MQ Signatures for PKI

Alan Szepieniec and Ward Beullens and Bart Preneel

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

Abstract. It is well known that multivariate quadratic (MQ) digital signature schemes have small signatures but huge public keys. However, in some settings, such as public key infrastructure (PKI), both variables are important. This paper explains how to transform any MQ signature scheme into one with a much smaller public key at the cost of a larger signature. The transformation aims to reduce the combined size of the public key and signature and this metric is improved significantly. The security of our transformation reduces to that of the underlying MQ signature scheme in the random oracle model. It is possible to decrease signature sizes even further but then its security is related to the conjectured hardness of a new problem, the Approximate MQ Problem (AMQ).

Keywords: multivariate quadratic, public key infrastructure, signature, random oracle, post-quantum, hard problem

1 Introduction

Post-quantum cryptography is gaining in popularity in recent years, largely due to the promise of Shor's algorithms to break most deployed public-key cryptography as soon as large enough quantum computers are built [25]. For instance, NIST [17] is looking to standardize one or more quantum-resistant public-key cryptographic algorithms [18]. The EU-funded PQCRYPTO project aims to develop a portfolio of fast and highly secure implementations of post-quantum cryptosystems [28]. The conference of the same name has been held semi-anually since 2006 with larger turnouts every edition [2]. Perhaps the most noteworthy illustration of the increased consideration afforded to post-quantum cryptography is the experimental but successful adoption of the New Hope key establishment algorithm by Google in Chrome browsers [1,15].

While certainly a step forward, the deployment of the New Hope key establishment algorithm only protects users against passive eavesdroppers. An active attacker can launch a man-in-the-middle attack and fool Alice and Bob into establishing a secure channel with the *attacker*, rather than directly with one other. Alice and Bob can *sign* their messages to guarantee authenticity and thus foil the attack. However, this countermeasure does not fundamentally solve the

problem as it requires that either Alice or Bob knows the other's public key, or at the very least is capable of verifying its authenticity when they receive it.

Public Key Infrastructure (PKI) solves this problem with certificates that authenticate the transmitted public key. The certificate itself is a linked list of public keys and signatures, where each signature authenticates the next public key under the previous one. The first public key in this link is the root public key of a Certificate Authority (CA), which in the case of web traffic is built into the user's browser.

The transmission of the certificate constitutes a significant bandwidth cost in any key establishment protocol and should consequently be minimized. However, most current proposals for post-quantum signatures do not seem to take this particular use case into account. By and large, post-quantum signature schemes fall into two camps. In camp (1) public keys are small but the signatures are huge. This is the case for hash-based signatures such as SPHINCS [3] and signatures based on non-interactive zero-knowledge proofs such as the MQ-based SSH protocol [24] and the subsequent MQDSS [7] or Stern's code-based identification scheme [27]. By contrast, in camp (2) the signatures are small but the public keys are huge, such as is the case for well-known MQ signature schemes such as UOV [14] or HFEv^- [19,21] but also notably the code-based trapdoor schemes such as CFS and derivates [8]. The odd exception to this polarization is the lattice-based BLISS [10] whose public keys and signatures clock in at roughly the same size.

In the case of PKI, only the root public key is not transmitted as part of the certificate as it is assumed to be present on the client already. For this purpose, camp (2) is ideal as it increases the certificate size by the smallest amount. At the other end of the chain, the public key should be small as it is transmitted every time; but more importantly its signature generation algorithm should be fast as it must produce new signatures every time the protocol runs — in contrast to the certificate itself, which can be copied straight from memory. Therefore, fast representatives from camp (1) or the odd exception between camps seem more suited for the tail end of the chain. In the middle of the chain, signatures are generated relatively infrequently and the chief concern is not so much the cost of the signature generation algorithm but rather the sizes of the public keys and signatures. Between these two size variables, one should not minimize one at the expense of the other, but rather *both variables at the same time*.

Fig. 1 plots several signature schemes and their positions in the quarter plane spanned by the signature size and public key size axes. While ECDSA enjoys both very short signatures and public keys, it offers zero security against quantum computers.

In this paper we present a generic transformation that turns MQ signature schemes — whose public keys are huge and whose signatures are small — into a new signature scheme with smaller public keys and larger signatures. The objective is a new signature scheme whose public keys pk and signatures s solve

$$\mathsf{min}\big(|\mathsf{pk}| + |s|\big) \ . \tag{1}$$

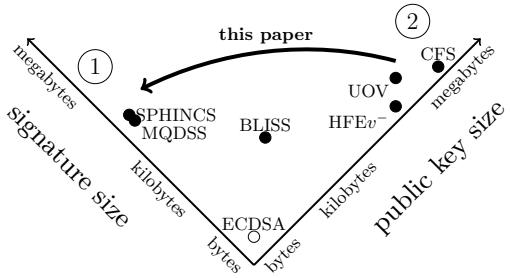

Fig. 1. Diagram of various digital signature schemes laid out according to signature and public key size.

It should be noted that it is easy to transform any representative from camp (2) to one from camp (1) using hash functions. Just replace the public key with its hash digest, and include the original public key as part of the new signatures. This naïve transformation does not improve the target quantity. However, as there is no equivalent reverse transformation, it shows perhaps that camp (2) is the more fruitful starting point.

Indeed, our transform can be thought of as applying the above naïve transformation and stopping half-way. Instead of including the entire original public key in the signatures, we *include only a small portion* of it — but just enough to keep the verification procedure meaningful. Which portion is to be included, is decided by the random oracle after being queried with the signature. Lastly, a small set of Merkle tree paths ending in linearly homomorphic MACs allows the verifier to verify that the released portion of the original public key matches the Merkle root, which is the new public key.

It is possible to choose parameters for which our transform generates a new signature scheme whose security reduces cleanly to that of the underlying MQ signature scheme. For parameters that lead to even smaller signatures we have no security proof but we are able to relate forgery to a hard problem called the *Approximate MQ Problem* (AMQ), which generalizes the MQ problem to allow erroneous solutions, as long as the errors live in a consistent small-dimension subspace. We offer several arguments supporting the hardness of the AMQ Problem.

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 $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\kappa$ 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*.

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: $\mathsf{KeyGen}(1^\kappa) = (\mathsf{sk}, \mathsf{pk})$; 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: $\mathsf{Verify}(\mathsf{pk}, m, s) \in \{0,1\}$. The signature scheme is *correct* if signing a message with the secret key produces a valid signature under the matching public key:

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

Security is defined with respect to the Existential Unforgeability under Chosen Message Attack (EUF-CMA) game [12] between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$, both polynomial-time Turing machines. The challenger generates a key pair and sends the public key to the adversary. The adversary 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: $s_i \leftarrow \mathsf{Sign}(\mathsf{sk}, m_i)$. 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 $\mathsf{Verify}(\mathsf{pk}, m', s') = 1$. A signature scheme is secure in the EUF-CMA model if for all quantum polynomial-time adversaries $\mathcal{A}$, the probability of winning is negligible, *i.e.*, drops faster than any polynomial's reciprocal:

$$\forall c > 1 \,.\, \exists N \in \mathbb{N} \,.\, \forall \kappa > N \,.\, \forall \mathcal{A} \,.$$
$$\Pr\left[\begin{array}{l|l} \mathsf{Verify}(\mathsf{pk}, m', s') = 1 & (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\kappa) \\ \wedge\, m' \notin \{m_i\}_{i=1}^q & (\{m_i, s_i\}_{i=1}^{q<\kappa^c}, m', s') \leftarrow \langle \mathcal{C}(\mathsf{sk}), \mathcal{A} \rangle (\mathsf{pk}) \end{array}\right] \leq \frac{1}{\kappa^c} \ .$$

3 Multivariate Quadratic Signature Schemes

Multivariate quadratic (MQ) cryptosystems rely on the cryptographic hardness of the MQ Problem, which asks to find a satisfying solution $\mathbf{x} \in \mathbb{F}_q^n$ to a list of m multivariate quadratic polynomials $\mathcal{P} \in (\mathbb{F}_q[\mathbf{x}])^m$. This problem is **NP**-hard as well as empirically hard on average, requiring an exponential running time for solution by state-of-the-art algorithms whenever $m \approx n$. This paper, and all other MQ-based cryptography, assumes that the MQ Problem is hard.

MQ Problem. Given: a list of m multivariate quadratic polynomials $\mathcal{P}(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}))^\mathsf{T}$ over a finite field $\mathbb{F}_q$ in n variables $(x_1, \ldots, x_n)^\mathsf{T} = \mathbf{x} \in \mathbb{F}_q^n$. Find an assignment to $\mathbf{x}$ that satisfies $p_1(\mathbf{x}) = \ldots = p_m(\mathbf{x}) = 0$. We write MQ to denote the problem class and $\mathsf{MQ}[m, n]$ to make the parameters explicit.

MQ Assumption. If $m \approx n$, there is no polynomial-time quantum computer that solves generic instances of $\mathsf{MQ}[m, n]$.

The MQ signature schemes considered in this paper have public keys that contain a trapdoor. The signature verification algorithm consists of evaluating the public key $\mathcal{P}$ in the signature $\mathbf{s} \in \mathbb{F}_q^n$, and checking whether this evaluation

results in the hash of the message $m \in \{0,1\}^*$ lifted to $\mathbb{F}_q^m$: $\mathcal{P}(\mathbf{s}) \stackrel{?}{=} \mathsf{H}(m)$. In order to sign a message, the signer must know a secret decomposition of $\mathcal{P}$ into $\mathcal{P} = T \circ \mathcal{F} \circ S$ where T and S are affine and where $\mathcal{F}$ is efficiently invertible[1]. Therefore, in addition to the MQ Problem, these signature schemes rely on the Extended Isomorphism of Polynomials (EIP) Problem, which asks to recover the factorization $T, \mathcal{F}, S$ from $\mathcal{P}$.

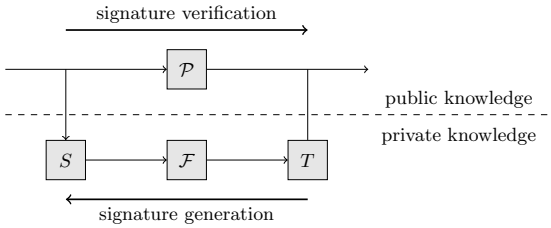

Fig. 2. Schematic representation of multivariate quadratic signature schemes.

The EIP problem is not hard in general. Rather, the central map $\mathcal{F}$ requires careful design to resist all known attacks. We recommend relying on the HFEv^- [21] or UOV [14] signature schemes, as these have remained unbroken for close to two decades. We omit a formal treatment of the EIP problem as it is not relevant to our transformation and it is assumed to be hard for the underlying signature scheme anyway.

3.1 Approximate MQ

Unfortunately, not all instances of our construction have a clean reduction towards the underlying MQ signature scheme. Instead, we relate their security to a new computational problem called the *Approximate MQ Problem* (AMQ for short). Roughly speaking, the AMQ Problem is a weaker variant of the MQ problem where the solution does not have to be exact; rather, the errors have to live in a subspace of small dimension.

AMQ Problem. Let $m, n, v, r \in \mathbb{N}$ be integers with $r < m$ and $r < v$. Given a list of m multivariate quadratic polynomials $\mathcal{P}(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_m(\mathbf{x}))^\mathsf{T}$ over a finite field $\mathbb{F}_q$ in n variables $(x_1, \ldots, x_n)^\mathsf{T} = \mathbf{x} \in \mathbb{F}_q^n$, and a list of v target vectors $\mathbf{y}_1, \ldots, \mathbf{y}_v \in \mathbb{F}_q^m$. Find a list of v vectors $\mathbf{x}_1, \ldots, \mathbf{x}_v \in \mathbb{F}_q^n$ such that

$$\mathsf{dim}\,\langle \{\mathcal{P}(\mathbf{x}_i) - \mathbf{y}_i\}_{i=1}^v \rangle \leq r \ .$$

We write AMQ to denote the problem class and write $\mathsf{AMQ}[m, n, v, r]$ to make the parameters explicit.

[1] If $n > m$, which is necessary for MQ signature schemes, any image is likely to have multiple inverses. By "efficiently invertible" we mean that there is an efficient algorithm to sample from the inverse set of any given image.

Obviously, we have $\mathsf{MQ}[m,n] \equiv \mathsf{AMQ}[m,n,v,0]$. Other trivial reductions include $\mathsf{AMQ}[m,n,v,r+1] \leq \mathsf{AMQ}[m,n,v,r]$; $\mathsf{AMQ}[m,n,v,r] \leq \mathsf{MQ}[m,n]$; $\mathsf{AMQ}[m,n,v,r] \leq \mathsf{AMQ}[m+1,n,v,r]$; and $\mathsf{AMQ}[m,n,v,r] \leq \mathsf{AMQ}[m,n,v+1,r]$.

Unfortunately, we know of no reduction showing that AMQ with $r \geq 1$ is at least as hard as another hard problem. Nevertheless, we argue that it is a hard problem by detailing three algorithms to solve it, each with an exponential running time, assuming $v \gg m \gg r$.

Exhaustive search. Modelling $\mathcal{P}$ as a random function, we have that a random choice of $\mathbf{x}_i$ will lie in a subspace of dimension r with probability $1/q^{(m-r)}$. The first r vectors $\mathbf{x}_1, \ldots, \mathbf{x}_r$ can be chosen at random and the next $v-r$ vectors should be chosen such that $\forall i \in \{r+1, \ldots, v\} \,.\, \mathcal{P}(\mathbf{x}_i) - \mathbf{y}_i \in \langle \{\mathcal{P}(\mathbf{x}_j) - \mathbf{y}_j\}_{j=1}^{r} \rangle$. This strategy takes about $O(q^{m-r})$ time.

Grover. A large asymptotical work factor can be saved by running the algorithm on a quantum computer, employing Grover's algorithm [13] or its generalization by the name of amplitude amplification [6]. The probability of $\mathcal{P}(\mathbf{x}_i) - \mathbf{y}_i$ lying in a targeted space of dimension r is still $1/q^{(m-r)}$ but a quantum search will find one in roughly $O(q^{(m-r)/2})$ steps.

Algebraic. This strategy attempts to repeatedly find one extra vector $\mathbf{x}_{r+i}$ by running a Gröbner basis algorithm such as F4 [11] or XL [9], or a hybrid approach [4,5]. Introduce r new indeterminates z_j for $j \in \{1, \ldots, r\}$ and in addition to the n variables $\mathbf{x}_{r+i}$. Then require that $z_1(\mathcal{P}(\mathbf{x}_1) - \mathbf{y}_1) + \ldots + z_r(\mathcal{P}(\mathbf{x}_r) - \mathbf{y}_r) + \mathcal{P}(\mathbf{x}_{r+i}) - \mathbf{y}_{r+i} = 0$. After applying a linear transformation, this is equivalent to an instance of the MQ Problem with n variables and $m-r$ equations, *i.e.*, $\mathsf{AMQ}[m,n,v,r] \leq \mathsf{MQ}[m-r,n]$. A similar search for $\mathbf{x}_1, \ldots, \mathbf{x}_v$ *simultaneously* will lead to a *cubic* system with a number of equations and variables that scale linearly in v.

It is clear that the AMQ problem gets easier as r approaches $\mathsf{min}(m,v)$. The algorithms above suggest that the complexity of a solution should be exponential in $m-r$, assuming v is large enough.

4 Construction

We now describe a transform that turns an MQ signature scheme ($\mathsf{ORIGINAL.KeyGen}$, $\mathsf{ORIGINAL.Sign}$, $\mathsf{ORIGINAL.Verify}$) into another one ($\mathsf{NEW.KeyGen}$, $\mathsf{NEW.Sign}$, $\mathsf{NEW.Verify}$) that has a smaller public key but larger signatures. The objective is to minimize $|\mathsf{pk}| + |s|$ (public key size plus signature size) subject to guaranteeing κ bits of security against attackers. In the following we denote by $\mathcal{P} \in (\mathbb{F}_q[\mathbf{x}])^m$ the list of polynomials of the original public key and by pk its representation as a bit string.

Only transmit a randomly chosen part of public key. New signatures consist of σ original signatures $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$ along with some information to verify them. The main idea is that it is not necessary to transmit the entire public key for this verification. Instead, it suffices to include a small number of randomly chosen linear combinations of polynomials of $\mathcal{P}$ in each signature. So besides the

σ original signatures, part of the new signature consists of a list of α quadratic polynomials $\mathcal{R}(\mathbf{x}) : \mathbb{F}_q^n \to \mathbb{F}_q^\alpha$ derived from the original public key as $\mathcal{R}(\mathbf{x}) = \mathbf{t}\mathcal{P}(\mathbf{x})$, where $\mathbf{t} \in \mathbb{F}_q^{\alpha \times m}$ is a randomly chosen matrix.

At the time of verifying the new signature, $\mathcal{P}$ might be unknown. Nevertheless, $\mathcal{R}(\mathbf{x})$ is known so it can be used instead to obtain some level of assurance of the signatures' validity. In particular, if $\mathbf{s}$ is a valid signature for document d, *i.e.*, $\mathcal{P}(\mathbf{s}) = \mathsf{H}_1(d)$, then the same holds after multiplication by $\mathbf{t}$, *i.e.*, $\mathcal{R}(\mathbf{s}) = \mathbf{t}\mathcal{P}(\mathbf{s}) = \mathbf{t}\mathsf{H}_1(d)$. The matrix $\mathbf{t}$ is chosen *after* the σ signatures $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$ are fixed by passing them through a hash function $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^\kappa$ whose output is lifted to the space of $\alpha \times m$ matrices: $\mathbf{t} \leftarrow \mathsf{H}_2(d\|\mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma)$. This delayed choice strategy forces the signer to produce signatures honestly, because any invalid signature has probability $1/q^\alpha$ of passing this test. This probability can be made negligible by choosing the parameter α sufficiently large.

Alternatively, one can keep α by increasing the number σ of original signatures $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$ on documents deterministically derived from d as $\mathbf{s}_i = \mathsf{Sign}(\mathsf{sk}, d\|i)$ for $i \in \{1, \ldots, \sigma\}$. The probability that a set of signatures $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$ all satisfy $\mathcal{R}(s_i) = \mathbf{t}\mathsf{H}_1(d||i)$ for a randomly chosen $\mathbf{t}$ is then $1/q^{\alpha D}$, where $D \leq \sigma$ is the dimension of the subspace spanned by the errors $\mathcal{P}(s_i) - \mathsf{H}(d||i)$. We then have to rely on the hardness of the AMQ problem, because it should be infeasible to forge the signatures $\mathbf{s}_1, \cdots, \mathbf{s}_\sigma$ such that D is small.

Assure the validity of $\mathcal{R}(\mathbf{x})$. Using $\mathcal{R}(\mathbf{x})$ instead of $\mathcal{P}(\mathbf{x})$ introduces a new attack strategy: to forge signatures for a polynomial system $\mathcal{R}(\mathbf{x})$ that is not at all related to $\mathcal{P}(\mathbf{x})$. To block this attack, we need to add some information to the signature such that the verifier can check that $\mathcal{R}(\mathbf{x}) = \mathbf{t}\mathcal{P}(\mathbf{x})$. An obvious way to do this is committing to $\mathcal{P}(\mathbf{x})$ in the public key and revealing it in the signature, but this would lead to a huge signature and defeat the purpose of our construction. Instead, we compute MAC (message authentication code) polynomials to authenticate $\mathcal{R}(\mathbf{x})$.

Fix any ordering of monomials and consider the list of $N = n(n+1)/2 + n + 1$ coefficients of $p_i(\mathbf{x})$, the ith polynomial of the original public key. Group these elements into $\lceil N/k \rceil$ adjacent tuples of k elements each, padding with zeros if necessary. Interpret these k-tuples as coefficients in $\mathbb{F}_{q^k}$ of a polynomial $\hat{p}_i(z) \in \mathbb{F}_{q^k}[z]$. Let $\hat{\mathcal{P}}(z)$ denote the vector of these MAC polynomials: $\hat{\mathcal{P}}(z) = (\hat{p}_i(z))_{i=0}^{m-1}$. Apply the same operation to the α polynomials of $\mathcal{R}(\mathbf{x})$ to obtain $\hat{\mathcal{R}}(z) \in (\mathbb{F}_{q^k}[z])^\alpha$. The following diagram commutes:

$$\begin{array}{ccc} \mathcal{P}(\mathbf{x}) & \xrightarrow{\quad\mathsf{MAC}\quad} & \hat{\mathcal{P}}(z) \\ \Big\downarrow \mathbf{t} & & \Big\downarrow \mathbf{t} \\ \mathbf{t}\mathcal{P}(\mathbf{x}) & \xrightarrow{\quad\mathsf{MAC}\quad} & \mathbf{t}\hat{\mathcal{P}}(z) \end{array}$$

In other words, we have that $\mathsf{MAC}(\mathbf{t}\mathcal{P}(\mathbf{x})) = \mathbf{t}\mathsf{MAC}(\mathcal{P}(\mathbf{x}))$. The public key represents a commitment to the evaluation of $\hat{\mathcal{P}}(z)$ in a large number τ of points $r \in Z \subset \mathbb{F}_{q^k}$. A signature reveals the evaluation of $\hat{\mathcal{P}}$ in a small number ϑ of randomly chosen points $r \in O \subset Z$, and the verifier can check for all $r \in O$ whether $\hat{\mathcal{R}}(r) = \mathbf{t}\hat{\mathcal{P}}(r)$. Since $\hat{\mathcal{R}}(z) - \mathbf{t}\hat{\mathcal{P}}(z)$ are α polynomials of degree at most $\lceil N/k \rceil - 1$ there are at most $\lceil N/k \rceil - 1$ values of $r \in \mathbb{F}_{q^k}$ for which this equality holds when $\hat{\mathcal{R}}(z) \neq \mathbf{t}\hat{\mathcal{P}}(z)$. Therefore, if the equality holds for enough randomly chosen values $r \in O$, this assures the verifier that $\mathcal{R}(\mathbf{x}) = \mathbf{t}\mathcal{P}(\mathbf{x})$. Exactly which evaluations are revealed is determined by the hash value of $d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma\|\mathcal{R}(\mathbf{x})$, *i.e.*, $O = \mathsf{H}_3(d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma\|\mathcal{R}(\mathbf{x}))$. For an incorrect $\mathcal{R}(\mathbf{x})$ at most $\lceil N/k \rceil - 1$ values of $r \in \mathbb{F}_{q^k}$ can satisfy $\hat{\mathcal{R}}(r) = \mathbf{t}\hat{\mathcal{P}}(r)$. Therefore the probability that an incorrect $\mathcal{R}(\mathbf{x})$ passes the tests is bounded above by $\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^{\vartheta}$. The parameters τ and ϑ have to be chosen so that this probability is negligible.

To save space, put the $\tau = \#Z$ evaluations of $\hat{\mathcal{P}}(z)$ as leaves into a Merkle tree. The public key is the root of this Merkle tree: it commits to all evaluations of $\hat{\mathcal{P}}(z)$. Revealing a single evaluation of $\hat{\mathcal{P}}(z)$ requires $(\mathsf{log}_2\tau) - 2$ hash values to trace and verify the path from the given point to the root.

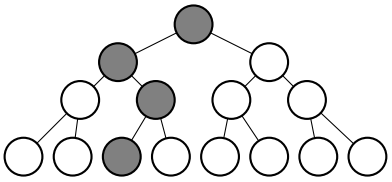

Fig. 3. Merkle tree with one opened path. The length of the path is logarithmic in the number of leafs.

We do not cover the exact implementation of the Merkle tree and instead invoke the following procedures abstractly. Merkle.generate_tree takes a list of 2^k (for some $k \in \mathbb{N}$) objects and generates the entire Merkle tree from them, returning the tree as its output. Merkle.root takes the tree and outputs its root node. Merkle.open_path takes the tree and leaf node and outputs that leaf along with all hashes needed to merge branches and travel to the root. Merkle.verify takes a root node and a path and verifies that both belong to the same tree.

While our transformation borrows the Merkle tree construction from hash-based signature schemes, we would like to stress that this is for compression only. In particular, reusing the same Merkle path for different signature poses no security threat as the transformed signature scheme is still stateless.

4.1 Summary

Figures 4, 5, 6 present pseudocode for the new key generation, signature generation, and signature verification algorithms. Aside from the standard parameter

names for MQ cryptosystems, we also rely on the following parameters or shorthand forms:

N — number of columns of the Macaulay matrix of $\mathcal{P}$, equal to $n(n+3)/2+1$.
σ — number of original signatures to include in the new signature;
α — number of polynomials to include in the new signature;
τ — number of MAC values; must be a power of two and $\tau \geq N$ must hold;
k — degree of the extension field $\mathbb{F}_{q^k}$ such that $\#\mathbb{F}_{q^k} \geq \tau$;
ϑ — number of Merkle paths to open; ϑ must be greater than 1;
Z — A subset of $\mathbb{F}_{q^k}$ that contains τ elements.

algorithm NEW.KeyGen
input: 1^κ — the security parameter in unary representation
output: sk' — new secret key
pk' – new public key

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{ORIGINAL.KeyGen}(1^\kappa)$
2: **for** i **from** 0 **to** $m-1$ **do:** ▷ obtain MAC polynomials $\hat{\mathcal{P}}(z)$
3: **for** j **from** 0 **to** $\lceil N/k \rceil - 1$ **do:**
4: $c_j \leftarrow \mathsf{cast\ coeffs}(p_i(\mathbf{x}))[jk : (j+1)k-1]$ to $\mathbb{F}_{q^k}$
5: **end**
6: $p_i(z) \leftarrow \sum_{j=0}^{\lceil N/k \rceil - 1} c_j z^j$
7: **end**
8: $\hat{\mathcal{P}}(z) \leftarrow (\hat{p}_i(z))_{i=0}^{m-1}$
9: $\mathsf{mt} \leftarrow \mathsf{Merkle.generate\_tree}(\{\hat{\mathcal{P}}(r)\}_{r \in Z})$ ▷ evaluate $\hat{\mathcal{P}}(z)$ in Z and Merkleize
10: $\mathsf{pk}' \leftarrow \mathsf{Merkle.root}(\mathsf{mt})$
11: $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathcal{P}(\mathbf{x}), \mathsf{mt})$

Fig. 4. New key generation algorithm.

5 Security

Security of the construction for large enough α is shown through a sequence of games reduction [26], going from an adversary winning the EUF-CMA game of the new scheme to one that wins the same game but associated with the original MQ signature scheme. Our reduction works in two steps. The intermediate game is also an EUF-CMA game but against a hybrid scheme defined as follows:

- HYBRID.KeyGen is identical to ORIGINAL.KeyGen. No MACs are generated.
- HYBRID.Sign retains steps 1–6 from NEW.Sign and drops the opened paths from the signature in step 12.
- HYBRID.Verify retains steps 1–8 from NEW.Verify and instead of verifying the MAC (steps 9–19) verifies that $\mathcal{R}(\mathbf{x}) = \mathbf{t}^\mathsf{T}\mathcal{P}(\mathbf{x})$ because the public key $\mathcal{P}(\mathbf{x})$ is now known and there is no longer any need to rely on the MACs.

algorithm NEW.Sign
input: sk' — secret key
$d \in \{0,1\}^*$ — document to be signed
output: s' — signature for d

1: $\mathsf{sk}, \mathcal{P}(\mathbf{x}), \mathsf{mt} \leftarrow \mathsf{sk}'$
2: **for** i **from** 1 **to** σ **do:** ▷ generate σ original signatures
3: $s_i \leftarrow \mathsf{ORIGINAL.Sign}(\mathsf{sk}, d\|i)$
4: **end**
5: $\mathbf{t} \leftarrow \mathsf{H}_2(d\|s_1\|\cdots\|s_\sigma)$
6: $\mathcal{R}(\mathbf{x}) \leftarrow \mathbf{t}^\mathsf{T}\mathcal{P}(\mathbf{x})$ ▷ get verification polynomials $\mathcal{R}(\mathbf{x})$
7: $O \leftarrow \mathsf{H}_3(d||s_1||\cdots||s_\sigma||\mathcal{R}(\mathbf{x}))$ such that $O \subset Z$ and $\#O = \vartheta$
8: open paths $\leftarrow$ empty_list
9: **for** r **in** O **do:** ▷ open indicated Merkle paths for MACs
10: open paths.append(Merkle.open_path(mt, $\hat{\mathcal{P}}(r)$))
11: **end**
12: $s' \leftarrow (s_1, \ldots, s_\sigma, \mathcal{R}(\mathbf{x}), \mathsf{open\ paths})$

Fig. 5. New signature generation algorithm.

Theorem 1. *If there is an adversary A against EUF-CMA-NEW in time T with Q random oracle queries and with success probability ϵ, then there is an adversary B^A that wins EUF-CMA-HYB in time $O(T)$ and with success probability at least* $\epsilon - (Q+1)\left(\frac{\lceil N/k\rceil - 1}{\tau}\right)^{\vartheta} - 2\tau(Q+1)/2^{\kappa}$.

Theorem 2. *If there is an adversary A against EUF-CMA-HYB in time T with Q random oracle queries and with success probability ϵ then there exists an adversary B^A against EUF-CMA-ORIGINAL in time $O(T)$ with success probability at least* $\epsilon - (Q+1)\left(\frac{1}{q}\right)^{\alpha}$.

Due to the space constraint, we defer the proofs to Appendix A. In both cases the simulated algorithm has unbridled access to the real random oracle; the simulator does not measure nor compute on queries or responses. The set of challenge-response pairs of the random oracle is random but fixed before the protocol starts. While the simulator algorithm works in the quantum random oracle model, the proof of the lower bound on the success probability models the queries as classical messages and therefore only holds in the classical random oracle model. Nevertheless, we believe that the success probability can still be proven to be significant in the quantum random oracle model.

Theorems 1 and 2 give a tight reduction of EUF-CMA-NEW to EUF-CMA-ORIGINAL if we select parameters such that $((\lceil N/k\rceil - 1)/\tau)^{\vartheta} < 2^{-\kappa}$ and $q^{-\alpha} < 2^{-\kappa}$ and the resulting scheme will be provably as secure as the underlying MQ signature scheme. If we choose α to be smaller and compensate with a larger σ then the conditions on τ and ϑ are identical but Thm. 2 fails to produce a winning adversary. Instead we must rely on the hardness of AMQ for small r.

algorithm NEW.Verify
input: pk' — public key
$d \in \{0,1\}^*$ — document
s' — signature on document
output: True or False

$s_1, \ldots, s_\sigma, \mathcal{R}(\mathbf{x}), \mathsf{open\ paths} \leftarrow s'$
$\mathbf{t} \leftarrow \mathsf{H}_2(d \| s_1 \| \ldots \| s_\sigma)$
for i **from** 1 **to** σ **do:** ▷ verify original signatures against $\mathcal{R}(\mathbf{x})$
  $\mathbf{s}_i \leftarrow \mathsf{cast}\ s_i\ \mathsf{to}\ \mathbb{F}_q^n$
  if $\mathcal{R}(\mathbf{s}_i) \neq \mathbf{t}\mathsf{H}_1(d \| i)$
    return False
  end
end
for j **from** 1 **to** α **do:** ▷ obtain MAC polynomials $\hat{\mathcal{R}}(z)$
  for i **from** 0 **to** $\lceil N/k \rceil - 1$ **do:**
    $a_i \leftarrow \mathsf{cast}\ \mathsf{coeffs}(r_j(\mathbf{x}))[ik : (i+1)k - 1]\ \mathsf{to}\ \mathbb{F}_{q^k}$
  end
  $\hat{r}_j(z) \leftarrow \sum_{i=0}^{\lceil N/k \rceil - 1} a_i z^i$
end
$\hat{\mathcal{R}}(z) \leftarrow (\hat{r}_j(z))_{j=1}^{\alpha}$
$O \leftarrow \mathsf{H}_3(d || s_1 || \cdots || s_\sigma || \mathcal{R}(\mathbf{x}))$ such that $O \subset Z$ and $\#O = \vartheta$
for i **from** 1 **to** $\#O$ **do:** ▷ validate $\hat{\mathcal{R}}(z)$ against opened Merkle paths
  $\hat{\mathcal{P}}(r), \mathsf{mp} \leftarrow \mathsf{open\ paths}[i]$
  if $\mathsf{Merkle.verify}(\mathsf{pk}', \mathsf{mp}) = \mathsf{False}$ **or** $\hat{\mathcal{R}}(O[i]) \neq \mathbf{t}^\mathsf{T}\hat{\mathcal{P}}(r)$
    return False
  end
end
return True

Fig. 6. New signature verification algorithm.

If the adversary wishes to produce a forgery then he must query H_2 on $d\|\mathbf{s}_1\| \cdots \|\mathsf{s}_\sigma$ and obtain a *suitable* $\mathbf{t} \in \mathbb{F}_q^{\alpha \times m}$, where suitable means $\forall j \,.\, \mathbf{t}\mathcal{P}(\mathbf{s}_j) = \mathbf{t}\mathsf{H}_1(d\|j)$ even though, potentially, $\exists j \,.\, \mathcal{P}(\mathbf{s}_j) \neq \mathsf{H}_1(d\|j)$. The probability of obtaining a suitable $\mathbf{t}$ is $q^{-\alpha r}$, where $r = \mathsf{dim}\langle\{\ldots, \mathcal{P}(\mathbf{s}_j) - \mathsf{H}_1(d\|j), \ldots\}\rangle$. So if the adversary can generate small-dimension AMQ solutions, then he can generate forgeries.

We have tried to find a formal reduction-based proof showing that the hardness of AMQ (for small r) is sufficient in addition to necessary. However, this task seems very non-trivial. Therefore, we leave the exact security of this parameter choice as an open problem.

6 Discussion

The public key contains only the Merkle root, and hence $|\mathsf{pk}| = \kappa$. By contrast, a signature contains σ original-scheme signatures $\mathbf{s}_1, \ldots, \mathbf{s}_\sigma$; α quadratic polynomials $\mathcal{R}(\mathbf{x})$; and ϑ Merkle paths of depth $\mathsf{log}_2\tau$ ending in an element of $\mathbb{F}_{q^k}^m$. Consequently, $|s| = (\sigma n + \alpha N + \vartheta mk)\lceil \mathsf{log}_2 q \rceil + \vartheta(\mathsf{log}_2\,\tau - 2)\kappa$.

Two constraints should be satisfied in order to guarantee at least κ bits of security. First, the MAC should be unforgeable: $\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^\vartheta \leq 2^{-\kappa}$. A larger τ is slower but generates smaller signatures. Second, forging approximate signatures should be hard. The case $\sigma = 1$ requires that $\alpha \geq \lceil \frac{\kappa}{\mathsf{log}_2\, q} \rceil$ and is provably secure. In contrast, smaller α does not lead to a concrete security estimate even if the AMQ Problem is hard. In this case we need at least $\alpha\sigma \geq \lceil \frac{\kappa}{\mathsf{log}_2\, q} \rceil$ to be safe against a trivial brute force attack. Table 1 presents several viable parameter choices and compares the schemes before and after transformation.

In the case of UOV, our technique is perfectly compatible with the compression technique of Petzoldt *et al.* [22], where the first $v(v+1)/2 + ov$ columns of the Macaulay matrix are generated from a pseudorandom generator and a short seed. The same number of coefficients can be dropped from $\mathcal{R}(\mathbf{x})$, and the smaller degree of $\hat{\mathcal{R}}(z)$ requires fewer opened Merkle paths. This combination shrinks signatures even more while only increasing the public key by κ bits.

Table 1. Comparison of public key and signature size of HFEv^- and UOV (with compression) before and after applying our transformation. The recommended parameters were drawn from the Gui signature scheme [21] and Petzoldt's dissertation [20]. In all cases, $\tau = 2^{20}$.

scheme	parameters	sec. lvl.	$\lvert\mathsf{pk}\rvert$	$\lvert s\rvert$
original HFEv^-	$q = 2, n = 98, m = 90$	80	56.8 kB	98 bits
transformed	$\alpha = 1, \sigma = 80, k = 21, \vartheta = 7$	?	80 bits	4.4 kB
original HFEv^-	$q = 2, n = 133, m = 123$	120	139.2 kB	123 bits
transformed	$\alpha = 1, \sigma = 120, k = 21, \vartheta = 11$	?	120 bits	9.4 kB
UOVrand	$q = 256, n = 135, m = 45$	128	45.5 kB	1080
transformed	$\alpha = 16, \sigma = 1, k = 3, \vartheta = 12$	128	256 bits	21.3 kB
UOVrand	$q = 256, n = 210, m = 70$	192	169.9 kB	1 680 bits
transformed	$\alpha = 24, \sigma = 1, k = 3, \vartheta = 19$	192	384 bits	70.4 kB
UOVrand	$q = 256, n = 285, m = 95$	256	423.0 kB	2 280 bits
transformed	$\alpha = 32, \sigma = 1, k = 3, \vartheta = 28$	256	512 bits	166.3 kB

The shrinkage is the most striking for $\sigma > 1$, in which case α can be small. However, this requires the AMQ Problem to be hard and offers no provable security. There is another possibility: security takes no hit when α is kept reasonably small and we choose a larger q instead. Unfortunately, not all MQ signature schemes can be adapted as-is to a larger field.

We close with a note on the flexibility of our construction. As we presented the transform, the entire public key is replaced by a single Merkle root. However,

some applications prefer to minimize the signature size while having a fixed and insufficient allowance for the public key. In this scenario, one can apply the Merkle tree MAC construction to only the second half of the public key's Macaulay matrix, and present the other half with the Merkle root as the new public key. Similarly, it is somewhat redundant to reduce the public key to a single Merkle root if both its children are released in nearly every signature. It is better to compute 2^δ separate trees of height $\log_2 \tau - \delta$ and shrink the signatures a little by tracing a shorter path for each MAC, at the expense of a factor-2^δ larger public key.

Acknowledgements. The authors would like to thank the reviewers for their helpful feedback. 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 ICT programme under contract FP7-ICT-2013-10-SEP-210076296 PRACTICE, through the Horizon 2020 research and innovation programme under grant agreement No H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-645622 PQCRYPTO. Alan Szepieniec is being supported by a doctoral grant from the Flemish Agency for Innovation and Entrepreneurship (VLAIO, formerly IWT).

References

1. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A New Hope. In: Holz, T., Savage, S. (eds.) 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016. pp. 327–343. USENIX Association (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim
2. Bernstein, D.J., Buchmann, J., Ding, J., Goubin, L., Lange, T., Nguyen, P., Okamoto, T., Salvail, L., Silverberg, A., Silverman, J., Stam, M., Wolf, C. (eds.): International Workshop on Post-Quantum Cryptography, PQCrypto 2006, Leuven, Belgium, May 23-26, 2006, Proceedings (2006), http://postquantum.cr.yp.to/
3. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 368–397. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46800-5_15
4. Bettale, L., Faugère, J., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. J. Mathematical Cryptology 3(3), 177–197 (2009), http://dx.doi.org/10.1515/JMC.2009.009
5. Bettale, L., Faugère, J., Perret, L.: Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: van der Hoeven, J., van Hoeij, M. (eds.) International Symposium on Symbolic and Algebraic Computation, ISSAC'12, Grenoble, France - July 22 - 25, 2012. pp. 67–74. ACM (2012), http://doi.acm.org/10.1145/2442829.2442843

6. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation, 2000. arXiv preprint quant-ph/0005055 https://arxiv.org/abs/quant-ph/0005055
7. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass MQ-based identification to mq-based signatures. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology —- ASIACRYPT 2016 — 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10032, pp. 135–165 (2016), http://dx.doi.org/10.1007/978-3-662-53890-6_5
8. Courtois, N., Finiasz, M., Sendrier, N.: How to achieve a mceliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2248, pp. 157–174. Springer (2001), http://dx.doi.org/10.1007/3-540-45682-1_10
9. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 392–407. Springer (2000), http://dx.doi.org/10.1007/3-540-45539-6_27
10. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal Gaussians. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 40–56. Springer (2013), http://dx.doi.org/10.1007/978-3-642-40041-4_3
11. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F 4). Journal of pure and applied algebra 139(1), 61–88 (1999)
12. 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), http://dx.doi.org/10.1137/0217017
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996), http://doi.acm.org/10.1145/237814.237866
14. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 206–222. Springer (1999), http://dx.doi.org/10.1007/3-540-48910-X_15
15. Matt Braithwaite, Google: Experimenting with post-quantum cryptography (2016), https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html
16. Maurer, U.M. (ed.): Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding, Lecture Notes in Computer Science, vol. 1070. Springer (1996)

17. http://csrc.nist.gov/groups/ST/toolkit/
18. National Institute for Standards and Technology (NIST): Post-quantum crypto standardization (2016), http://csrc.nist.gov/groups/ST/post-quantum-crypto/
19. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer [16], pp. 33–48, http://dx.doi.org/10.1007/3-540-68339-9_4
20. Petzoldt, A.: Selecting and reducing key sizes for multivariate cryptography (July 2013), http://tuprints.ulb.tu-darmstadt.de/3523/
21. Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for HFEv-based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9452, pp. 311–334. Springer (2015), http://dx.doi.org/10.1007/978-3-662-48797-6_14
22. Petzoldt, A., Thomae, E., Bulygin, S., Wolf, C.: Small public keys and fast verification for multivariate quadratic public key systems. In: Preneel, B., Takagi, T. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6917, pp. 475–490. Springer (2011), http://dx.doi.org/10.1007/978-3-642-23951-9_31
23. Rogaway, P. (ed.): Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings, Lecture Notes in Computer Science, vol. 6841. Springer (2011), http://dx.doi.org/10.1007/978-3-642-22792-9
24. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-key identification schemes based on multivariate quadratic polynomials. In: Rogaway [23], pp. 706–723, http://dx.doi.org/10.1007/978-3-642-22792-9_40
25. Shor, P.W.: Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In: Adleman, L.M., Huang, M.A. (eds.) Algorithmic Number Theory, First International Symposium, ANTS-I, Ithaca, NY, USA, May 6-9, 1994, Proceedings. Lecture Notes in Computer Science, vol. 877, p. 289. Springer (1994), http://dx.doi.org/10.1007/3-540-58691-1_68
26. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004), http://eprint.iacr.org/2004/332
27. Stern, J.: A new identification scheme based on syndrome decoding. In: Stinson, D.R. (ed.) Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Lecture Notes in Computer Science, vol. 773, pp. 13–21. Springer (1993), http://dx.doi.org/10.1007/3-540-48329-2_2
28. PQCRYPTO ICT-645622 (2015), http://pqcrypto.eu.org/

A Proofs

A.1 Proof of Theorem 1

Theorem 1. If there is an adversary A against EUF-CMA-NEW in time T with Q random oracle queries and with success probability ϵ, then there is an

adversary B^A that wins EUF-CMA-HYB in time $O(T)$ and success probability at least $\epsilon - (Q+1)\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^{\vartheta} - 2\tau(Q+1)/2^{\kappa}$.

Proof. Firstly, we describe how the adversary B^A plays the EUF-CMA-HYB game. We denote by C the challenger for the EUF-CMA-HYB game. The EUF-CMA-HYB game begins when our adversary B^A receives the public key $\mathsf{pk} = \mathcal{P}(\mathbf{x})$ from the challenger C. Upon receiving this message, the hybrid adversary B^A runs steps 5–12 of $\mathsf{NEW.KeyGen}$ to produce a new public key pk' and Merkle tree mt. The public key pk' is sent to the EUF-CMA-NEW adversary A.

Whenever A requests a message $d_i \in \{0,1\}^*$ be signed, B^A requests C to sign the message d_i. Then, C responds with the signature $s_i = (\mathbf{s}_1, \ldots, \mathbf{s}_\sigma, \mathcal{R}(\mathbf{x}))$. At this point A^B runs steps 7–12 of $\mathsf{NEW.Sign}$ to compute O and open the associated Merkle paths necessary to complete the signature, which he then sends to A. After making some message-queries, A terminates his end of the protocol by producing a message-signature pair (d, s). The adversary B^A simply drops the Merkle paths from the signature s to get a signature s' for the hybrid signature scheme and sends the message-signature (d, s') on to the challenger C.

It is clear that B^A runs with overhead linear in the number of signing queries done by A, so the overhead is $O(T)$. We show that B^A wins the EUF-CMA-HYB game with probability at least $\epsilon - (Q+1)\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^{\vartheta} - 2\tau(Q+1)/2^{\kappa}$, where Q is the number of random oracle queries made by A.

Our adversary B^A wins the EUF-CMA-HYB game if the message-signature pair (d, s') it outputs is a valid signature for the hybrid signature scheme and if B^A has not queried C to sign d before. This is the case if the message-signature pair (d, s) output by A wins the EUF-CMA-NEW game and the polynomial map included in s is correct, meaning that if $s = (\mathbf{s}_1, \cdots, \mathsf{s}_\sigma, \mathcal{R}(x))$ and $\mathbf{t} = \mathsf{H}_2(d \| \mathbf{s}_1 \| \cdots \| \mathbf{s}_\sigma)$, then $\mathcal{R}(x) = \mathbf{t}\mathcal{P}(x)$.

By assumption the first event occurs with probability ϵ. We finish the proof by showing that that the probability that the first event occurs, but the second event fails is bounded by $2\tau(Q+1)/2^{\kappa} + (Q+1)\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^{\vartheta}$.

Assume the message-signature pair (d, s) that is output by A wins the EUF-CMA-NEW game. First consider the case where for one of the $r \in O$ the leaf of the merkle paths corresponding to r in s is not equal to $\hat{\mathcal{P}}(r)$. Since s is a valid signature for the new signature scheme this means that A has forged a valid merkle path that ends in the merkle root. This requires finding a second preimage to one of the $2\tau - 1$ values in the Merkle tree. The probability that any algorithm does that is bounded by $2\tau(Q+1)/2^{\kappa}$, so in the rest of the proof we can assume that all the leaves included in the signature are valid.

Without loss of generality we can assume that A only outputs a message-signature pair (d, s) with $s = (\mathbf{s}_1, \cdots, \mathbf{s}_\sigma, \mathcal{R}(\mathbf{x}), \mathsf{open\ paths})$ after having queried the random oracle H_3 for its value at $d\|\mathbf{s}_1\| \cdots \|\mathbf{s}_\sigma\|\mathcal{R}(\mathbf{x})$. Indeed, if this is not the case A can be transformed into an adversary that does this at the cost of only one extra random oracle query. Let $d^{(i)}\|\mathbf{s}_1^{(i)}\| \cdots \|\mathbf{s}_1^{(i)}\|\mathcal{R}^{(i)}$ for i running from 1 to $Q+1$ be all the values of this form that A has queried H_3 for. Then A can only output a message-signature pair which is a valid pair and such that $\mathcal{R}(\mathbf{x}) \neq \mathbf{t}\mathcal{P}(\mathbf{x})$ if this is true for one of the message pairs $(d^{(i)}, s^{(i)})$. But for any message-signature pair (d, s) the probability that for a randomly chosen elements $r \in Z$ we have that $\hat{\mathcal{R}}(r) = \mathbf{t}\hat{\mathcal{P}}(r)$ is either 1 in the case that $\mathcal{R}(\mathbf{x}) = \mathbf{t}\mathcal{P}(\mathbf{x})$, or bounded by $\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)$ otherwise. This is so because if $\mathcal{R}(\mathbf{x}) \neq \mathbf{t}\mathcal{P}(\mathbf{x})$, then $\hat{\mathcal{R}}(z) \neq \mathbf{t}\hat{\mathcal{P}}(z)$, so the list of polynomials $\hat{\mathcal{R}}(z) - \mathbf{t}\hat{\mathcal{P}}(z)$ contains at least one nonzero polynomial of degree at most $\lceil N/k \rceil - 1$, so it has at most $\lceil N/k \rceil - 1$ zeros in Z. The probability that a zero is chosen randomly from Z is therefore at most $\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)$.

Since ϑ elements of Z are chosen randomly and independently by the random oracle H_3 the probability that a message-signature pair (d, s) for which $\mathcal{R}(\mathbf{x}) \neq t\mathcal{P}(\mathbf{x})$ is valid for the new signature scheme is at most $\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^\vartheta$. The union bound implies that the probability that for any $1 \leq i \leq Q+1$ the message-signature pair $(d^{(i)}, s^{(i)})$ is valid and $\mathcal{R}^{(i)}(\mathbf{x}) \neq \mathbf{t}^{(i)}\mathcal{P}(\mathbf{x})$ is bounded by $(Q+1)\left(\frac{\lceil N/k \rceil - 1}{\tau}\right)^\vartheta$. The probability that A outputs such a signature pair is also bounded by this probability. □

A.2 Proof of Theorem 2

Theorem 2. If there is an adversary A against EUF-CMA-HYB in time T with Q random oracle queries and with success probability ϵ then there exists an adversary B^A against EUF-CMA-ORIGINAL in time $O(T)$ with success probability at least $\epsilon - (Q+1)\left(\frac{1}{q}\right)^\alpha$.

Proof. Firstly, we describe how the adversary B^A plays the EUF-CMA-ORIGINAL game. We denote by C the challenger for the EUF-CMA-ORIGINAL game. The EUF-CMA-ORIGINAL game begins when the challenger C sends the public key $\mathsf{pk} = \mathcal{P}(\mathbf{x})$ to B^A. This is also the public key under the hybrid scheme, so B^A sends it to the adversary A to initiate the EUF-CMA-HYB game.

Whenever the adversary A requests a message $d_i \in \{0,1\}^*$ be signed, B^A requests C to sign the messages $d_i\|1, \ldots, d_i\|\sigma$. Then, C responds with signatures $\mathbf{s}_i^{(1)}, \ldots, \mathbf{s}_i^{(\sigma)}$. Using this set of σ original scheme signatures, the B^A runs steps 5 and 6 of $\mathsf{NEW.Sign}$ to compute $\mathcal{R}(\mathbf{x})$. He then sends $s_i = (\mathbf{s}_i^{(1)}, \ldots, \mathbf{s}_i^{(\sigma)}, \mathcal{R}(\mathbf{x}))$ to A.

After making some message queries A terminates the protocol by producing a pair (d, s). Let $s = (\mathbf{s}_1, \cdots, \mathbf{s}_\sigma, \mathcal{R}(x))$, then B^A sends the message-signature pair $(d\|1, \mathbf{s}_1)$ to C. (Any one of the pairs $(d\|i, \mathbf{s}_i)$ would do the job.)

It is clear that B^A runs with overhead linear in the number of signing queries done by A, so the overhead is $O(T)$. We show that B^A wins the EUF-CMA-ORIGINAL game with probability at least $\epsilon - (Q+1)q^{-\alpha}$, where Q is the number of random oracle queries made by A.

Our adversary B^A wins the EUF-CMA-HYB game if the message-signature pair (d, s') it outputs is a valid signature for the hybrid signature scheme and if B^A has not queried C to sign d before. This is the case if the message-signature pair (d, s) outputted by A wins the EUF-CMA-HYB game and $\mathcal{P}(s_1) = \mathsf{H}_1(d||1)$.

By assumption the first event occurs with probability ϵ. We finish the proof by showing that that the probability that the first event occurs, but the second event fails is bounded by $(Q+1)q^{-\alpha}$.

Without loss of generality we can assume that A only outputs a message-signature pair (d, s) with $s = (\mathbf{s}_1, \cdots, \mathbf{s}_\sigma, \mathcal{R}(\mathbf{x}))$ after having queried the random oracle H_2 for its value at $d\|\mathbf{s}_1\|\cdots\|\mathbf{s}_\sigma$. Indeed, if this is not the case A can be transformed into an adversary that does this at the cost of only one extra random oracle query. Let $d^{(i)}\|\mathbf{s}_1^{(i)}\|\cdots\|\mathbf{s}_1^{(i)}$ for i running from 1 to $(Q+1)$ be all the values of this form that A has queried H_2 for. Then A can only output a message-signature pair which is valid and such that $\mathcal{P}(\mathbf{s}_1) \neq \mathsf{H}_1(d\|1)$ if this is true for one of the message pairs $(d^{(i)}, s^{(i)})$. But, for any message-signature pair (d, s) the probability that for a random $\mathbf{t} \in \mathbb{F}_q^{\alpha\times m}$ we have that $\mathbf{t}\mathcal{P}(\mathbf{s}_1) = \mathbf{t}\mathsf{H}_1(d\|1)$ is either 1 in the case that $\mathcal{P}(\mathbf{s}_1) = \mathsf{H}_1(d\|1)$, or $q^{-\alpha}$ otherwise. This is so because if $\mathcal{P}(\mathbf{s}_1) \neq \mathsf{H}_1(d\|1)$, then $\mathcal{P}(\mathbf{s}_1) - \mathsf{H}_1(d\|1)$ is a nonzero vector in $\mathbb{F}_q^m$. The probability that a nonzero vector is in the kernel of a randomly chosen matrix $\mathbf{t} \in \mathbb{F}_q^{\alpha\times m}$ is exactly $q^{-\alpha}$.

Since $\mathbf{t}$ is chosen randomly by the random oracle H_2 the probability that a message-signature pair (d, s) for which $\mathcal{P}(\mathbf{s}_1) \neq \mathsf{H}_1(d\|1)$ is valid for the new signature scheme is $q^{-\alpha}$. The union bound implies that the probability that for any $1 \leq i \leq Q+1$ the message-signature pair $(d^{(i)}, s^{(i)})$ is valid and $\mathcal{P}(\mathbf{s}_1) \neq \mathsf{H}_1(d\|1)$ is bounded by $(Q+1)q^{-\alpha}$. The probability that A outputs such a signature pair is also bounded by this probability. □