
output: $(\mathbf{s}_1, \cdots, \mathbf{s}_\sigma, T, v_{b_1}, \cdots, v_{b_\vartheta}, \mathit{paths})$ — A signature for $d$

 1: for $i$ from 1 to $\sigma$ do
 2:     $\mathbf{s}_i \leftarrow \mathsf{Sign}(d\|i, sk)$
 3: end for
 4: $R \leftarrow \mathsf{H}_2(d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma)$
 5: $T \leftarrow RM$
 6: $E \leftarrow \mathsf{Enc}(M)$                          ▷ Encode $M$ row by row.
 7: $b_1, \cdots, b_\vartheta \leftarrow \mathsf{H}_3(d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma\|T)$
 8: $\mathit{paths} \leftarrow$ empty list
 9: for $i$ from 1 to $\vartheta$ do
10:     $\mathit{paths}.\mathsf{append}(\mathsf{OpenMerklePath}(e_1, \cdots, e_\tau, b_i))$
11: end for
12: return $(\mathbf{s}_1, \cdots, \mathbf{s}_\vartheta, T, e_{b_1}, \cdots, e_{b_\vartheta}, \mathit{paths})$
```

Algorithm 2. The signature generation algorithm.

- In the *single-function, multiple-target one-wayness* (SM-OW) game, the adversary is given a list of target outputs and it wins if it can produce a single input that maps to any one of the outputs. We write $\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries and with respect to the hash function family $\mathcal{H}$ and where P is the number of target outputs.
- In the *single-function, multiple-target second-preimage resistance* (SM-SPR) game, the adversary is given a list of inputs and it wins if it can produce a second preimage that maps to the same output as any one of the input preimages. We write $\mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries and with respect to the hash function family $\mathcal{H}$ and where P is the number of input preimages.

Game SM-OW

```
1: $\mathsf{H} \xleftarrow{\$} \mathcal{H}$
2: for $i$ from 1 to $P$ do
3:     $M_i \xleftarrow{\$} \{0,1\}^m$
4:     $Y_i \leftarrow \mathsf{H}(M_i)$
5: end for
6: $M' \leftarrow \mathsf{A}^{\mathsf{H}}(Y_1, \dots, Y_P)$
7: return $[\![\exists i \,.\, \mathsf{H}(M') = Y_i]\!]$
```

Game SM-SPR

```
1: $\mathsf{H} \xleftarrow{\$} \mathcal{H}$
2: for $i$ from 1 to $P$ do
3:     $M_i \xleftarrow{\$} \{0,1\}^m$
4: end for
5: $M' \leftarrow \mathsf{A}^{\mathsf{H}}(M_1, \dots, M_P)$
6: return $[\![\exists i \,.\, \mathsf{H}(M') = Y_i \wedge M' \neq M_i]\!]$
```

Hülsing *et al.* obtain values for these insecurity functions in the random oracle model, *i.e.* where H is drawn uniformly at random from the set of all functions from the given input space to the given output space. In the classical random oracle model we have

$$\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(Q) = \mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(Q) = \frac{(Q+1)P}{|\mathsf{range}(\mathsf{H})|}. \tag{5}$$

In the quantum random oracle model, where the adversary is allowed $\hat{Q}$ quantum queries, we have

$$\mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H},P}(\hat{Q}) = \mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H},P}(\hat{Q}) = \Theta\left(\frac{(\hat{Q}+1)^2 P}{|\mathsf{range}(\mathsf{H})|}\right). \tag{6}$$

The SM-OW game does not quite capture one of the transitions in our security proof. The reason for this is that the adversary cannot be given a definite list of target output images because whether an output of the hash function is suitable for the adversary depends on the input of the hash function. We model this task by a new game, *marked element search* (MES), in which the adversary does not have a list of target outputs but a marking function $\mathsf{mark} : \mathsf{domain}(\mathsf{H}) \times \mathsf{range}(\mathsf{H}) \rightarrow \{0,1\}$ that determines whether the pair

```
Algorithm NEW.Verify
input: $d$ — document
       $(\mathbf{s}_1, \cdots, \mathbf{s}_\vartheta, T, v_{b_1}, \cdots, v_{b_\vartheta}, paths)$ — signature
       $root$ — public key
output: 1 if the signature is valid, 0 otherwise
1: $R \leftarrow \mathsf{H}_2(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma)$
2: for $i$ from 1 to $\sigma$ do
3:     if $T\mathbf{s}_i \neq R\mathsf{H}_1(d \| i)$ or $\mathsf{nc}(\mathbf{s}_i) = \mathsf{False}$ then
4:         return 0
5:     end if
6: end for
7: $b_1, \cdots, b_\vartheta \leftarrow \mathsf{H}_3(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma \| T)$
8: for $i$ from 1 to $\vartheta$ do
9:     if $\mathsf{Enc}(T)_{*,b_i} \neq Re_{b_i}$ then
10:        return 0
11:    end if
12:    if $\mathsf{VerifyMerklePath}(b_i, e_{b_i}, paths[i], root) = \mathsf{Fail}$ then
13:        return 0
14:    end if
15: end for
16: return 1
```

Algorithm 3. The signature verification algorithm.

(*input*, *output*) is suitable. We write $\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H},\mathsf{mark}}(Q)$ to denote the maximum success probability across all adversaries that make at most Q queries to the hash oracle in the MES game. In the quantum random oracle model this notion is reducible to SM-OW.

```
Game MES
1: $\mathsf{H} \xleftarrow{\$} \mathcal{H}$
2: $M \leftarrow \mathsf{A}^{\mathsf{H}}()$
3: return $\mathsf{mark}(M, \mathsf{H}(M))$
```

Proposition 1 (SM-OW $\leq$ MES)**.** *In the (quantum) random oracle model, we have that for any marking function* mark *with* $P = \max_X |\{Y \mid \mathsf{mark}(X, Y) = 1\}|$,

$$\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H},\mathsf{mark}}(Q) \leq \mathsf{InSec}^{\mathrm{SM\text{-}OW}}_{\mathsf{H},P}(Q). \tag{7}$$

Proof. We show an algorithm, $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ in the SM-OW game, that simulates a given algorithm $\mathsf{A}_{\mathsf{MES}}$ for the MES game with marking function mark, and wins with at least the same probability. The input of $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is a list of P images $\{Y_1, \ldots, Y_P\}$ and access to a random oracle H. The algorithm $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ programs a random oracle H' that on input X returns $\sigma_X^{-1}(\mathsf{H}(X))$, where σ_X is a permutation (chosen deterministically) with the property that the elements Y that satisfy $\mathsf{mark}(X, Y) = 1$ are mapped into the set $\{Y_1, \ldots, Y_P\}$. By assumption, $|\{Y \mid \mathsf{mark}(X, Y) = 1\}| \leq P$, so such a permutation always exists. Note that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is bounded in the number of queries it can make to H, but not bounded in time or memory. Therefore it will be able to choose such a permutation σ_X. Then, $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ invokes $\mathsf{A}_{\mathsf{MES}}$ with the programmed random oracle H'. Since H' only applies a permutation to the ouput of H, the ouputs of H' will be independent and uniformly distributed. Hence, H' is itself a perfect random oracle. Pseudocode for $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ is given below.

Algorithm $\mathsf{B}_{\mathsf{SM\text{-}OW}}$

```
1: define $\mathsf{H}'(X)$ as
2:     pick $\sigma_X$ s.t. $\sigma_X(\{Y \mid \mathsf{mark}(X, Y) = 1\}) \subset \{Y_1, \cdots, Y_P\}$
3:     return $\sigma_X^{-1} \circ \mathsf{H}(X)$
4: end definition
5: return $\mathsf{A}_{\mathsf{MES}}^{\mathsf{H}'(\cdot)}()$
```

Clearly, the number of queries that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ makes to H is identical to the number of queries made by the simulated algorithm $\mathsf{A}_{\mathsf{MES}}$. Eventually, $\mathsf{A}_{\mathsf{MES}}$ returns a preimage X. $\mathsf{A}_{\mathsf{MES}}$ wins the MES game if $\mathsf{mark}(X, \sigma_X^{-1}(\mathsf{H}(X))) = \mathsf{True}$. By our choice of σ_X this implies that $\sigma_X(\sigma_X^{-1}(\mathsf{H}(X))) = \mathsf{H}(X) \in \{Y_1, \cdots, Y_P\}$, which shows that $\mathsf{B}_{\mathsf{SM\text{-}OW}}$ wins his SM-OW game in this case. So $\mathsf{InSec}^{\mathsf{MES}}_{\mathsf{H},\mathsf{mark}}(Q) \leq \mathsf{InSec}^{\mathsf{SM\text{-}OW}}_{\mathsf{H},P}(Q)$. □

We are now in a position to state and prove our security claim.

Theorem 1. *Let* NEW *be the signature scheme derived from applying the transformation to a constrained linear scheme* OLD. *The maximum winning probability across all time-t adversaries in the EUF-CMA game against* NEW *that make* Q_s *signature queries and* Q_1, Q_2, Q_3, Q_4 *queries to the random oracles* $\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4$ *respectively is bounded by*

$$\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{NEW}}(Q_s, Q_1, Q_2, Q_3, Q_4; t) \leq \mathsf{InSec}^{(\sigma,\mathrm{r})\text{-HSS}}_{f}(Q_s, Q_1; O(t)) + \mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H}_4, 2\tau-1}(Q_4)$$
$$+ \mathsf{InSec}^{\text{SM}-\text{OW}}_{\mathsf{H}_3, L^{\vartheta}}(Q_3) + \mathsf{InSec}^{\text{SM}-\text{OW}}_{\mathsf{H}_2, q^{\alpha \times (k-r+1)}}(Q_2). \quad (8)$$

Proof. We show through a sequence of four games how an adversary for the EUF-CMA game against NEW can be transformed into an adversary for the (σ, r)-HSS property of the underlying constrained linear trapdoor function f that wins with the same probability conditional on each of the transitions being successful. By bounding the failure probability of each transition and summing the terms we obtain a bound on the winning probability of the adversary against NEW. The sequence of games is as follows:

- The first game G_1 is the EUF-CMA game against NEW.
- The second game G_2 drops the Merkle tree. Instead, the public key consists of all the τ columns of E, and the verifier checks directly if the columns that are included in the signature are correct.
- The game G_3 drops the codeword identity testing. Instead, the public key is now the original public key (*i.e.*, M), and the verifier tests directly if the matrix T, which is included in the signature is equal to RM.
- The last game G_4 drops the random linear combinations for signature validity testing, instead G_4 is won if the errors $f(\mathbf{s_i}) - \mathsf{H}_1(m||i)$ are contained in a subspace of dimension r. G_4 is thus the (σ, r)-HSS game for the constrained linear trapdoor function f.

In games G_2, G_3 and G_4, the adversary B simulates the previous game's adversary A in order to win his own game. In particular, this means that B must answer the signing queries that A makes. This is not a problem, because in all cases B can just forward the queries that A makes to its own signing oracle, remove some information that is not required for the game that A is playing from the signature and pass the response back to A. In each case, we define the transition's failure probability as the probability that A wins but B does not. In all cases the adversary A has unbridled access (perhaps even quantum access) to the hash functions H_1, H_2, H_3 and H_4.

The event that A wins G_1 but B does not win G_2 occurs only if the signature outputted by A passes the Merkle root test, but the columns included in this signature do not agree with the columns in $E = \mathsf{Enc}(M)$. This event requires finding a second preimage for one of the $2\tau - 1$ nodes of the Merkle tree, so the failure probability is bounded by

$$\mathsf{InSec}^{\text{SM-SPR}}_{\mathsf{H}_4, 2\tau-1}(Q_4)\,.$$

Likewise, the event that A wins the G_2 game, but B does not win the G_3 game occurs only if the columns $e_{b_1}, \cdots, e_{b_\vartheta}$ of E in the signature outputted by A are correct, but still T is not equal to RM. This implies that $\mathsf{Enc}(T)$ differs from RE in at least $\tau - L$ columns (since the rows are codewords from a code with minimal distance $\tau - L$), but that none of these columns were not chosen by the random oracle H_3. Finding $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma||T$, such that this happens is a marked element search with marking function

$$\mathsf{mark}_1(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma||T, b_1||\cdots||b_\vartheta) = \begin{cases} \mathsf{False} & \text{if } T = RM \\ \mathsf{False} & Re_{b_i} \neq \mathsf{Enc}(T)_{\star, b_i} \text{ for some } i \\ \mathsf{True} & \text{otherwise} \end{cases}.$$

Since there are at most L indices for which the columns of $\mathsf{Enc}(T)$ and $R\mathsf{Enc}(E)$ are identical, there are at most $\binom{L}{\vartheta} \leq L^\vartheta$ marked elements for a given input. The failure probability is therefore bounded by

$$\mathsf{InSec}^{\text{MES}}_{\mathsf{H}_3, \mathsf{mark}_1}(Q_3) \leq \mathsf{InSec}^{\text{SM-OW}}_{\mathsf{H}_3, L^\vartheta}(Q_3)\,.$$

Finally, the event that A wins game G_3 but that B does not win G_4 happens when the errors span a vector space of dimension strictly larger than r (B does not win), but that all these error lie in the kernel of $R = \mathsf{H}_2(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma)$ (otherwise A does not win). Finding $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$ such that this happens is a marked element search for the marking function

$$\mathsf{mark}_2(m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma, R) = \begin{cases} \mathsf{False} & \text{if } R(f(\mathbf{s}_i) - \mathsf{H}_1(m||i)) \neq 0 \text{ for some } i \\ \mathsf{False} & \text{if } \dim(\langle f(\mathsf{s}_i) - \mathsf{H}_1(m||i)\rangle_{i=0,\cdots,\sigma}) > r \\ \mathsf{True} & \text{otherwise} \end{cases} .$$

For a choice of $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$ there are only good matrices R if the space spanned by the errors $f(\mathsf{s}_i) - \mathsf{H}_1(m||i)$ has dimension at least $r+1$. If this is the case then the good matrices R are precisely the α-by-k matrices whose kernel contains the error space. Therefore there are at most $q^{\alpha(k-r+1)}$ good matrices for each choice of $m||\mathbf{s}_1||\cdots||\mathbf{s}_\sigma$. Therefore the failure probability of the last step is bounded by

$$\mathsf{InSec}^{\mathrm{MES}}_{\mathsf{H}_2,\mathsf{mark}_2}(Q_2) \leq \mathsf{InSec}^{\mathrm{SM-OW}}_{\mathsf{H}_2,q^{\alpha\times(k-r+1)}}(Q_2)\,. \qquad \square$$

Joining Theorem 1 with Eqs. (5) and (6) gives the following corollaries.

Corollary 1. *In the classical random oracle model,*

$$\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{NEW}}(Q_s, Q_1, Q_2, Q_3, Q_4; t) \leq \mathsf{InSec}^{(\sigma,\,r)\text{-HSS}}_{f}(Q_s, Q_1; t) + (Q_2+1)q^{-\alpha(r+1)} \\ +(Q_3+1)(\ell/\tau)^{\vartheta} + (Q_4+1)(2\tau-1)/2^{\kappa}.$$

Corollary 2. *In the quantum random oracle model,*

$$\mathsf{InSec}^{\text{EUF-CMA}}_{\mathsf{NEW}}(Q_s, \hat{Q}_1, \hat{Q}_2, \hat{Q}_3, \hat{Q}_4; t) \leq \mathsf{InSec}^{(\sigma,\,r)\text{-HSS}}_{f}(Q_s, \hat{Q}_1; t) + \Theta\left((\hat{Q}_2+1)^2 q^{-\alpha(r+1)}\right) \\ +\Theta\left((\hat{Q}_3+1)^2(\ell/\tau)^{\vartheta}\right) + \Theta\left((\hat{Q}_4+1)^2(2\tau-1)/2^{\kappa}\right).$$

There are two ways to use the transformation. One can choose $\sigma = 1$ and α large enough such that $q^{\alpha/2}$ reaches the required post-quantum security level, *i.e.*, $q^{\alpha/2} > 2^{\kappa}$. Corollary 2 with $r = 0$ then guarantees that the resulting signature scheme is EUF-CMA secure, provided that the constrained linear trapdoor function f that we started from is $(1,0)$-HSS. This assumption is equivalent to the EUF-CMA security of the original signature scheme OLD. We also note that in this case the security proof is tight, meaning that no security is lost (in the QROM) by applying the transformation in this way.

One can also use the transformation with $\sigma > r$, and a lower value of α such that $q^{\alpha\cdot(r+1)/2}$ reaches the required security level. This reduces the size of the public keys even further, but this comes at the cost of a stronger security assumption on the constrained linear trapdoor function f. In this case Corollary 2 says that the resulting signature scheme is EUF-CMA secure, if the underlying constrained linear trapdoor function is (σ, r)-HSS.

4.3 Applying the Transformation

Table 1 presents a comparison of the transformation applied to the three constrained linear trapdoor signature schemes treated in Sect. 3. For the Rainbow and Micciancio-Peikert schemes part of the public key can be generated with a PRNG to reduce the size of the public key. This trick is compatible with our construction, so we have taken this into account. In all cases, 128 bits of security against quantum computers was targeted for an apples-to-apples comparison.

Table 1. Comparison of constrained linear signature schemes before and after public key compression. Legend: NC = no compression; PS = our provably secure technique based on the assumption that the original hash-and-sign signature scheme is secure; SA = the approach relying on stronger assumptions.

Scheme	q	Other parameters	α	σ	ϑ	τ	e	$\|pk\|$	$\|sig\|$
Rainbow NC	256	$v = 68, o_1 = 36, o_2 = 36$	-	-	-	-	-	0.35 MB	0.14 kB
Rainbow PS			32	1	25	2^{20}	3	64 bytes	0.18 MB
Rainbow SA			2	16	25	2^{20}	3	64 bytes	35.51 kB
CFS NC	2	$m = 26, t = 15$	-	-	-	-	-	3.05 GB	59 bytes
CFS PS			256	1	71	2^{25}	25	32 bytes	2.00 GB
CFS SA			1	256	71	2^{25}	25	32 bytes	8.15 MB
Micciancio-Peikert NC	$2^{26} - 5$	$n = 321, m = 16692, \beta = 112296$	-	-	-	-	-	8.30 MB	34.64 kB
Micciancio-Peikert PS			10	1	37	2^{20}	1	64 bytes	0.35 MB
Micciancio-Peikert SA			5	2	37	2^{20}	1	64 bytes	0.26 MB

The shrinkage is the most striking when $k \gg \alpha \cdot \sigma$, because this is when the largest part of the matrix M is omitted. The mediocre shrinkage of $|pk| + |sig|$ for the provably secure case ($\sigma = 1$) suggests that for the trapdoors considered, k is already quite close to the lower bound $k \geq \kappa/\log_2 q$ needed for κ bits of security. The greater compression factor attained when $\sigma > 1$ is due mostly to the representation of the old signatures in far less than $\ell \cdot \log_2 q$ bits.

5 Conclusion

This paper generalizes the construction of Szepieniec *et al.* [39] to a wide class of signature schemes called constrained linear signature schemes. This construction transforms a constrained linear signature scheme into a new signature scheme with tiny public keys, at the cost of larger signatures and while reducing their combined size. We prove the EUF-CMA security of the resulting signature scheme in the quantum random oracle model, and for a more aggressive parametrization we identify the (σ, r)-hash-and-sign security notion as a sufficient property for security. This improves the understanding of the security of instantiations of this construction, which includes the DualModeMS submission to the NIST PQC standardization project [12, 29]. Finally, to showcase the generality and facilitate comparison, the construction is applied to an $\mathcal{MQ}$-based, a code-based and a lattice-based signature scheme, all targeting the same security level. In some cases the combined size of a signature and a public key can be reduced by more than a factor 300.

We close with some notes on the practicality of the transformation. From Table 1 we see that our transformation improves the practicality of state of the art multivariate and code-based signature schemes for applications such as public key infrastructure (PKI), where the metric $|\mathsf{sig}| + |\mathsf{pk}|$ is important and the performance of signing a message is less critical (most signatures in a PKI chain are long-lived and need not be created often). Code-based signature schemes remain not very practical, despite the improvements our construction makes. For example, applying the construction to the CFS scheme results in signatures of 8.15 MB. Still, if better code based signature schemes are developed, the construction will likely to be able to improve the quantity $|\mathsf{sig}|+|\mathsf{pk}|$. For example, even though the pqsigRM [22] proposal to the NIST PQC project does not have a completely unstructured matrix as public key, our construction can still reduce $|\mathsf{sig}| + |\mathsf{pk}|$ by a factor 6 from 329 kB to 60 kB in this case (with $\alpha = 4, \sigma = 64$). Unfortunately, comments on the NIST forum indicate that the pqsigRM proposal might not be secure [2].

State of the art hash-and-sign lattice-based signature schemes are built on structured lattices to achieve smaller public keys (e.g. Falcon relies on NTRU lattices [14]). Therefore, our construction does not improve on state of the art lattice-based schemes. Rather, our construction can be seen as an alternative to using structured lattices that provably does not deteriorate the security of the original schemes. In contrast, it is possible that switching to structured lattices has a negative impact on security.

Acknowledgements. This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the EC H2020 FENTEC under grant agreement No 780108. In addition, this work was supported by imec through ICON Diskman and by FWO through SBO SPITE S002417N. Ward Beullens is funded by an FWO SB fellowship. Alan Szepieniec is being supported by a doctoral grant from the Flemish Agency for Innovation and Entrepreneurship (VLAIO, formerly IWT).

A CFS Parameters

Perhaps surprisingly, the most efficient attack on the CFS cryptosystem is not information set decoding (as is the case for the closely related Niederreiter cryptosystem) but a generalized birthday algorithm credited to Bleichenbacher by Finiasz and Sendrier [13]. The offline phase of this attack consists of building three lists L_0, L_1, L_2 containing sums of respectively w_0, w_1, w_2 columns from H, where $t = w_0 + w_1 + w_2$. Next, L_0 and L_1 are merged and pruned by taking the sum of each pair and keeping it only if it starts with λ zeros; the result of this operation is stored in L_0'. In the online phase a random counter i is appended to the document and the sum of $\mathsf{H}(d\|i)$ with every element of L_2 that agrees on the first λ positions is looked up in L_0'—because if this sum is present then that means that $\mathsf{H}(d\|i)$ equals the sum of $w_1 + w_2 + w_3 = t$ columns of H which can be identified by tracing the origins of the elements from L_0', L_2, L_0, L_1 that were used. Let L_1' denote the list obtained from pruning the sums of elements of L_2 and $\mathsf{H}(d\|i)$.

A single trial is successful if there is a collision between L_0' and L_1'. This is essentially a generalized birthday problem as described by Wendl [41], and the same result shows that the much more easily computed binomial distribution approximates the probability of zero collisions very well when this quantity is overwhelming. The number of pairs to consider is $\#L_0' \times \#L_1'$ and the proportion of pairs representing a collision is $1/2^{k-\lambda}$. All considered pairs fail to collide with probability $(1-2^{\lambda-k})^{\#L_0 \times \#L_1}$. By approximating $\#L_0' \approx \mathrm{E}[\#L_0'] = 2^{-\lambda}\binom{n}{w_0+w_1}$ and $\#L_1' \approx \mathrm{E}[\#L_1'] = 2^{-\lambda}\binom{n}{w_2}$ we have a probability of success of

$$\mathrm{P}_s = 1 - \left(1 - 2^{\lambda-k}\right)^{2^{-2\lambda}\binom{n}{w_0+w_1}\binom{n}{w_2}} \tag{9}$$

$$\approx 2^{-\lambda-k}\binom{n}{w_0+w_1}\binom{n}{w_2} + O(2^{2(\lambda-k)}). \tag{10}$$

The online complexity is $O(\mathrm{C} \cdot \mathrm{P}_s^{-1})$. The offline complexity is dominated by sorting the largest list of L_0, L_1 and L_2, as merging L_0 and L_1 can be done in linear time. Therefore, the offline complexity is $O\left(\binom{n}{\lceil w/3 \rceil}\log_2\binom{n}{\lceil w/3 \rceil}\right)$.

Quantumly, there is no speed-up for sorting, and so the offline phase might as well remain classical. The online phase can be improved by applying Grover's algorithm to the "random" guess for the counter i. While sorted list lookup requires only $\frac{1}{\pi}(\ln(n) - 1)$ operations [19], this speed-up factor is hidden by the big-O. The λ that minimizes the online quantum complexity $O(\mathrm{C} \cdot \mathrm{P}_s^{-1/2})$ is small enough to make the offline complexity the algorithm's bottleneck. All complexities are larger than 2^{128} for the parameter set $m = 26, t = 15$, with $\lambda = 31$ being the smallest such value for which the offline complexity is larger than the quantum online complexity. At this point the public key is a bit matrix of $(15{\cdot}26) \times 2^{26}$ elements, or roughly 3.05 GB. In contrast, a signature represents a bitstring of length 2^{26} and of Hamming weight 15, which can be straightforwardly represented as 15 integers of 26 bits each, by 390 bits in total.

References

1. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)
2. Alperin-Sheriff, J., Lee, Y., Perlner, R., Lee, W., Moody, D.: Official comments on pqsigRM (2018). https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/pqsigRM-official-comment.pdf
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: Albers, S., Marion, J. (eds.) 26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009. Proceedings of LIPIcs, Freiburg, Germany, 26–28 February 2009, vol. 3, pp. 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009). https://doi.org/10.4230/LIPIcs.STACS.2009.1832

4. Aumasson, J.P., Endignoux, G.: Improving stateless hash-based signatures. Cryptology ePrint Archive, Report 2017/933 (2017). http://eprint.iacr.org/2017/933
5. Bernstein, D., et al.: SPHINCS: Practical Stateless Hash-Based Signatures. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 368–397. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_15
6. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_3
7. Chen, M.-S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 135–165. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_5
8. Courtois, N.T., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_10
9. Debris-Alazard, T., Sendrier, N., Tillich, J.: A new signature scheme based on (U|U+V) codes. IACR Cryptology ePrint Archive 2017/662 (2017). http://eprint.iacr.org/2017/662
10. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (1976). https://doi.org/10.1109/TIT.1976.1055638
11. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_12
12. Faugère, J.C., Perret, L., Ryckeghem, J.: DualModeMS: a dual mode for Multivariate-based signature 20170918 draft. UPMC-Paris 6 Sorbonne Universités; INRIA Paris; CNRS (2017)
13. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_6. [24]
14. Fouque, P.A., et al.: Falcon (2017). submission to the NIST PQC project
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 17–20 May 2008, pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407
16. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112–131. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052231
17. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988). https://doi.org/10.1137/0217017
18. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: a signature scheme for embedded systems. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 530–547. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33027-8_31
19. Høyer, P., Neerbek, J., Shi, Y.: Quantum complexities of ordered searching, sorting, and element distinctness. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) ICALP 2001. LNCS, vol. 2076, pp. 346–357. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-48224-5_29

20. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 387–416. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_15
21. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
22. Lee, W., Kim, Y.S., Lee, Y.W., Kim, J.-S.: pqsigRM (2017). submission to the NIST PQC project
23. Lyubashevsky, V.: Fiat-Shamir with aborts: applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_35. [24]
24. Matsui, M. (ed.): ASIACRYPT 2009. LNCS, vol. 5912. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7
25. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_39
26. Merkle, R.C., Charles, R., et al.: Secrecy, authentication, and public key systems (1979)
27. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. IACR Cryptology ePrint Archive 2011/501 (2011). http://eprint.iacr.org/2011/501
28. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5
29. National Institute for Standards and Technology (NIST): post-quantum crypto standardization (2018). http://csrc.nist.gov/groups/ST/post-quantum-crypto/
30. National Institute of Standards and Technology: FIPS PUB 186–4: Digital Signature Standard (DSS) (2013). http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf
31. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: cryptanalysis of GGH and NTRU signatures. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 271–288. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_17
32. Patarin, J.: Hidden fields equations (HFE) and Isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_4
33. Petzoldt, A., Bulygin, S., Buchmann, J.: CyclicRainbow – a multivariate signature scheme with a partially cyclic public key. In: Gong, G., Gupta, K.C. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 33–48. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17401-8_4
34. Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., Ding, J.: Design principles for HFEv- based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 311–334. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_14
35. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978). https://doi.org/10.1145/359340.359342

36. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_22
37. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20–22 November 1994, pp. 124–134. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365700
38. Stern, J.: A new paradigm for public key identification. IEEE Trans. Inf. Theor. **42**(6), 1757–1768 (1996). https://doi.org/10.1109/18.556672
39. Szepieniec, A., Beullens, W., Preneel, B.: MQ signatures for PKI. In: Lange, T., Takagi, T. (eds.) PQCrypto 2017. LNCS, vol. 10346, pp. 224–240. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59879-6_13
40. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 755–784. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_25
41. Wendl, M.C.: Collision probability between sets of random variables. Stat. Probab. Lett. **64**(3), 249–254 (2003)