SOFIA: $\mathcal{MQ}$-based signatures in the QROM

Ming-Shing Chen[1] and Andreas Hülsing[2] and Joost Rijneveld[3] and
Simona Samardjiska[3,4] and Peter Schwabe[3]

[1] Department of Electrical Engineering, National Taiwan University,
and Research Center for Information Technology Innovation, Academia Sinica,
Taipei, Taiwan
mschen@crypto.tw
[2] Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven, Eindhoven, The Netherlands
andreas@huelsing.net
[3] Digital Security Group, Radboud University, Nijmegen, The Netherlands
joost@joostrijneveld.nl, peter@cryptojedi.org
[4] Faculty of Computer Science and Engineering,
"Ss. Cyril and Methodius" University, Skopje, R. Macedonia
simona.samardjiska@finki.ukim.mk

Abstract. We propose SOFIA, the first $\mathcal{MQ}$-based signature scheme provably secure in the quantum-accessible random oracle model (QROM). Our construction relies on an extended version of Unruh's transform for 5-pass identification schemes that we describe and prove secure both in the ROM and QROM.
Based on a detailed security analysis, we provide concrete parameters for SOFIA that achieve 128 bit post-quantum security. The result is SOFIA-4-128 with parameters that are carefully optimized to minimize signature size and maximize performance. SOFIA-4-128 comes with an implementation targeting recent Intel processors with the AVX2 vector-instruction set; the implementation is fully protected against timing attacks.
Keywords: Post-quantum cryptography, multivariate cryptography, 5-pass identification schemes, QROM, Unruh's transform, vectorized implementation.

1 Introduction

At Asiacrypt 2016, Chen, Hülsing, Rijneveld, Samardjiska, and Schwabe presented a post-quantum signature scheme called MQDSS [CHR+16], obtained by applying a generalized Fiat-Shamir transform to a 5-pass identification schemes (IDS) with security based on the hardness of solving a system of multivariate quadratic equations ($\mathcal{MQ}$ problem). Unlike previous $\mathcal{MQ}$-based signature schemes, MQDSS comes with a reduction from a random instance of $\mathcal{MQ}$; it

This work was supported by the Netherlands Organization for Scientific Research (NWO) under Veni 2013 project 13114 and by the European Commission through the ICT Programme under contract ICT-645622 PQCRYPTO. Permanent ID of this document: 355ce8e96788011f95a9ea9db7793836. Date: July 7, 2017

does not need additional assumptions on the hardness of the Isomorphism-of-Polynomials (IP) [Pat96] or related problems like MinRank [Cou01,FLP08].

Unfortunately, the security reduction of MQDSS is in the random-oracle model and highly non-tight. The authors state that they see the proposal as *"a major step"* towards a scheme with *"a tight reduction to [sic] $\mathcal{MQ}$ in the quantum-random-oracle model (QROM) or even better in the standard model"*. In this paper, we make another major step towards such a scheme. More specifically, we propose SOFIA, a digital signature scheme that is provably EU-CMA secure in the QROM if the $\mathcal{MQ}$ problem is hard and allows for a tight reduction in the ROM (albeit not in the QROM).

To achieve this, we start from Unruh's transform [Unr15] for transforming sigma protocols to non-interactive zero-knowledge proofs (and signatures) in the QROM. The reason for a different transform comes from the inherent problems of the Fiat-Shamir transform (and also the generalization to 5-pass schemes) in the QROM. Namely, the proof technique introduced by Pointcheval and Stern [PS96] requires rewinding of the adversary and adaptively programming the random oracle. Not only does this cause problems in the QROM, it also produces non-tight proofs. Unruh's transform avoids these problems by adopting and tweaking an idea from Fischlin's transform [Fis05] that solves the rewinding problem.

Instead of simply applying the transform to a 3-pass IDS, we extend it such that it applies to any 5-pass IDS with binary second challenge (named $q2$-IDS in [CHR$^+$16]) and thus to the $\mathcal{MQ}$-based IDS from [SSH11]. This is in the same spirit as the generalization of the Fiat-Shamir transform in [CHR$^+$16]. We prove that the signature scheme resulting from the application of the transform is post-quantum EU-CMA secure (PQ-EU-CMA) in the QROM. This proof follows a two-step approach: We first give a (tight) proof in the ROM, and then discuss the changes necessary to carry over to the QROM. We then instantiate the construction with the 5-pass $\mathcal{MQ}$-based IDS introduced by Sakumoto, Shirai, and Hiwatari in [SSH11] and provide various optimizations particularly suited for this specific IDS. These optimizations almost halve the size of the signature compared to the non-optimized generic transform.

We instantiate SOFIA with carefully optimized parameters aiming at the 128-bit post-quantum security level; we refer to this instance as SOFIA-4-128. A comparison with MQDSS-31-64 from [CHR$^+$16], which targets the same security level, shows that the improvements in security guarantees come at a cost: with 123 KiB, SOFIA-4-128 signatures are about a factor of 3 larger than MQDSS-31-64 signatures and our optimized SOFIA-4-128 software takes about a factor of 3 longer for both signing and verification than the optimized MQDSS-31-64 software presented in [CHR$^+$16]. However, like MQDSS, SOFIA features extremely short keys; specifically, SOFIA-4-128 public keys have 64 bytes and secret keys have 32 bytes.

SOFIA is not the first concrete signature scheme with a proof in the QROM. Notably, TESLA-768 [ABBD15] is a lattice-based signature scheme with a reduction in the QROM, while Picnic-10-38 [CDG$^+$17] is the result of constructing a signature scheme from a symmetric primitive using the transform by

Unruh [Unr15] that was mentioned above. Relying on even more conservative assumptions, the hash-based signature scheme SPHINCS-256 [BHH$^+$15] has a tight proof in the standard model. Although SOFIA-4-128 remains faster than SPHINCS-256 (which is, because of its standard model assumptions, arguably the 'scheme to beat'), we do significantly exceed its 40 KiB signature size. Conversely, but on a similar note, SOFIA-4-128 outperforms Picnic-10-38 both in terms of signing speed and signature size. TESLA-768 remains the 'odd one out' with its small signatures but much larger keys; it strongly depends on context whether this is an upside or a problem. See Table 3 for a numeric overview of the comparison.

Organization of this paper. Section 2 gives the necessary background on identification schemes and signature schemes. Section 3 presents the modified Unruh transform to support $q2$ identification schemes. Section 4 revisits the 5-pass identification scheme introduced in [SSH11]. Section 5 introduces the SOFIA signature scheme and finally Section 6 explains our parameter choices for SOFIA-4-128 and gives details of our optimized implementation.

Availability of software. We place all software presented in this paper into the public domain to maximize reusability of our results. It is available for download at `https://joostrijneveld.nl/papers/sofia`.

2 Preliminaries

In the following we provide basic definitions used throughout this work. We are concerned with post-quantum security, i.e., a setting where honest parties use classical computers but adversaries might have access to a quantum computer. Therefore, we adapt some common security notions accordingly, modeling adversaries as quantum algorithms.

Digital signatures. In this work we are concerned with the construction of digital-signature schemes. These are defined as follows.

Definition 2.1 (Digital signature scheme). *A digital-signature scheme with security parameter k, denoted $\mathsf{Dss}(1^k)$ is a triplet of polynomial-time algorithms $\mathsf{Dss} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ defined as follows:*

- *The key-generation algorithm* KGen *is a probabilistic algorithm that outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$.
- *The signing algorithm* Sign *is a possibly probabilistic algorithm that on input a secret key* sk *and a message M outputs a signature σ.*
- *The verification algorithm* Vf *is a deterministic algorithm that on input a public key* pk*, a message M and a signature σ outputs a bit b, where $b = 1$ indicates that the signature is accepted and $b = 0$ indicates a reject.*

We write Dss instead of $\mathsf{Dss}(1^k)$, whenever the security parameter k is clear from context or irrelevant. For correctness of a Dss, we require that for all

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()$, all messages M and all signatures $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, M)$, we get $\mathsf{Vf}(\mathsf{pk}, M, \sigma) = 1$, i.e., that correctly generated signatures are accepted.

Existential Unforgeability under Adaptive Chosen Message Attacks. The standard security notion for digital signature schemes is existential unforgeability under adaptive chosen message attacks (EU-CMA) [GMR88] which is defined using the following experiment.

```
Experiment $\mathsf{Exp}^{\mathsf{eu\text{-}cma}}_{\mathsf{Dss}(1^k)}(\mathcal{A})$
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}()$
    $(M^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})$
    Let $\{(M_i)\}_1^{Q_s}$ be the queries to $\mathsf{Sign}(\mathsf{sk}, \cdot)$.
    Return 1 iff $\mathsf{Vf}(\mathsf{pk}, M^\star, \sigma^\star) = 1$ and $M^\star \notin \{M_i\}_1^{Q_s}$.
```

For the success probability of an adversary $\mathcal{A}$ in the above experiment we write

$$\mathrm{Succ}^{\mathsf{eu\text{-}cma}}_{\mathsf{Dss}(1^k)}(\mathcal{A}) = \mathsf{Pr}\left[\mathsf{Exp}^{\mathsf{eu\text{-}cma}}_{\mathsf{Dss}(1^k)}(\mathcal{A}) = 1\right].$$

A signature scheme is called EU-CMA-secure if any PPT algorithm $\mathcal{A}$ has only negligible success probability. Similarly, a signature scheme is called PQ-EU-CMA-secure if any quantum polynomial time adversary has only negligible success probability. We only give a formal definition for the latter.

Definition 2.2 (PQ-EU-CMA). *Let $k \in \mathbb{N}$ and Dss(1^k) a digital signature scheme with security parameter k as defined above. We call Dss(1^k) post-quantum existentially unforgeable under chosen message attacks or PQ-EU-CMA-secure if for all $Q_s, t = \mathrm{poly}(k)$ the success probability of any quantum algorithm $\mathcal{A}$ (the adversary) running in time $\leq t$, making at most Q_s queries to* Sign *in the above experiment, is negligible in k:*

$$\mathrm{Succ}^{\mathit{eu\text{-}cma}}_{\mathit{Dss}(1^k)}(\mathcal{A}) = \mathrm{negl}(k).$$

Post-quantum security against key-only attacks (PQ-KOA) is defined the same as PQ-EU-CMA but with $Q_s = 0$, i.e., the adversary is given no access to the signing oracle.

Identification Schemes. An identification scheme (IDS) is a protocol that allows a prover $\mathcal{P}$ to prove its identity to a verifier $\mathcal{V}$. More formally this is covered by the following definition.

Definition 2.3 (Identification scheme). *An identification scheme with security parameter k, denoted* $\mathsf{IDS}(1^k)$, *consists of three PPT algorithms* $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ *such that:*

- *the key generation algorithm* KGen *is a probabilistic algorithm that outputs a key pair* $(\mathsf{sk}, \mathsf{pk})$.
- *$\mathcal{P}$ and $\mathcal{V}$ are interactive algorithms, executing a common protocol. The prover $\mathcal{P}$ takes as input a secret key* sk *and the verifier $\mathcal{V}$ takes as input a public key* pk. *At the conclusion of the protocol, $\mathcal{V}$ outputs a bit b with $b = 1$ indicating "accept" and $b = 0$ indicating "reject".*

We write IDS instead of $IDS(1^k)$, whenever the security parameter k is clear from context or irrelevant. For correctness of an IDS, *we require that for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}()$ *we have*

$$\mathsf{Pr}\left[\langle \mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk})\rangle = 1\right] = 1,$$

where $\langle \mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk})\rangle$ *refers to the common execution of the protocol between* $\mathcal{P}$ *with input* sk *and* $\mathcal{V}$ *on input* pk.

In this work we are concerned with 5-pass IDS, i.e. IDS where a transcript consists of five messages. A canonical 5-pass IDS is an IDS where the prover and the verifier exchange two challenges and replies. More formally:

Definition 2.4 (Canonical 5-pass identification schemes). *Consider* $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$, *a* 5*-pass identification scheme with two challenge spaces* C_1 *and* C_2. *We call* IDS *a canonical* 5*-pass identification scheme if the prover can be split into three subroutines* $\mathcal{P} = (\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2)$ *and the verifier into three subroutines* $\mathcal{V} = (\mathsf{ChS}_1, \mathsf{ChS}_2, \mathsf{Vf})$ *such that*

- $\mathcal{P}_0(\mathsf{sk})$ *computes the initial commitment* com *sent as the first message and a state* state *fed forward to* $\mathcal{P}_1$.
- ChS_1, *computes the first challenge message* $\mathsf{ch}_1 \leftarrow_R \mathsf{C}_1$, *sampling at random from the challenge space* C_1.
- $\mathcal{P}_1(\mathsf{state}, \mathsf{ch}_1)$, *computes the first response* resp_1 *of the prover (and updates the state* state*) given access to the state and the first challenge.*
- ChS_2, *computes the second challenge message* $\mathsf{ch}_2 \leftarrow_R \mathsf{C}_2$.
- $\mathcal{P}_2(\mathsf{state}, \mathsf{ch}_2)$, *computes the second response* resp_2 *of the prover given access to the state and the second challenge.*
- $\mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$, *upon access to the public key and the whole transcript outputs* $\mathcal{V}$*'s final decision.*

Note that the state forwarded among the prover algorithms can contain all inputs to previous prover algorithms if they are needed later. We also assume that the verifier keeps all sent and received messages to feed them to Vf. Figure 1 describes a canonical 5-pass IDS.

We will consider a particular type of 5-pass identification protocols where the size of the two challenge spaces is restricted to q and 2.

Definition 2.5 ($q2$-Identification scheme). *A $q2$-Identification scheme* IDS *is a canonical 5-pass identification scheme where for the challenge spaces* C_1 *and* C_2 *it holds that* $|\mathsf{C}_1| = q$ *and* $|\mathsf{C}_2| = 2$. *Moreover, the probability that the commitment* com *takes a given value is* $\leq 2^{-k}$, *where the probability is taken over the random choice of the input and the used randomness.*

Our goal is to construct signature schemes from identification schemes. It is well known that for this passively secure identification schemes suffice. In this setting, security is defined in terms of two properties: soundness and honest-verifier zero-knowledge (HVZK). To prove security of our signature scheme, we

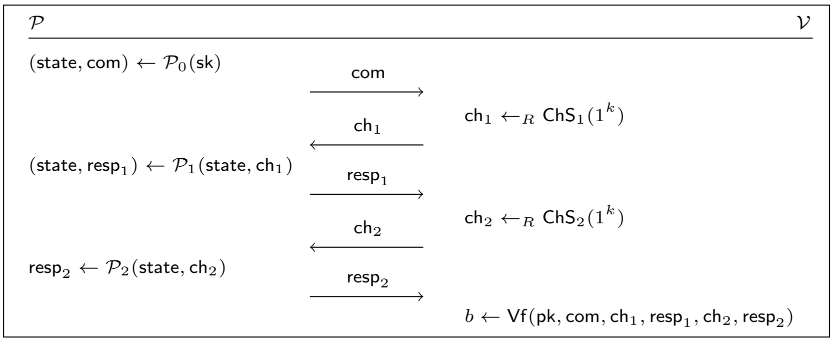
$\mathcal{P}$
$\mathcal{V}$
$(\mathsf{state}, \mathsf{com}) \leftarrow \mathcal{P}_0(\mathsf{sk})$
$\xrightarrow{\mathsf{com}}$
$\mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k)$
$\xleftarrow{\mathsf{ch}_1}$
$(\mathsf{state}, \mathsf{resp}_1) \leftarrow \mathcal{P}_1(\mathsf{state}, \mathsf{ch}_1)$
$\xrightarrow{\mathsf{resp}_1}$
$\mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k)$
$\xleftarrow{\mathsf{ch}_2}$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{state}, \mathsf{ch}_2)$
$\xrightarrow{\mathsf{resp}_2}$
$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

Fig. 1. Canonical 5-pass IDS

will not make use of soundness but of a similar property called $q2$-extractor which is a variant of special soundness. This is combined with a notion of key-one-wayness to later be able to argue about security. The issue with soundness is that the common proof-technique makes use of rewinding which becomes troublesome when dealing with quantum adversaries.

Definition 2.6 ((statistical) Honest-verifier zero-knowledge). *Let $k \in \mathbb{N}$, $\mathsf{IDS}(1^k) = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ an identification scheme with security parameter k. We say that* IDS *is statistical honest-verifier zero-knowledge if there exists a probabilistic polynomial time algorithm $\mathcal{S}$, called the simulator, such that the statistical distance between the following two distribution ensembles is negligible in k:*

$$\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}() : (\mathsf{sk}, \mathsf{pk}, \mathsf{trans}(\langle \mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk}) \rangle))\}$$
$$\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}() : (\mathsf{sk}, \mathsf{pk}, \mathcal{S}(\mathsf{pk}))\} .$$

Intuitively it must be hard for any cryptographic scheme to derive a valid secret key given a public key. To formally capture this intuition, we need to define what valid means. For this we define the notion of a key relation.

Definition 2.7. *(Key relation) Let* IDS *be a $q2$-Identification scheme and R some relation. We say* IDS *has key relation R if*

$$\forall (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}() : (\mathsf{pk}, \mathsf{sk}) \in R$$

Now that we have defined what valid means, we can define key-one-wayness.

Definition 2.8. *(PQ-KOW) Let $k \in \mathbb{N}$ be the security parameter, $\mathsf{IDS}(1^k)$ be a $q2$-Identification scheme with key relation R. We call* IDS *post-quantum key-one-way (PQ-KOW) (with respect to key relation R) if for all quantum polynomial time algorithms $\mathcal{A}$*

$$\begin{aligned} \mathrm{Succ}^{pq-kow}_{\mathsf{IDS}(1^k)}(\mathcal{A}) &= \mathsf{Pr}\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(), \mathsf{sk}' \leftarrow \mathcal{A}(\mathsf{pk}) : (\mathsf{pk}, \mathsf{sk}') \in R\right] \\ &= \mathrm{negl}(k) \end{aligned}$$

In [CHR+16] it was shown that in general, for $q2$-Identification Schemes, it is not possible to extract efficiently a matching secret key-only from two related transcripts (as in the case of 3-pass schemes fulfilling special soundness). In order to capture the nature of these schemes and provide sufficient conditions for efficient extraction, the authors proposed the definition of a $q2$-Extractor. In the following we give a post-quantum version of $q2$-Extractor that fixes two slight technical shortcomings of the definition in [CHR+16]. On the one hand, we add the algorithm that actually generates the transcripts to the definition, on the other hand we use the notion of key relation to capture what kind of secret key the extractor returns.

Definition 2.9 (PQ-$q2$-Extractor). *Let* $\mathsf{IDS}(1^k)$ *be a $q2$-Identification scheme with key relation R and $\mathcal{A}$ a quantum polynomial time algorithm that upon input of security parameter 1^k and an* $\mathsf{IDS}(1^k)$ *public key* pk *outputs, with non-negligible probability, four valid transcripts with respect to* pk*:*

$$\begin{aligned} \mathsf{trans}^{(1)} &= (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2) \\ \mathsf{trans}^{(2)} &= (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}'_2, \mathsf{resp}'_2) \\ \mathsf{trans}^{(3)} &= (\mathsf{com}, \mathsf{ch}'_1, \mathsf{resp}'_1, \mathsf{ch}_2, \mathsf{resp}_2) \\ \mathsf{trans}^{(4)} &= (\mathsf{com}, \mathsf{ch}'_1, \mathsf{resp}'_1, \mathsf{ch}'_2, \mathsf{resp}'_2) \end{aligned} \tag{1}$$

where $\mathsf{ch}_1 \neq \mathsf{ch}'_1$ *and* $\mathsf{ch}_2 \neq \mathsf{ch}'_2$.

We say that $\mathsf{IDS}(1^k)$ *has a* PQ-$q2$-Extractor *if there exists a quantum polynomial time algorithm $\mathcal{K}_{IDS}$, the extractor, that, given a public key* pk *and access to $\mathcal{A}$, outputs a secret key* sk *such that* $(\mathsf{pk}, \mathsf{sk}) \in R$ *with non-negligible success probability in k.*

A classical version of this definition is obtained by restricting $\mathcal{A}$ and $\mathcal{K}_{IDS}$ to classical PPT algorithms.

3 From $q2$-IDS to signatures in the QROM

In [CHR+16], the authors show that the Fiat-Shamir transformation can be generalized to the case of 5-pass IDS whose ChS_2 is bounded to two elements. They show that the Pointcheval-Stern proof [PS96] can be extended to this case, and the obtained signature scheme can be shown EU-CMA secure in the random oracle model. This result is further extended to any $2n+1$ round identification scheme that fulfills a certain kind of special soundness in [DGV+16]. However, similar to the standard Fiat-Shamir transform, their proofs rely on the forking lemma, which introduces two serious problems in the post-quantum setting: rewinding of the adversary, and adaptively programming the random oracle. While it is known how to deal with the latter [Unr15], the former seems to become a real show stopper [ARU14]. The only known way to fix the Fiat-Shamir transform in the QROM setting [DFG13] is using oblivious commitments, which are a certain kind of trapdoor commitments, effectively avoiding rewinding at the cost of introducing the necessity of a trapdoor function. This makes the solution not applicable in our setting as there are no trapdoor functions with a reduction from the $\mathcal{MQ}$-problem.

In [Unr15], Unruh proposes a different transform, based on Fischlin's transform [Fis05], that turns 3-pass zero-knowledge proofs into non-interactive ones in the QROM. In addition, Unruh shows how to use his transform to obtain a signature scheme. The transform essentially works by "unrolling" Fischlin's transform and then applying a few tweaks. This works, as Fischlin's transform already avoids rewinding. The basic idea is to let the signer generate several transcripts for a commitment. This is iterated for several initial commitments. Next, the signer "blinds" all responses in the transcripts by applying a length-preserving hash. All the obtained data is hashed together with the public key and the message to obtain a challenge vector. This challenge vector determines one transcript per commitment that has to be unblinded, i.e., for which the response must be included in the signature. The signature consists of all the transcripts with "blinded" responses and the unblinded responses for the transcripts identified by the challenge vector. The reasoning behind the transform is that without knowing the secret key, a forger cannot know sufficiently many valid openings to be able to include all the challenged responses. On the other hand, a security reduction can replace the length-preserving hash (modeled as QRO) by an invertible function (e.g. a QPRP). That way, a reduction can "unblind" the remaining responses in the signature by inverting the function. Now, it can be argued that an adversary with non-negligible success probability must have known several valid transcripts for at least one commitment. The unblinding reveals those transcripts and they can be used to run the extractor.

Here, we show that a similar transform can be applied to 5-pass IDS with a binary second challenge (i.e., q2-IDS). Basically, we treat the second challenge-response round like the first. However, as we have a binary second challenge, we ask that for each first challenge, a transcript for both values of the second challenge is generated.

The main difference between the security reduction of Unruh's transform and our extension to q2-IDS is a more involved argument to show that we get sufficiently many valid transcripts that follow the pattern needed to extract a valid secret key. As this argument is essentially independent of the RO, we first give a proof in the classical ROM. This also allows us to show that the reduction is tight in the ROM. Afterwards we describe how things change in the QROM along the lines of Unruh's QROM proof. This is where the reduction becomes loose. It remains an interesting open question if this is a fundamental issue with QROM reductions or if the existing techniques are just not sufficiently evolved, yet.

3.1 Extending Unruh's transform to q2-IDS

Let $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ be a q2-IDS, where we have $\mathcal{P} = (\mathcal{P}_0, \mathcal{P}_1, \mathcal{P}_2)$ and $\mathcal{V} = (\mathsf{ChS}_1, \mathsf{ChS}_2, \mathsf{Vf})$, and let $r, t \in \mathbb{N}$ be two parameters, where $2 \leqslant t \leqslant q$. Moreover let $\mathrm{H}_1 : \{0,1\}^{|\mathsf{resp}_1|} \rightarrow \{0,1\}^{|\mathsf{resp}_1|}$, $\mathrm{H}_2 : \{0,1\}^{|\mathsf{resp}_2|} \rightarrow \{0,1\}^{|\mathsf{resp}_2|}$, and $\mathcal{H}$:

$\{0,1\}^* \rightarrow \{0,1\}^{\lceil \log 2t \rceil r}$ be hash functions, later modeled as random oracles. We define the following digital signature scheme $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$. The key generation algorithm just runs $\mathsf{IDS.KGen}()$. Signature and verification algorithms are given in Figures 2 and 3.

```
$\mathsf{Sign}(\mathsf{sk}, M)$

For $j \in \{1, \ldots, r\}$ do
    $(\mathsf{state}^{(j)}, \mathsf{com}^{(j)}) \leftarrow \mathcal{P}_0(\mathsf{sk})$
    For $i \in \{1, \ldots, t\}$ do
        $\mathsf{ch}_1^{(i,j)} \leftarrow_R \mathsf{ChS}_1 \setminus \{\mathsf{ch}_1^{(1,j)}, \ldots, \mathsf{ch}_1^{(i-1,j)}\}$
        $(\mathsf{state}^{(i,j)}, \mathsf{resp}_1^{(i,j)}) \leftarrow \mathcal{P}_1(\mathsf{state}^{(j)}, \mathsf{ch}_1^{(i,j)})$
        $\mathsf{cr}_1^{(i,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(i,j)})$
        $\mathsf{resp}_2^{(i,j,0)} \leftarrow \mathcal{P}_2(\mathsf{state}^{(i,j)}, \mathsf{ch}_2 = 0), \mathsf{resp}_2^{(i,j,1)} \leftarrow \mathcal{P}_2(\mathsf{state}^{(i,j)}, \mathsf{ch}_2 = 1)$
        $\mathsf{cr}_2^{(i,j,0)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(i,j,0)}), \mathsf{cr}_2^{(i,j,1)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(i,j,1)})$

$\mathsf{trans}_{\mathsf{full}}(j) := \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}, (\mathsf{cr}_2^{(i,j,0)}, \mathsf{cr}_2^{(i,j,1)})\right\}_{i=1}^{t}$
$\mathsf{md} \leftarrow \mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$
Read $\mathsf{md}$ as vector $((I_1, B_1), \ldots, (I_r, B_r))$
$\mathsf{trans}_{\mathsf{red}}(j) := \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}, (\mathsf{cr}_2^{(i,j,0)}, \mathsf{cr}_2^{(i,j,1)})\right\}_{i \neq I_j, i=1}^{t}$
$\sigma := \left(\mathsf{md}, \left\{\mathsf{trans}_{\mathsf{red}}(j), \mathsf{ch}_1^{(I_j,j)}, \mathsf{resp}_1^{(I_j,j)}, \mathsf{resp}_2^{(I_j,j,B_j)}, \mathsf{cr}_2^{(I_j,j,\neg B_j)}\right\}_{j=1}^{r}\right)$
```

Fig. 2. Signature generation

For ease of exposition, we will use the notation $T(j, i, b)$ for a string that has the format of a transcript of the IDS (not necessarily a valid transcript), corresponding to the j-th round of the non-interactive protocol, with i and b being the indices of the corresponding challenges ch_1 and ch_2, i.e.

$$T(j, i, b) := (\mathsf{com}^{(j)}, \mathsf{ch}_1^{(i,j)}, \mathsf{resp}_1^{(i,j)}, \mathsf{ch}_2 = b, \mathsf{resp}_2^{(i,j,b)}),$$

where $j \in \{1, \ldots, r\}$, $i \in \{1, \ldots, t\}$, $b \in \{0, 1\}$.

3.2 PQ-EU-CMA-Security in the ROM

In the following, we first establish post-quantum security under key-only attacks (PQ-KOA). More specifically, we will show that a successful KOA-forger $\mathcal{A}$ can be used to extract a valid secret key for the underlying IDS. Afterwards, we will extend the result to existential unforgeability under chosen message attacks.

```
$\mathsf{Vf}(\mathsf{pk}, \sigma, \mathbf{M})$
Read $\mathsf{md}$ as vector $((I_1, B_1), \ldots, (I_r, B_r))$
For $j \in \{1, \ldots, r\}$ do
    $\mathsf{cr}_1^{(I_j,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(I_j,j)})$
    $\mathsf{cr}_2^{(I_j,j,B_j)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(I_j,j,B_j)})$

$\mathsf{md}' \leftarrow \mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$
Check that $\mathsf{md}' \stackrel{?}{=} \mathsf{md}$
For $j \in \{1, \ldots, r\}$ do
    Check that $\mathsf{ch}_1^{(1,j)}, \ldots \mathsf{ch}_1^{(t,j)}$are all distinct
    Check $1 \stackrel{?}{=} b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}^{(j)}, \mathsf{ch}_1^{(I_j,j)}, \mathsf{resp}_1^{(I_j,j)}, B_j, \mathsf{resp}_2^{(I_j,j,B_j)})$
If all checks succeed, output success.
```

Fig. 3. Verification

PQ-KOW ⇒ PQ-KOA. The following lemma gives an exact relation between the key-one-wayness of the identification scheme and the security of the proposed signature scheme under key-only attacks.

Lemma 3.1. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme from Figures 2 and 3 above, using a q2-IDS that has a key relation R, a PQ-q2-extractor, and is PQ-KOW secure. Let $\mathcal{A}$ be a quantum algorithm that implements a KOA forger which given only the public key* pk *outputs a valid message-signature pair (M, σ) with probability ϵ. Then, in the random oracle model there exists an algorithm $\mathcal{M}^{\mathcal{A}}$ that given oracle access to any such $\mathcal{A}$ breaks the KOW security of* IDS *in essentially the same running time as the given $\mathcal{A}$ and with success probability*

$$\epsilon' \geq \epsilon - (q_{\mathcal{H}} + 1) 2^{-r \log \frac{2t}{t+1}}. \qquad (2)$$

Proof. We now show how to construct such an algorithm $\mathcal{M}^{\mathcal{A}}$. On input of an IDS public key pk, $\mathcal{M}^{\mathcal{A}}$ first runs $\mathcal{A}(\mathsf{pk})$. Let $\mathcal{E}_A$ be the event that $\mathcal{A}$ outputs a valid message-signature pair (M, σ) with

$$\sigma = \left(\mathsf{md}, \left\{\mathsf{trans}_{\mathsf{red}}(j), \mathsf{ch}_1^{(I_j,j)}, \mathsf{resp}_1^{(I_j,j)}, \mathsf{resp}_2^{(I_j,j,B_j)}, \mathsf{cr}_2^{(I_j,j,\neg B_j)}\right\}_{j=1}^{r}\right).$$

Then $\mathcal{E}_A$ implies that for every $j \in \{1, \ldots, r\}$, $T(j, I_j, J_j)$ is a valid transcript of IDS and the Verifier Vf accepts. Now, our goal is to use the $q2$-extractor to extract. This means, we need to obtain four valid transcripts $T(j, i_1, 0)$, $T(j, i_1, 1)$, $T(j, i_2, 0)$, $T(j, i_2, 1)$ for some $j \in \{1, \ldots, r\}$. To this end, $\mathcal{M}^{\mathcal{A}}$ simulates the random oracles H_1 and H_2 for $\mathcal{A}$ in the common way. The important point is that this way $\mathcal{M}^{\mathcal{A}}$ learns all of $\mathcal{A}$'s queries together with the given responses. Hence, when given $\mathcal{A}$'s forgery, $\mathcal{M}^{\mathcal{A}}$ can open all blinded responses in the signature.

Now, $\mathcal{M}^{\mathcal{A}}$ will only fail to extract if among all the $2tr$ opened transcripts of the signature, there are no four valid transcripts with the above pattern. Consider the event $\mathcal{E}_{\neg\text{ext}}$ which describes this case.

$\mathcal{E}_{\neg\text{ext}}$: For every $j \in \{1, \ldots, r\}$, and for every $i_1, i_2 \in \{1, \ldots, t\}$, $i_1 \neq i_2$, at least one of $T(j, i_1, 0)$, $T(j, i_1, 1)$, $T(j, i_2, 0)$, $T(j, i_2, 1)$ is not a valid transcript of the IDS.

We will now upper bound $\Pr[(\mathcal{E}_A \cap \mathcal{E}_{\neg\text{ext}})]$ and thereby lower bound $\mathcal{M}^{\mathcal{A}}$'s success probability.

Let (M, σ) be $\mathcal{A}$'s output under the event $(\mathcal{E}_A \cap \mathcal{E}_{\neg\text{ext}})$. First, (M, σ) must be valid because of $\mathcal{E}_{\mathcal{A}}$. Now, consider the set $\mathcal{S}_{\neg\text{ext}}$ of tuples $\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$, such that for every $j \in \{1, \ldots, r\}$ there is at most one I_j^* with $T(j, I_j^*, 0)$ and $T(j, I_j^*, 1)$ being valid transcripts of IDS. It is clear that $\mathcal{A}$'s output under the event $\mathcal{E}_A \cap \mathcal{E}_{\neg\text{ext}}$ must come from $\mathcal{S}_{\neg\text{ext}}$. Indeed, if a tuple does not satisfy the given condition, then there exist at least two indices I_j^*, I_j^{**} such that $T(j, I_j^*, 0)$, $T(j, I_j^*, 1)$, $T(j, I_j^{**}, 0)$, $T(j, I_j^{**}, 1)$ are valid transcripts of IDS, which is in contradiction to the event $\mathcal{E}_{\neg\text{ext}}$.

Let $\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$ be such a tuple. Then the indexes that define the required openings in σ are obtained as the output of the random oracle $\mathcal{H}$ on input of the tuple, i.e.

$$((I_1, B_1), \ldots, (I_r, B_r)) \leftarrow \mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$$

In order for the signature to pass verification, for each $j \in \{1, \ldots, r\}$, the transcript $T(j, I_j, B_j)$ must be valid. Given the conditions of $\mathcal{E}_{\neg\text{ext}}$, for each $j \in \{1, \ldots, r\}$, there are at most $t+1$ valid transcripts per j. Hence for the entire $((I_1, B_1), \ldots, (I_r, B_r))$ at most $(t+1)^r$ possible values. Thus, the probability for the adversary to produce a valid signature from such a tuple is $\frac{(t+1)^r}{(2t)^r} = 2^{-r \log \frac{2t}{t+1}}$.

Now let $q_{\mathcal{H}}$ be the number of queries of the adversary to the oracle $\mathcal{H}$. We have that

$$\Pr(\mathcal{E}_A \cap \mathcal{E}_{\neg\text{ext}}) \leq (q_{\mathcal{H}} + 1) 2^{-r \log \frac{2t}{t+1}},$$

as $\mathcal{A}$ can try at most $q_{\mathcal{H}}$ tuples to obtain a valid signature and output a signature based on a new tuple otherwise. Towards obtaining a bound on $\mathcal{M}^{\mathcal{A}}$'s success probability, note that $\mathcal{M}^{\mathcal{A}}$ succeeds in the event $(\mathcal{E}_A \cap \neg\mathcal{E}_{\neg\text{ext}})$. For this event we get

$$\Pr(\mathcal{E}_A \cap \neg\mathcal{E}_{\neg\text{ext}}) = \Pr(\mathcal{E}_A) - \Pr(\mathcal{E}_A \cap \mathcal{E}_{\neg\text{ext}}) \geq \epsilon - (q_{\mathcal{H}} + 1) 2^{-r \log \frac{2t}{t+1}}.$$

This proves the claimed bound. □

PQ-KOA ⇒ PQ-EU-CMA. Given the above lemma, it suffices to reduce PQ-KOA to PQ-EU-CMA security to eventually prove PQ-EU-CMA security of the proposed scheme, i.e. we have to show that we can answer an adversary's signature queries without knowledge of a secret key. This is done in the following lemma. Afterwards we can derive the main theorem of the section.

Lemma 3.2. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme from Figures 2 and 3 above, using a q2-IDS that is honest-verifier zero-knowledge. Let $\mathcal{A}$ be a quantum algorithm that breaks the PQ-EU-CMA security of the signature scheme with probability ϵ. Then, in the random oracle model there exists an algorithm $\mathcal{M}^{\mathcal{A}}$ that breaks the PQ-KOA security of the signature scheme in essentially the same running time as $\mathcal{A}$ and with success probability*

$$\epsilon' \geq \epsilon(1 - q_{\mathsf{Sign}} q_{\mathcal{H}} 2^{-rk}). \tag{3}$$

Proof. We show how to construct $\mathcal{M}^{\mathcal{A}}$ that on input a public key pk of the signature scheme (which is also a public key for IDS), access to a HVZK-simulator $\mathcal{S}_{\mathsf{IDS}}$ for IDS and the random oracles $\mathrm{H}_1, \mathrm{H}_2, \mathcal{H}$, breaks the KOA security of the signature scheme. The running time and success probability of $\mathcal{M}^{\mathcal{A}}$ are essentially the same as that of $\mathcal{A}$ up to a negligible difference.

Upon receiving the public key pk, $\mathcal{M}^{\mathcal{A}}$ runs $\mathcal{A}(\mathsf{pk})$, simulating all signature and random oracle queries for $\mathcal{A}$. Whenever $\mathcal{A}$ queries H_1 or H_2, $\mathcal{M}^{\mathcal{A}}$ simply forwards the query to his respective RO. For $\mathcal{H}$, $\mathcal{M}^{\mathcal{A}}$ keeps a local list $\mathcal{L}_{\mathcal{H}}$. Whenever $\mathcal{A}$ queries $\mathcal{H}$, $\mathcal{M}^{\mathcal{A}}$ first checks $\mathcal{L}_{\mathcal{H}}$ and returns the stored answer if one exists. Otherwise, $\mathcal{M}^{\mathcal{A}}$ forwards the query to his oracle $\mathcal{H}$ and stores the query together with the result in $\mathcal{L}_{\mathcal{H}}$ before returning the response.

Whenever $\mathcal{A}$ makes a signature query on a message M, $\mathcal{M}^{\mathcal{A}}$ does the following:

1. Samples $\tilde{\mathsf{md}} \leftarrow_R \{0,1\}^{\lceil \log 2t \rceil r}$ and interprets it as challenge string, i.e., $((I_1, B_1), \dots, (I_r, B_r)) := \tilde{\mathsf{md}}$.
2. Runs the HVZK-simulator $\mathcal{S}_{\mathsf{IDS}}$ r times to obtain r valid transcripts of IDS:
$$\left\{(\mathsf{com}^{(j)}, \mathsf{ch}_1^{(I_j,j)}, \mathsf{resp}_1^{(I_j,j)}, \mathsf{ch}_2^{(j)}, \mathsf{resp}_2^{(I_j,j,B_j)})\right\}_{j=1}^{r},$$
and uses them as the challenged transcripts $T(j, I_j, B_j)$ for $j \in \{1, \dots, r\}$.
3. Blinds the responses $\mathsf{resp}_1^{(I_j,j)}$ and $\mathsf{resp}_2^{(I_j,j,B_j)}$ for every $j \in \{1, \dots, r\}$:
$$\mathsf{cr}_1^{(I_j,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(I_j,j)}), \quad \mathsf{cr}_2^{(I_j,j,B_j)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(I_j,j,B_j)})$$
4. For all $j \in \{1, \dots, r\}$, and all $(i, b) \in \{1, \dots, t\} \times \{0, 1\} \backslash \{(I_j, B_j)\}_{j=1}^{r}$ samples a first challenge
$$\mathsf{ch}_1^{(i,j)} \leftarrow_R \mathsf{ChS}_1 \setminus \{\mathsf{ch}_1^{(I_j,j)}, \mathsf{ch}_1^{(1,j)}, \dots, \mathsf{ch}_1^{(i-1,j)}\},$$
samples fake responses
$$\mathsf{resp}_1^{(i,j)} \leftarrow_R \mathsf{RespS}_1, \quad \mathsf{resp}_2^{(i,j,b)} \leftarrow_R \mathsf{RespS}_2,$$
blinds the fake responses
$$\mathsf{cr}_1^{(i,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(i,j)}), \mathsf{cr}_2^{(i,j,b)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(i,j,b)}),$$
and sets
$$\mathsf{trans}_{\mathsf{full}}(j) := \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}, (\mathsf{cr}_2^{(i,j,0)}, \mathsf{cr}_2^{(i,j,1)})\right\}_{i=1}^{t}.$$

5. Checks if there is already an entry for $\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$ in $\mathcal{L}_{\mathcal{H}}$. If so, $\mathcal{M}^{\mathcal{A}}$ aborts. Otherwise, $\mathcal{M}^{\mathcal{A}}$ stores $\left(\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right), \tilde{\mathsf{md}}\right)$ in $\mathcal{L}_{\mathcal{H}}$.
6. Outputs the signature

$$\sigma = \left(\mathsf{md}, \left\{\mathsf{trans}_{\mathsf{red}}(j), \mathsf{ch}_1^{(I_j,j)}, \mathsf{resp}_1^{(I_j,j)}, \mathsf{resp}_2^{(I_j,j,B_j)}, \mathsf{cr}_2^{(I_j,j,\neg B_j)}\right\}_{j=1}^{r}\right),$$

where $\mathsf{trans}_{\mathsf{red}}(j) := \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}, (\mathsf{cr}_2^{(i,j,0)}, \mathsf{cr}_2^{(i,j,1)})\right\}_{i \neq I_j, i=1}^{t}$

Finally, $\mathcal{M}^{\mathcal{A}}$ outputs whatever $\mathcal{A}$ outputs.

Now, $\mathcal{M}^{\mathcal{A}}$ succeeds exactly with $\mathcal{A}$'s success probability as long as it does not abort. All RO queries follow the correct distribution and so do the signatures. An abort only occurs if $\mathcal{A}$ queried $\mathcal{H}$ before on the value for which $\mathcal{M}^{\mathcal{A}}$ wants to program. The value has the form $\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$ with $\mathsf{trans}_{\mathsf{full}}(j) := \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}, (\mathsf{cr}_2^{(i,j,0)}, \mathsf{cr}_2^{(i,j,1)})\right\}_{i=1}^{t}$. The $\{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}$ term has at least rk bits of entropy as the commitments have at least k bits of entropy according to the definition of $q2$-IDS and there is one commitment for each of the r rounds. This is merely a very loose (but more than sufficient) lower bound on the entropy as the blinded responses also add additional entropy. Hence, if $\mathcal{A}$ makes a total of $q_{\mathcal{H}}$ queries for $\mathcal{H}$ and q_{Sign} signature queries, an abort occurs with probability

$$\mathsf{Pr}[abort] \leq q_{\mathsf{Sign}} q_{\mathcal{H}} 2^{-rk}.$$

Hence, $\mathcal{M}^{\mathcal{A}}$ succeeds with probability

$$\epsilon' \geq \epsilon(1 - q_{\mathsf{Sign}} q_{\mathcal{H}} 2^{-rk}).$$

□

PQ-KOW $\Rightarrow$ PQ-EU-CMA. Combining the two previous lemmas we obtain the following theorem.

Theorem 3.3. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme from Figures 2 and 3 above using a $q2$-IDS* IDS *that is statistical honest-verifier zero-knowledge and has a PQ-$q2$-extractor. Let $\mathcal{A}$ be a PQ-EU-CMA forger that succeeds with probability ϵ. Then, there exists an algorithm $\mathcal{M}^{\mathcal{A}}$, that in the random oracle model breaks the PQ-KOW security of* IDS *in essentially the same running time as $\mathcal{A}$ and with success probability*

$$\epsilon' \geq \epsilon - \epsilon q_{\mathsf{Sign}} q_{\mathcal{H}} 2^{-rk} - (q_{\mathcal{H}} + 1) 2^{-r \log \frac{2t}{t+1}},$$

Proof. Suppose there exists a PQ-EU-CMA forger $\mathcal{A}$ that succeeds with non-negligible probability ϵ. We construct a PQ-KOW adversary $\mathcal{C}$ for the $q2$-IDS as follows.

$\mathcal{C}$ runs $\mathcal{A}(\mathsf{pk})$, to construct a key-only forger $\mathcal{M}^{\mathcal{A}}$ as in Lemma 3.2, that succeeds with probability (3). Now as in Lemma 3.1, $\mathcal{C}$ can extract a valid secret key sk, in approximately the same time, and with only negligibly smaller probability (see (2)). In total the success probability of $\mathcal{C}$ is

$$\epsilon' \geq \epsilon - \epsilon q_{\mathsf{Sign}} q_{\mathcal{H}} 2^{-rk} - (q_{\mathcal{H}} + 1) 2^{-r \log \frac{2t}{t+1}},$$

and the running time of $\mathcal{C}$ is essentially the same as that of $\mathcal{A}$. □

3.3 PQ-EU-CMA security in the QROM

We now show that with only slight changes, the two lemmas above also hold in the QROM. We do this in reverse order, starting with the PQ-KOA to PQ-EU-CMA reduction as it is the easier case. As already the QROM proofs in Unruh's work which we build on are non-tight, we only give our arguments in the asymptotic regime.

PQ-KOA ⇒ PQ-EU-CMA. We will first revisit the reduction from PQ-KOA to PQ-EU-CMA. We show the following lemma:

Lemma 3.4. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme from Figures 2 and 3 above, using a q2-IDS that is honest-verifier zero-knowledge. Let $\mathcal{A}$ be a quantum algorithm that breaks the PQ-EU-CMA security of the signature scheme with probability ϵ. Then, in the quantum-accessible random oracle model there exists a quantum algorithm $\mathcal{M}^{\mathcal{A}}$ that breaks the PQ-KOA security of the signature scheme in essentially the same running time as $\mathcal{A}$ and with success probability*

$$\epsilon' \geq \epsilon(1 - \mathrm{negl}(k)). \tag{4}$$

Proof (Sketch). The proof in the ROM above also applies in the QROM with essentially a single change. The queries to H_1 and H_2 are still just forwarded by $\mathcal{M}^{\mathcal{A}}$ without interaction. This works without any issues in the QROM given that $\mathcal{M}^{\mathcal{A}}$ is now a quantum algorithm (which is unavoidable in the QROM). The only issue is the way $\mathcal{M}^{\mathcal{A}}$ handles $\mathcal{H}$. It is not possible anymore for $\mathcal{M}^{\mathcal{A}}$ to learn $\mathcal{A}$'s queries to $\mathcal{H}$ and thereby not possible to abort. However, we only added the abort condition above for clarity: in the classical case $\mathcal{M}^{\mathcal{A}}$ could also simply always program $\mathcal{H}$. Then $\mathcal{A}$'s success probability might change if $\mathcal{M}^{\mathcal{A}}$ programmed on an input previously queried by $\mathcal{A}$. However, we still obtain the same bound on the probability. In the QROM, Unruh showed in [Unr15, Corollary 11] that this adaptive programming only negligibly changes $\mathcal{A}$'s success probability (the exact argument for our specific case is exactly the one made in the first game hop of the proof of Theorem 15 in [Unr15]). From this it follows that $\mathcal{M}^{\mathcal{A}}$'s success probability still only negligibly deviates from that of $\mathcal{A}$. □

PQ-KOW ⇒ PQ-KOA. Now we revisit the reduction from PQ-KOW to PQ-KOA in the quantum-accessible ROM. While we still do this in the asymptotic regime, we make the parts of the reduction loss explicit which depend on the parameters r, t of the scheme. This is to support parameter selection in later sections.

Lemma 3.5. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme from Figures 2 and 3 above, using a q2-IDS that has a key relation R, a PQ-q2-extractor, and is PQ-KOW secure. Let $\mathcal{A}$ be a quantum algorithm that implements a KOA forger which given only the public key* pk *outputs a valid message-signature pair (M, σ) with probability ϵ. Then, in the quantum-accessible random oracle model there exists a quantum algorithm $\mathcal{M}^{\mathcal{A}}$ that given oracle access to any such $\mathcal{A}$ breaks the KOW security of* IDS *in essentially the same running time as the given $\mathcal{A}$ and with success probability*

$$\epsilon' \geq \epsilon - 2(q_{\mathcal{H}} + 1)2^{-(r \log \frac{2t}{t+1})/2}. \tag{5}$$

Proof (Sketch). A QROM version of our proof is obtained by essentially following the proof of Lemma 17 in [Unr15]. The changes in the proof above are as follows. First, $\mathcal{M}^{\mathcal{A}}$ cannot learn $\mathcal{A}$'s RO queries to H_1 and H_2 by simulating these the classical way, anymore. Instead, $\mathcal{M}^{\mathcal{A}}$ simulates these oracles using one quantum PRP (QPRP) per oracle with a random secret key per QPRP. QPRPs exist as shown in [Zha16] and they are quantum indistinguishable from random functions. Now, $\mathcal{M}^{\mathcal{A}}$ can open the blinded responses in the signature by inverting the QPRP using the secret key. Second, the analysis of $\Pr(\mathcal{E}_A \cap \mathcal{E}_{\neg\mathrm{ext}})$ changes. As we have shown, the probability of a tuple from $\mathcal{E}_{\neg\mathrm{ext}}$ to lead to a valid signature is $2^{-r \log \frac{2t}{t+1}}$. We can now follow the analysis in [Unr15] that reduces distinguishing the constant zero function from a Bernoulli distributed boolean function to finding a tuple in $\mathcal{E}_{\neg\mathrm{ext}}$ that leads to a valid signature. Thereby we get the claimed bound:

$$\Pr(\mathcal{E}_A \cap \mathcal{E}_{\neg\mathrm{ext}}) \leq 2(q_{\mathcal{H}} + 1)2^{-(r \log \frac{2t}{t+1})/2}.$$

□

PQ-KOW $\Rightarrow$ PQ-EU-CMA. Putting the above two lemmas together allows us to state the following theorem.

Theorem 3.6. *Let $k, t, r \in \mathbb{N}$ be the parameters of the signature scheme above using a q2-IDS* IDS *that is statistical honest-verifier zero-knowledge and has a PQ-q2-extractor. Let $\mathcal{A}$ be a PQ-EU-CMA forger that succeeds with probability ϵ. Then, there exists a quantum algorithm $\mathcal{M}^{\mathcal{A}}$, that in the quantum-accessible random oracle model breaks the PQ-KOW security of* IDS *in essentially the same running time as $\mathcal{A}$ and with success probability*

$$\epsilon' \geq (\epsilon - 2(q_{\mathcal{H}} + 1)2^{-(r \log \frac{2t}{t+1})/2})(1 - \mathrm{negl}(k)).$$

4 The Sakumoto-Shirai-Hiwatari 5-pass IDS scheme

In [SSH11], Sakumoto, Shirai, and Hiwatari proposed two new identification schemes, a 3-pass and a 5-pass IDS, based on the intractability of the $\mathcal{MQ}$ problem. Unlike previous public key schemes, their solution provably relies only on the $\mathcal{MQ}$ problem (and the security of the commitment scheme), and not on other related problems in multivariate cryptography such as the Isomorphism of Polynomials (IP) problem [Pat96], the related Extended IP [DHYC06] and IP with partial knowledge [Tho13] problems or the MinRank problem [Cou01,FLP08]. Let us quickly recall the $\mathcal{MQ}$ problem.

Definition 4.1 ($\mathcal{MQ}$ problem (search version)). *Let $m, n, q \in \mathbb{N}$, $\mathbf{x} = (x_1, \ldots, x_n)$ and let $\mathcal{MQ}(n, m, \mathbb{F}_q)$ denote the family of vectorial functions $\boldsymbol{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ of degree 2 over $\mathbb{F}_q$:*

$$\begin{aligned}\mathcal{MQ}(n, m, \mathbb{F}_q) = \{&\boldsymbol{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))|\\ &f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i, s \in \{1, \ldots, m\}\}.\end{aligned}$$

An instance $\mathcal{MQ}(\boldsymbol{F}, \mathbf{v})$ of the $\mathcal{MQ}$ (search) problem is defined as:
Given $\boldsymbol{F} \in \mathcal{MQ}(n, m, \mathbb{F}_q), \mathbf{v} \in \mathbb{F}_q^m$ find, if any, $\mathbf{s} \in \mathbb{F}_q^n$ such that $\boldsymbol{F}(\mathbf{s}) = \mathbf{v}$.

The decisional version of the $\mathcal{MQ}$ problem is NP-complete [GJ79]. It is widely believed that the $\mathcal{MQ}$ problem is intractable even for quantum computers in the average case, i.e., that there exists no polynomial time quantum algorithm that given $\mathbf{F} \leftarrow_R \mathcal{MQ}(n, m, \mathbb{F}_q)$ and $\mathbf{v} = \mathbf{F}(\mathbf{s})$ (for random $\mathbf{s} \leftarrow_R \mathbb{F}_q^n$) outputs a solution $\mathbf{s}'$ to the $\mathcal{MQ}(\mathbf{F}, \mathbf{v})$ problem with non-negligible probability.

We will later also need the $\mathcal{MQ}$ relation $R_{\mathcal{MQ}}$ which is the relation of $\mathcal{MQ}$ instances and solutions:

Definition 4.2 ($\mathcal{MQ}$ relation). *The $\mathcal{MQ}$ relation is a binary relation $R_{\mathcal{MQ}(m,n,q)} : (\mathcal{MQ}(n, m, \mathbb{F}_q) \times \mathbb{F}_q^m) \times \mathbb{F}_q^n$ with*

$$((\boldsymbol{F}, \mathbf{v}), \mathbf{s}) \in R_{\mathcal{MQ}(m,n,q)}, \text{ iff } \boldsymbol{F}(\mathbf{s}) = \mathbf{v}.$$

We will omit m, n, q whenever they are clear from the context.

In [SSH11], Sakumoto, Shirai, and Hiwatari propose a clever splitting technique, using the so-called polar form of the function $\mathbf{F}$ which is the function $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$. Using the polar form and its bilinearity, it becomes possible to split a secret into two shares, such that none of the shares on its own leaks anything about the secret.

Using this result, they showed how to construct zero knowledge arguments of knowledge for the $\mathcal{MQ}$ problem, using a statistically hiding and computationally binding commitment scheme. They present a 3- and a 5-pass protocol with differing performance properties. Later, in [CHR⁺16] the security properties of the 5-pass scheme were reexamined to provide the minimal requirements for Fiat-Shamir type signatures from 5-pass IDS. For completeness and better readability we provide the description of the 5-pass IDS, together with the properties that we will use.

Let $(\mathsf{pk}, \mathsf{sk}) = ((\mathbf{F}, \mathbf{v}), \mathbf{s}) \in R_{\mathcal{MQ}}$ be the public and private keys of the Prover (i.e., key generation just samples from the $\mathcal{MQ}$ relation). Without loss of generality, let the elements from $\mathbb{F}_q$ be $\alpha_1, \ldots, \alpha_q$. The 5-pass IDS from [SSH11] is given in Figure 4.

$\mathcal{P}(\mathsf{pk},\mathsf{sk})$		$\mathcal{V}(\mathsf{pk})$
$//setup$		
$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$		
$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$		
$//commit$		
$c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$		
$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$	$\xrightarrow{\mathsf{com} = (c_0, c_1)}$	$//challenge\ 1$ $I \leftarrow_R \{1, \ldots, q\}$
$//first\ response$	$\xleftarrow{\mathsf{ch}_1 = I}$	
$\mathbf{t}_1 \leftarrow \alpha_I \mathbf{r}_0 - \mathbf{t}_0$		
$\mathbf{e}_1 \leftarrow \alpha_I \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$	$\xrightarrow{\mathsf{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$	$//challenge\ 2$ $\mathsf{ch}_2 \leftarrow_R \{0, 1\}$
$//second\ response$	$\xleftarrow{\mathsf{ch}_2}$	
If $\mathsf{ch}_2 = 0,\ \mathsf{resp}_2 \leftarrow \mathbf{r}_0$		
Else $\mathsf{resp}_2 \leftarrow \mathbf{r}_1$	$\xrightarrow{\mathsf{resp}_2}$	$//verify$ If $\mathsf{ch}_2 = 0,$ parse $\mathsf{resp}_2 = \mathbf{r}_0,$ check $c_0 \stackrel{?}{=} Com(\mathbf{r}_0, \alpha_I \mathbf{r}_0 - \mathbf{t}_1, \alpha_I \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1)$ Else, parse $\mathsf{resp}_2 = \mathbf{r}_1,$ check $c_1 \stackrel{?}{=} Com(\mathbf{r}_1, \alpha_I(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$

Fig. 4. The 5-pass IDS by Sakumoto, Shirai, and Hiwatari

The following theorem summarizes the security properties of the 5-pass identification scheme from [SSH11] that we need.

Theorem 4.3. *The 5-pass identification scheme from [SSH11] (see Fig. 4)*

1. *is statistically honest verifier zero-knowledge when the commitment scheme Com is statistically hiding,*
2. *has key relation $R_{\mathcal{MQ}(m,n,q)}$,*
3. *is post-quantum key-one-way if the $\mathcal{MQ}$ search problem is hard on average, and*
4. *has a PQ-q2-Extractor when the commitment scheme Com is computationally binding against quantum polynomial time algorithms.*

The first statement was shown in [SSH11]. The second statement holds by construction. The third statement follows from the second. The PQ-q2-Extractor essentially follows from a proof in [CHR$^+$16]. In [CHR$^+$16] the existence of a (non-quantum) q2-Extractor was proven under the condition that the commitment scheme is computationally binding. The proof shows that there exists a PPT algorithm that given four valid transcripts of the IDS with the correct pattern always either extracts a secret key or outputs two valid openings for the commitment. Hence, as long as the used commitment scheme achieves the traditional definition of computationally binding also against quantum polynomial time algorithms, the 5-pass identification scheme from [SSH11] has a PQ-q2-Extractor (as the probability to output two valid openings must be negligible).

5 Instantiation from the Sakumoto-Shirai-Hiwatari 5-pass IDS

In the previous sections, we have defined a signature scheme as the result of a transformed $q2$-IDS scheme. Here, we define it instantiated with the 5-pass identification scheme proposed in [SSH11].

5.1 SOFIA

We define the signature scheme in generic terms by describing the required parameters and the functions KGen, Sign and Vf, and defer giving concrete parameters m, n, r, t and $\mathbb{F}_q$ for a specific security parameter k to the next section. For now, we only need to fix $2 \leqslant t \leqslant q$ elements of the field $\mathbb{F}_q$. Without loss of generality, we denote them by $\alpha_1, \ldots, \alpha_t$.

Key generation. The SOFIA key generation algorithm formally just samples a $\mathcal{MQ}$ relation. Practically, the algorithm is realized as shown in Figure 5.

```
KGen()
$\mathsf{sk} \leftarrow_R \{0,1\}^k$
$S_{\mathbf{F}}, \mathbf{s}, S_{\mathbf{rte}} \leftarrow \mathrm{PRG}_{\mathsf{sk}}(\mathsf{sk})$
$\mathbf{F} \leftarrow \mathrm{XOF}_{\mathbf{F}}(S_{\mathbf{F}})$
$\mathbf{v} \leftarrow \mathbf{F}(\mathbf{s})$
$\mathsf{pk} := (S_{\mathbf{F}}, \mathbf{v})$
Return $(\mathsf{pk}, \mathsf{sk})$
```

Fig. 5. SOFIA key generation

The secret key is used as a seed to derive the following values:

- $S_{\mathbf{F}}$, a seed from which the system parameter $\mathbf{F}$ is expanded;
- $\mathbf{s}$, the secret input to the $\mathcal{MQ}$ function;
- $S_{\mathbf{rte}}$, a seed that is used to sample all vectors $\mathbf{r}_0^{(i)}$, $\mathbf{t}_0^{(i)}$ and $\mathbf{e}_0^{(i)}$. Note that this seed is not yet needed during key generation, but is required during signing.

Signature generation. For the signing procedure, we assume as input a message $M \in \{0,1\}^*$ and a secret key sk. The signing procedure is given in Figure 6. Note that the scheme definition includes several optimizations to reduce the signature size. We discuss these later in this section.

```
$\mathsf{Sign}(\mathsf{sk}, M)$
$S_{\mathbf{F}}, \mathbf{s}, S_{\mathbf{rte}} \leftarrow \mathrm{PRG}_{\mathsf{sk}}(\mathsf{sk})$
$\mathbf{F} \leftarrow \mathrm{XOF}_{\mathbf{F}}(S_{\mathbf{F}})$
$\mathsf{pk} := (S_{\mathbf{F}}, \mathbf{F}(\mathbf{s}))$
$\mathbf{r}_0^{(1)}, \ldots, \mathbf{r}_0^{(r)}, \mathbf{t}_0^{(1)}, \ldots, \mathbf{t}_0^{(r)}, \mathbf{e}_0^{(1)}, \ldots, \mathbf{e}_0^{(r)} \leftarrow \mathrm{PRG}_{\mathbf{rte}}(S_{\mathbf{rte}}, M)$
For $j \in \{1, \ldots, r\}$ do
    $\mathbf{r}_1^{(j)} \leftarrow \mathbf{s}^{(j)} - \mathbf{r}_0^{(j)}$
    $c_0^{(j)} \leftarrow Com(\mathbf{r}_0^{(j)}, \mathbf{t}_0^{(j)}, \mathbf{e}_0^{(j)})$
    $c_1^{(j)} \leftarrow Com(\mathbf{r}_1^{(j)}, \mathbf{G}(\mathbf{t}_0^{(j)}, \mathbf{r}_1^{(j)}) + \mathbf{e}_0^{(j)})$
    $\mathsf{com}^{(j)} := (c_0^{(j)}, c_1^{(j)})$
    For $i \in \{1, \ldots, t\}$ do
        $\mathbf{t}_1^{(i,j)} \leftarrow \alpha_i \mathbf{r}_0^{(j)} - \mathbf{t}_0^{(j)}, \quad \mathbf{e}_1^{(i,j)} \leftarrow \alpha_i \mathbf{F}(\mathbf{r}_0^{(j)}) - \mathbf{e}_0^{(j)}$
        $\mathsf{resp}_1^{(i,j)} := (\mathbf{t}_1^{(i,j)}, \mathbf{e}_1^{(i,j)})$
        $\mathsf{cr}_1^{(i,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(i,j)})$
    $\mathsf{resp}_2^{(j,0)} := \mathbf{r}_0^{(j)}, \mathsf{resp}_2^{(j,1)} := \mathbf{r}_1^{(j)}$
    $\mathsf{cr}_2^{(j,0)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(j,0)}), \mathsf{cr}_2^{(j,1)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(j,1)})$
    $\mathsf{trans}_{\mathsf{full}}(j) := (\mathsf{com}^{(j)}, \left\{\mathsf{cr}_1^{(i,j)}\right\}_{i=1}^{t}, \mathsf{cr}_2^{(j,0)}, \mathsf{cr}_2^{(j,1)})$
$\mathsf{md} \leftarrow \mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$
$((I_1, B_1), \ldots, (I_r, B_r)) \leftarrow \mathrm{XOF}_{\mathsf{trans}}(\mathsf{md})$
$\mathsf{trans}_{\mathsf{red}}(j) := (c_{\neg B_j}^{(j)}, \left\{\mathsf{cr}_1^{(i,j)}\right\}_{i \neq I_j, i=1}^{t}, \mathsf{cr}_2^{(j,\neg B_j)})$
Return $\left(\mathsf{md}, \left\{\mathsf{trans}_{\mathsf{red}}(j), \alpha_{I_j}, \mathsf{resp}_1^{(I_j,j)}, \mathsf{resp}_2^{(j,B_j)}\right\}_{j=1}^{r}\right)$
```

Fig. 6. SOFIA signature generation

The signer begins by effectively performing $\mathsf{KGen}()$ to obtain pk and $\mathbf{F}$, and then iterates through r rounds of the transformed identification scheme to obtain the transcript. He then uses this as input for $\mathrm{XOF}_{\mathsf{trans}}$ to derive a sequence of indices $((I_1, B_1), \ldots, (I_r, B_r))$, which effectively dictate the responses that should be included unblinded in the signature.

Verification. Upon receiving a message M, a signature σ, and a public key $\mathsf{pk} = (S_{\mathbf{F}}, \mathbf{v})$, the verifier begins by obtaining the system parameter $\mathbf{F}$ and parsing the signature σ as defined by its construction in $\mathsf{Sign}()$, above. The verification routine that follows is listed in Figure 7.

```
$\mathsf{Vf}(\mathsf{pk}, \sigma, M)$

$\mathbf{F} \leftarrow \mathrm{XOF}_{\mathbf{F}}(S_{\mathbf{F}})$
$((I_1, B_1), \ldots, (I_r, B_r)) \leftarrow \mathrm{XOF}_{\mathsf{trans}}(\mathsf{md})$
For $j \in \{1, \ldots, r\}$ do
    $\mathsf{cr}_1^{(I_j,j)} \leftarrow \mathrm{H}_1(\mathsf{resp}_1^{(I_j,j)})$
    $\mathsf{cr}_2^{(I_j,B_j)} \leftarrow \mathrm{H}_2(\mathsf{resp}_2^{(I_j,B_j)})$

For $j \in \{1, \ldots, r\}$ do
    If $B_j = 0$ then
        $\mathbf{r}_0^{(j)} := \mathsf{resp}_2^{(I_j,B_j)}$
        $c_0^{(j)} \leftarrow Com(\mathbf{r}_0^{(j)}, \alpha_{I_j}\mathbf{r}_0^{(j)} - \mathbf{t}_1^{(I_j,j)}, \alpha_{I_j}\mathbf{F}(\mathbf{r}_0^{(j)}) - \mathbf{e}_1^{(I_j,j)})$
    Else
        $\mathbf{r}_1^{(j)} := \mathsf{resp}_2^{(I_j,B_j)}$
        $c_1^{(j)} \leftarrow Com(\mathbf{r}_1^{(j)}, \alpha_{I_j}(\mathbf{v} - \mathbf{F}(\mathbf{r}_1^{(j)})) - \mathbf{G}(\mathbf{t}_1^{(I_j,j)}, \mathbf{r}_1^{(j)}) - \mathbf{e}_1^{(I_j,j)})$
$\mathsf{md}' \leftarrow \mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$
Return $\mathsf{md}' = \mathsf{md}$
```

Fig. 7. SOFIA signature verification

Optimizations. There are several optimizations that can be applied to signatures resulting from a transformed q2-IDS. Some of them are specific for SOFIA and some are more general; similar and related optimizations were suggested in [Unr15], [CHR+16] and [CDG+17].

Excluding unnecessary blindings. The signature contains blindings of all computed responses, as well as a selection of opened responses $\mathsf{resp}_1^{(I_j,j)}$ and $\mathsf{resp}_2^{(j,B_j)}$. It is redundant to include the values $\mathsf{cr}_1^{(I_j,j)}$ and $\mathsf{cr}_2^{(j,B_j)}$, as these can be recomputed based on the opened responses. This optimization was actually proposed in the generic Unruh transform [Unr15], and applies to any construction similar to Unruh's and ours. However, for the verifier to know which responses were actually opened, they must be able to reproduce the indices $((I_1, B_1), \ldots, (I_r, B_r))$, which are derived from the transcript, and without the blinded responses, this transcript is incomplete. To solve this circular dependency, we could include the selected indices in the signature. However, for typical parameters[5], we can do this more efficiently by breaking $\mathrm{XOF}_{\mathsf{trans}}$ into two parts, composing it of a hash function over the transcript $\mathcal{H}$ and a extendable output function XOF_{IB} to derive the indices from the hash output. We then include $\mathcal{H}\left(\mathsf{pk}, M, \{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}\right)$ as part of the signature, so that the verifier can reconstruct the indices, blind

[5] See Section 6.1

the corresponding responses, construct $\mathsf{trans}_{\mathsf{full}}$, and recompute the same hash for comparison.

Fixed challenge space definition. Following the generic description of the signature, the selected $\alpha^{(i,j)}$ are included in the signature. Depending on the specific choice of t and q, it may be more efficient to include the challenges $\alpha^{(i,j)}$ that were *not* selected. However, there is no reason not to take this a step further and simply fix a challenge space ChS_1 of t elements. That way, all the α's from ChS_1 will be selected and there is no need to include them in the signature. This not only reduces the signature size, but also simplifies the implementation.

Excluding unnecessary second responses. The underlying IDS from [SSH11] has a specific property, namely that the second responses do not depend on the previous state (that is, on the first challenge and response). Therefore, regardless of the value of α, the second responses are always the same. For this reason, they need to be included only once per commitment (rather than repeating the same value t times). Combined with the previous optimization, this implies that one of the second responses will be opened, and the other will be included blinded.

Omitting commitments. The check that the verifier performs for each round consists of recomputing $c^{(j)}_{B_j}$, and comparing it to one of the commits supplied by the signer. Similar to the above, and as already suggested in [SSH11], the signer can omit the commits that the verifier will recompute. A hash over all commits could be included instead, which the verifier can reconstruct using the commits $c^{(j)}_{B_j}$ he recomputes and the commits $c^{(j)}_{\neg B_j}$ the signer includes. However, it turns out that this hash is not necessary either: as these commitments are part of the transcript and the verifier is already checking the correctness of the transcript as per the first optimization, the correctness of the recomputed commitments is implicitly checked in the process.

While constructing this scheme, we attempted several other variations. Notably, we explored opening for multiple α-challenges, but that led to no improvement in the number of rounds, and, in some cases, to a contradiction of the zero-knowledge property. Variants that employ a form of internal parallelization by committing to multiple values for $\mathbf{t}_0$ do reduce the number of rounds, but increase the size of the transcript disproportionately.

Altogether, the above optimizations are crucial: they add up to almost 110 KiB, nearly halving the signature size of the scheme that results from the transform.

5.2 Security of SOFIA

In Section 3 we described an extension of Unruh's transform to q2-IDS and have proven that it provides PQ-EU-CMA security in the QROM for *any* underlying q2-IDS with a PQ-q2-extractor, the HVZK property, and PQ-key-one-wayness. This, of course, implies that this transform can immediately be applied to the 5-pass $\mathcal{MQ}$ IDS from [SSH11], to give an $\mathcal{MQ}$ signature provably secure in the QROM.

As discussed in the previous subsection, some optimizations can significantly improve the performance of the scheme. They deviate from the generic construction, however, causing a need for some changes in the security proof. Fortunately, only minor changes are required. We specify the following theorem.

Theorem 5.1. *Let $k \in \mathbb{N}$ be the security parameter. The signature scheme* SOFIA *is post-quantum existentially unforgeable under adaptive chosen message attacks in the quantum-accessible random oracle model if the following conditions are satisfied:*

- *The search version of the $\mathcal{MQ}$ problem is intractable in the average case,*
- *the hash functions $\mathcal{H}$, H_1, and H_2 are modeled as quantum-accessible random oracles,*
- *the commitment function Com is computationally binding against quantum adversaries, statistically hiding, and has $O(k)$ bits of output entropy,*
- *the pseudorandom generators, $PRG_{\mathbf{rte}}$, PRG_{sk} have outputs computationally indistinguishable from random for any polynomial-time quantum adversary.*

Proof. First let's consider a signature scheme obtained by applying the optimizations from the previous section on the signature scheme from Figures 2 and 3. We will refer to it as the optimized scheme throughout this proof. We will show that this optimized scheme is PQ-EU-CMA secure, if the underlying q2-IDS satisfies the conditions from Theorem 4.3. We will assume some additional properties of the IDS, that represent a special case of q2-IDS schemes. The optimized scheme is characterized by the following optimizations.

- We fix the challenge space ChS_1 to t elements. Note that this change does not influence the security arguments at all.
- We assume that the underlying IDS of the optimized scheme is such that the second response does not depend on the first challenge and response, but only on the second challenge and the initial output by the prover $\mathcal{P}_0$. In this case, in the signature generation, instead of calculating the second response as $\mathsf{resp}_2^{(i,j,\mathsf{ch}_2)} \leftarrow \mathcal{P}_2(\mathsf{state}^{(i,j)}, \mathsf{ch}_2)$ for every $i \in \{1, \dots, t\}$, we calculate it once per round as $\mathsf{resp}_2^{(j,\mathsf{ch}_2)} \leftarrow \mathcal{P}_2(\mathsf{state}^{(j)}, \mathsf{ch}_2)$. The full transcript is now $\{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}$, with $\mathsf{trans}_{\mathsf{full}}(j) = \mathsf{com}^{(j)}, \left\{\mathsf{ch}_1^{(i,j)}, \mathsf{cr}_1^{(i,j)}\right\}_{i=1}^{t}, (\mathsf{cr}_2^{(j,0)}, \mathsf{cr}_2^{(j,1)})$. The reduced transcript $\mathsf{trans}_{\mathsf{red}}(j)$ that is included in the signature is influenced similarly.
- Assuming that the underlying IDS is such that $\mathsf{com} = (c_0, c_1)$, we omit from the signature the commitment c_{ch_2} that the verifier recomputes, depending on the challenge ch_2. This alters the content of $\mathsf{trans}_{\mathsf{red}}(j)$ but not of $\mathsf{trans}_{\mathsf{full}}(j)$.[6]

It is straight forward to verify that Lemma 3.1 (and in the QROM, Lemma 3.5) still hold for the optimized scheme. We only removed duplicate information from

[6] If we would have used the optimization from [SSH11] directly, this would have influenced the full transcript as well.

the signature, not affecting the reductions ability to extract. The exact probability of abort in Lemma 3.2 might change as we remove some values from $\{\mathsf{trans}_{\mathsf{full}}(j)\}_{j=1}^{r}$, maybe reducing its entropy. However, the given bound only depends on the amount of entropy coming from the commitments which remains unchanged for the optimized scheme. Therefore, the claims of Lemma 3.2 remain valid.

Next, recall (cf. Theorem 4.3) that, under the assumption of intractability of the $\mathcal{MQ}$ problem on average, and assuming computationally binding and statistically hiding properties of *Com*, the 5-pass IDS from [SSH11] is PQ-KOW, is HVZK, and has a PQ-q2-Extractor. Furthermore, it satisfies the particular properties that the optimized scheme above requires. Thus applying the optimized transform on the Sakumoto-Shirai-Hiwatari 5-pass IDS scheme, we obtain a PQ-EU-CMA secure signature (cf. Theorems 3.3 and 3.6)

To complete the proof, we note that using a standard game hopping argument, it is straightforward to show that the success probability of a PQ-EU-CMA adversary against SOFIA is negligibly close to the success probability of a PQ-EU-CMA adversary against the optimized scheme from the Sakumoto-Shirai-Hiwatari 5-pass IDS scheme when the outputs of $\mathrm{PRG}_{\mathsf{rte}}$ and $\mathrm{PRG}_{\mathsf{sk}}$ are post-quantum computationally indistinguishable from random. □

6 SOFIA-4-128

Having described the scheme in general terms, we now provide concrete parameters that allow us to specify a specific instance, which we will refer to as SOFIA-4-128. We present an optimized software implementation and list the results, in particular in comparison to MQDSS-31-64. All benchmarks mentioned below were obtained on a single core of an Intel Core i7-4770K (Haswell) CPU, following the standard practice of disabling TurboBoost and hyper-threading. We compiled the code using gcc 4.9.2-10, with -O3 and -march=native.

6.1 Parameters

The previous section assumed a number of parameters and functions. Notably, we must define $\mathbb{F}_q$, the field in which we perform the arithmetic, and n and m, the number of variables and equations defining the $\mathcal{MQ}$ problem. The number of rounds r is determined by t (i.e. the number of responses $\mathsf{resp}_1^{(i,j)}$, bounded by q in SOFIA) and the targeted security level, using Theorem 3.6.

For MQDSS-31-64, the choice of $\mathbb{F}_{31}$ was motivated by the fact that it brings the soundness error close to $\frac{1}{2}$ while providing convenient characteristics for fast implementation [CHR+16]. For SOFIA-4-128, our primary focus is on optimizing for signature size while still maintaining efficiency. To do so, we compute signature sizes for a wide range of candidate systems[7], and investigate several in more detail by implementing and measuring the resulting $\mathcal{MQ}$ evaluation functions. Notably, we look at the results of $\mathcal{MQ}(128, 128, \mathbb{F}_4)$, $\mathcal{MQ}(96, 96, \mathbb{F}_7)$ and

[7] A calculation script is included in the software package.

$\mathcal{MQ}(72, 72, \mathbb{F}_{16})$, and compare to $\mathcal{MQ}(64, 64, \mathbb{F}_{31})$ from [CHR+16]. Of these, $\mathcal{MQ}(128, 128, \mathbb{F}_4)$ is the decisive winner, resulting in the smallest[8] signatures while still providing decent performance. See Table 2 on page 28 for benchmarks of single evaluation functions and the related signature sizes. Note that, as the number of rounds r does not depend on the choice of $\mathbb{F}_q$ but merely on t, the signing time scales proportionally[9].

Parameters for $\mathcal{MQ}(m, n, \mathbb{F}_q)$. A straightforward method for solving systems of m quadratic equations in n variables over $\mathbb{F}_q$ is by performing exhaustive search on all possible q^n values for the variables, and testing whether they satisfy the system. Currently, [BCC+10] provide the fastest enumeration algorithm for systems over $\mathbb{F}_2$, needing $4 \log n \cdot 2^n$ operations[10].

In addition, there exist algebraic techniques that analyze the properties of the ideal generated by the given polynomials. The most important are the algorithms from the F4/F5 family [Fau99,Fau02,BFS15,BFP12], and the variants of the XL algorithm [CKPS00,Die04,YC05,YC04]. Although different in description, the two families bear many similarities, which results in similar complexity [YCY13].

In the Boolean case, today's state of the art algorithms from the aforementioned families, BooleanSolve [BFSS13] and FXL [YC04], provide improvement over exhaustive search. In particular, the improvement is visible for polynomials with more than 200 variables (respectively, $\Theta(2^{0.792n})$ and $\Theta(2^{0.875n})$ complexity when $m = n$). A very recent algorithm, the Crossbred algorithm [JV17] over $\mathbb{F}_2$, is likely to further improve the asymptotic complexity, as the authors report that it passes the exhaustive search barrier already for 37 Boolean variables. Unfortunately, at the time of writing, the preprint does not include a detailed complexity analysis that we can use[11].

Interestingly, the current best known algorithms, BooleanSolve [BFSS13], FXL [YC04,YC05], the Crossbred algorithm [JV17] and the Hybrid approach [BFP12] all combine algebraic techniques with exhaustive search. This immediately allows for improvement in their quantum version using Grover's quantum search algorithm [Gro96], provided the cost of implementing them on a quantum computer does not diminish the gain from Grover. Unfortunately, the current literature lacks analysis of the quantum version of these algorithms. To the best of our knowledge, a detailed analysis has only been done for pure enumeration using Grover's search [WS16], showing that a system of n equations in n variables can be solved using $\Theta(n \cdot 2^{n/2})$ operations.

[8] This is also the minimum amongst all candidate systems we looked at – it is not merely beating $\mathbb{F}_7$ and $\mathbb{F}_{16}$, but also less common options such as $\mathbb{F}_5$ and $\mathbb{F}_8$.

[9] This disregards the overhead for the various hash computations, which can be considered to scale similarly.

[10] The techniques from [BCC+10] can be extended to other fields $\mathbb{F}_q$ with the same expected complexity of $\Theta(\log_q n \cdot q^n)$.

[11] The authors of [JV17] confirmed that the complexity analysis is an ongoing work, and will soon be made public.

In what follows we will analyze the complexity of the quantum versions of the Hybrid approach and BooleanSolve, and use the results as a reference point in choosing parameters for $\mathcal{MQ}(m, n, \mathbb{F}_q)$ that provide 128 bit post-quantum security[12].

First of all, we note that $m = n$ is the best choice in terms of hardness of the $\mathcal{MQ}$ problem. Indeed, if there are more equations than variables, they provide more information about the solution, so finding one becomes easier. On the other hand, if there are more variables than equations, we can simply fix $n - m$ variables and reduce the problem to a smaller one, with m variables.

Let $\mathbf{F} = (f_1, \ldots, f_m), f_i \in \mathbb{F}_q[x_1, \ldots, x_n]$. Without loss of generality, the equation system that we want to solve is $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

The main complexity in both the Hybrid approach and BooleanSolve comes from performing linear algebra on a Macalay matrix $Mac_D(\mathbf{F})$ of degree D (with rows formed by the coefficients of the monomials of uf_i of maximal degree D). The degree D should be big enough so that a Gröbner basis of the ideal generated by the polynomials can be obtained by performing linear algebra on the Macaulay matrix. The smallest such D is called the degree of regularity D_{reg}, and for semi-regular systems (which is a very plausible assumption for randomly generated polynomials) it is given by $D_{reg}(n, m) = 1 + deg(HS_q(t))$, where

$$HS_q(t) = \left[\frac{(1-t^2)^m}{(1-t)^n}\right]_+, \text{for } q > 2, \text{ and } HS_2(t) = \left[\frac{(1+t)^n}{(1+t^2)^m}\right]_+,$$

and the $_+$ subscript denotes that the series has been truncated before the first non-positive coefficient.

Since D_{reg} determines the size of the matrix, and thus the complexity of the linear algebra performed on it, both algorithms first fix k among the n variables in order to reduce the complexity of the costliest computational step. Now the linear algebra step is instead performed on $Mac_{D_{reg}}(\tilde{\mathbf{F}})$, where $\tilde{\mathbf{F}} = (\tilde{f}_1, \ldots, \tilde{f}_m)$ and $\tilde{f}_i(x_1, \ldots, x_{n-k}) = f_i(x_1, \ldots, x_{n-k}, a_{n-k+1}, \ldots, a_n)$, for some $(a_{n-k+1}, \ldots, a_n) \in \mathbb{F}_2^k$.

Given the linear algebra constant $2 \leqslant \omega \leqslant 3$, the complexity of the Hybrid approach for solving systems of n equations in n variables over $\mathbb{F}_q$ is

$$C_{Hyb}(n, k) = Guess(q, k) \cdot C_{F5}(n - k, n). \tag{6}$$

where

$$C_{F5}(n, m) = \Theta\left(\left(m\binom{n + D_{reg}(n, m) - 1}{D_{reg}(n, m)}\right)^\omega\right),$$

is the complexity of computing a Gröbner basis of a system of m equations in n variables, $m \geqslant n$, using the F5 algorithm [Fau02], $Guess(q, k) = \log_q(k)q^k$ in the classical case and $Guess(q, k) = \log_q(k)q^{k/2}$ in the quantum case using Grover's algorithm[13]. Also, k is an optimization parameter chosen such that the overall complexity is minimized.

[12] A similar analysis can be made using the algorithms from the XL family.

[13] We assume that in the quantum case the factor $\log_q(k)$ is the same as in the classical case, as opposed to the factor k from [WS16].

In the case of $\mathbb{F}_2$, the BooleanSolve algorithm performs better than the Hybrid approach. It reduces the problem to testing the consistency of a related linear system

$$\mathbf{u} \cdot Mac_{D_{reg}}(\tilde{\mathbf{F}}) = (0, \ldots, 0, 1) \tag{7}$$

If the system is consistent, then the original system does not have a solution. This allows for pruning of all the inconsistent branches corresponding to some $a \in \mathbb{F}_2^k$. A simple exhaustive search is then performed on the remaining branches. It can be shown that the running time of the algorithm is dominated by the first part of the algorithm in both the classical and the quantum version, although in the quantum case the difference is not as big as a consequence of the reduced complexity of the first part. Therefore, for simplicity, we omit the exhaustive search on the remaining branches from our analysis.

The complexity of the BooleanSolve algorithm is given by

$$C_{Bool}(n, k) = Guess(2, k) \cdot C_{cons}(Mac_{D_{reg}}(\tilde{\mathbf{F}})). \tag{8}$$

where $Guess(2, k)$ is defined the same as in the Hybrid approach, and

$$C_{cons}(Mac_{D_{reg}}(\tilde{\mathbf{F}})) = \Theta(N^2 \log^2 N \log\log N), \quad N = \sum_{i=0}^{D_{reg}(n-k,n)} \binom{n}{i}$$

is the complexity of testing consistency of the matrix (7), using the sparse linear algebra algorithm from [GLS98].

Table 1 below provides estimates of the minimum requirements for 128 bit post-quantum security of $\mathcal{MQ}(n, n, \mathbb{F}_q)$ with regard to BooleanSolve and Hybrid Approach using Grover's search, as well as plain use of Grover's search. In the estimates we used $\omega = 2.3$, which is smaller than the best known value $\omega = 2.3728639$ [Gal14]. We provide the optimal number of fixed variables in brackets, where actually this number does not equal the number of variables in the initial system. When this is the case, the optimal strategy is to simply use Grover (fix all variables), which we denote with G. Note that since any system of n variables over $\mathbb{F}_{2^s}$ can be efficiently transformed into a system of sn variables over $\mathbb{F}_2$, we have scaled the results for BooleanSolve for larger $\mathbb{F}_{2^s}$ accordingly.

As mentioned earlier, a new algebraic method for equations over $\mathbb{F}_2$, the Crossbred algorithm, was proposed very recently [JV17]. The main idea of this approach is to first perform some operations on the Macalay matrix of degree $D_{reg}(n-k, n)$ of the given system, and only afterwards fix variables. In particular, the algorithm first tries to find enough linearly independent elements in the kernel of a submatrix of $Mac_{D_{reg}(n-k,n)}$, corresponding to monomials of specialized degree in the variables that will later remain in the system (i.e. will not be fixed). These can then be used to form new polynomials in the $n-k$ remaining variables of total small degree d, which added to $Mac_d(\tilde{\mathbf{F}})$ will result in working with a much smaller Macalay matrix. The advantage here comes from using sparse linear algebra algorithms on $Mac_{D_{reg}(n-k,n)}$ for the first part and dense

	$\mathbb{F}_2$	$\mathbb{F}_3$	$\mathbb{F}_4$	$\mathbb{F}_5$	$\mathbb{F}_7$	$\mathbb{F}_8$	$\mathbb{F}_{11}$	$\mathbb{F}_{13}$	$\mathbb{F}_{16}$	$\mathbb{F}_{17}$	$\mathbb{F}_{31}$	$\mathbb{F}_{32}$
BooleanSolve	221 (200)	/	111	/	/	56	/	/	28	/	/	14
Hybrid	G	G	G	G	G	84 (57)	77 (51)	73 (43)	69 (40)	69 (40)	61 (30)	60 (21)
Grover	251	158	126	108	90	84	73	68	63	62	51	51

Table 1. Lower bound on number of variables n for 128 bit post quantum security against the quantum versions of Hybrid approach [BFP12] and BooleanSolve [BFSS13]. In brackets is the number of fixed variables. G denotes that the best strategy is to fix all variables, i.e. plain Grover search.

linear algebra only on the smaller Macalay matrix in the second part[14]. Since [JV17] does not contain a complexity analysis, we refrain from claiming exact security requirements based on the quantum version of the algorithm. Nevertheless, following the description of the algorithm we have estimated that a system of 256 variables over $\mathbb{F}_2$ (which applies to $\mathcal{MQ}(128, 128, \mathbb{F}_4)$ or $\mathcal{MQ}(64, 64, \mathbb{F}_{16})$) provides 128-bit security against the quantum version of Crossbred algorithm. We will include a more detailed analysis for the quantum version once a classical complexity analysis is available.

Number of rounds r and blinded responses per round t. The choice of t provides a trade-off between size and speed; a larger t implies a smaller error, resulting in less rounds, but more included blinded responses per round (the additional computational cost of which is insignificant). Interestingly, $t = 3$ provides the minimal size, followed by $t = 4$, and, only then, $t = 2$. The decrease in rounds quickly diminishes, however, making $t = 3$ and $t = 4$ the most attractive choices[15].

Given the above considerations (and with a prospect of some convenience of implementation), we select the parameters $n = m = 128$, $q = 4$ and $t = 3$. For a security level of 128 bits post-quantum security, it follows from Theorem 3.6 that we must select r such that $2^{-(r \log \frac{2t}{t+1})/2} < 2^{-128}$. This implies $r = 438$.

[14] An external specialization of variables is also possible, but this does not bring any improvement classically, and we have verified for some parameters that this is the case also quantumly.

[15] Note that t is naturally bounded by q, making these the *only* options for $\mathbb{F}_4$.

	cycles[b]	size $t = 3, r = 438$	size $t = 4, r = 378$
$\mathcal{MQ}(128, 128, \mathbb{F}_4)$	21 412	123.22 KiB	129.97 KiB
$\mathcal{MQ}(96, 96, \mathbb{F}_7)$	36 501	129.00 KiB[a]	136.20 KiB[a]
$\mathcal{MQ}(72, 72, \mathbb{F}_{16})$	25 014	136.91 KiB	144.73 KiB
$\mathcal{MQ}(64, 64, \mathbb{F}_{31})$	6 616[c]	149.34 KiB[a]	158.15 KiB[a]

[a] This assumes optimally packing the elements of $\mathbb{F}_q$, which may not be practical.

[b] This is the cost of a single evaluation. In practice, batching provides a speedup. See Section 6.2.

[c] As reported in [CHR+16].

Table 2. Benchmarks for varying parameter sets

Required functions. Before being able to implement the scheme, we must still define several of the functions we have assumed to exist. In particular, we need: a string commitment function *Com*; pseudo-random generators PRG_{sk} and $\text{PRG}_{\mathbf{rte}}$; extendable output functions $\text{XOF}_{\mathbf{F}}$ and XOF_{IB}; permutation functions H_1 and H_2; and a cryptographic hash function $\mathcal{H}$.

We instantiate the extendable output functions, the string commitment functions, the permutations and the hash function with SHAKE-128 [BDPV11]. This applies trivially, except for XOF_{IB}, of which the output domain is a series of ternary and binary indices (as $t = 3$). We resolve this by applying rejection sampling to the output of SHAKE-128 to derive the ternary challenges. For $\text{XOF}_{\mathbf{F}}$, we achieve a significant speedup by dividing its output in four separate pieces, generating each of them with a domain-separated call to cSHAKE-128 [BDPV11]. For the application of $\mathcal{H}$ to the public key, the message and the transcript, we note that collision resilience is achieved when the message is absorbed into the SHAKE-128 state after the transcript, as absorbing the transcript randomizes the state sufficiently to prevent internal collisions.

For $\text{PRG}_{\mathbf{rte}}$ and PRG_{sk}, we also instantiate with SHAKE-128. We note that different systems can make different choices here without breaking compatibility – this is a local decision for the signer. In fact, for the optimized Haswell implementation discussed in the next section, we instantiate $\text{PRG}_{\mathbf{rte}}$ with AES in counter mode, using the AES-NI instruction set.

6.2 Implementation

As part of this work, we provide a C reference implementation and an implementation optimized for AVX2. The focus of this section is the evaluation of the $\mathcal{MQ}$ function, given the abovementioned parameter set $\mathcal{MQ}(128, 128, \mathbb{F}_4)$. The rest of the scheme depends on fairly straight-forward operations (such as multiplying vectors of $\mathbb{F}_4$ elements by a constant scalar) and applications of existing public domain implementations of AES-CTR and SHAKE-128 (although we do briefly discuss parallel evaluation of the latter).

Before discussing the computation, we note that the chosen parameters lend themselves to a very natural data representation. Throughout the entire scheme, we interpret 256 bit vectors as vectors of 128 bitsliced $\mathbb{F}_4$ elements: the low 128 bits make up the lower bits of the two-bit elements, and the high 128 bits make up the higher bits of each element. This makes operations such as scalar multiplication in C code very convenient, as this can be easily expressed as logical operations on bit sequences, but provides an even more important benefit for AVX2 assembly code. Notably, one vector of $\mathbb{F}_4$ elements now fits exactly into one 256 bit vector register, with the lower bits fitting into the low lane and the higher bits into the high lane. Whereas other parameter sets could result in having to consider crossing the lanes, in this case the separation is very natural.

When it comes to sampling elements in $\mathbb{F}_4$ from the output of SHAKE-128 or AES-CTR, we can freely interpret the random data to be in bitsliced representation. Similarly, we simply write the elements to the signature in this representation, as signature verification enjoys precisely the same benefits. All in all, there is no point throughout the entire scheme at which we need to actually perform a bitslicing operation.

Evaluating $\mathcal{MQ}$. For a given input $\mathbf{x}$, we split the evaluation into two phases: computing all quadratic monomial terms $x_i x_j$, and composing them together to evaluate the quadratic polynomials.

Computing the quadratic terms. To perform the first step, we use a similar approach as was used in [CHR+16]. It can be viewed as a combination of their approach for $\mathbb{F}_2$ and for $\mathbb{F}_{31}$, as we now operate on a single register that contains all input elements, but effectively view each lane as 16 separate single-byte registers. Using `vpshufb` instructions, it is very cheap to arrange the elements in such a way that all multiplications can be performed using only a minimal number of rotations. We used the script from [CHR+16] as a starting point to generate the arrangement.

A bitsliced multiplication in $\mathbb{F}_4$ can be efficiently performed using only a few logical operations. The inputs to these multiplications are a register containing $\mathbf{x}$ and a register containing some rotated arrangement of $\mathbf{x}$. However, some of these operations require the low and high lanes of the vector registers to interact, which is typically costly. As $\mathbf{x}$ is constant, we can speed up these multiplications by rewriting them as shown below, and presetting two registers that contain $[\mathbf{x}_{high}|\mathbf{x}_{high}]$ and $[\mathbf{x}_{high} \oplus \mathbf{x}_{low}|\mathbf{x}_{low}]$, respectively. Note that all of these operations are not performed on single bits, but rather on 128 bit vector lanes. The multiplication of 128 elements then requires only two `vpand` instructions, one `vperm` instruction, and a `vpxor` to combine the results.

$$\begin{aligned} c_{high} &= (a_{high} \wedge (b_{low} \oplus b_{high})) \oplus (a_{low} \wedge b_{high}) \\ c_{low} &= (a_{low} \wedge b_{low}) \oplus (a_{high} \wedge b_{high}) \end{aligned}$$

Multiplying, and accumulating results. We focus on two approaches to perform the second and most costly part of the evaluation, in which all of the above monomials need to be multiplied with coefficients from $\mathbf{F}$ and summed into the output vector. They are best described as iterating either 'horizontally' or 'vertically' through the required multiplications. For the vertical approach, we iterate over all[16] registers of monomials, broadcasting each of the monomials to each of the 128 possible positions by shuffling bytewise and applying eight bit-rotations before multiplying with a sequence of coefficients from $\mathbf{F}$ and adding into an accumulator. Alternatively, we iterate over the output elements in the outermost loop. For each output element, we iterate over all registers of monomials, perform the multiplications and horizontally sum the results by making use of the `popcnt` instruction.

Intuitively, the latter approach may seem like more work (notably because it requires more loads from memory), but in practice it turns out to be faster for our parameters. The main reason for this is that by maintaining multiple separate accumulators, loaded monomials can be re-used while still maintaining chains of logic operations that operate on independent results (as the accumulators are only joined together later), which leads to highly efficient scheduling.

For both cases, a lot of computational effort can be saved by delaying part of the multiplication in $\mathbb{F}_4$. This is done by computing both $[\hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{low}]$ and $[\hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{low}]$, with $\mathbf{f}$ from $\mathbf{F}$ and $\hat{\mathbf{x}}$ a sequence of quadratic monomials, and accumulating these results separately. After accumulating, these values can be used as inputs to finish all multiplications and reductions at once, eliminating the majority of the logic operations that would otherwise be performed for each of the 65 multiplications.

Evaluating $\mathcal{MQ}$ instances in parallel. While repeatedly iterating over the monomials can be done quickly (especially since they easily fit in L1 data cache), each of the coefficients in $\mathbf{F}$ is used only once, making loading these a considerable burden. As $\mathbf{F}$ is constant for each evaluation, however, a significant speedup can be achieved by processing multiple instances of the $\mathcal{MQ}$ function in parallel. In particular the vertical approach lends itself nicely for this, as its critical section leaves some registers unused. For the horizontal approach, it becomes a trade-off with registers that are used for parallel accumulators, but, as loading of $\mathbf{F}$ is an important bottleneck, there is still significant gain from parallelizing multiple instances.

For SOFIA-4-128, the signer evaluates $r = 438$ instances of $\mathbf{F}$ and its polar form $\mathbf{G}$ on completely independent inputs, which can be trivially batched. The verifier performs 438 evaluations of $\mathbf{F}$, and on average half as many evaluations of $\mathbf{G}$, which can also be batched together.

[16] There are $\frac{n \cdot (n+1)}{2} = 8256$ such monomials, which results in $64\frac{1}{2}$ 256-bit sequences. We round up to 65 by zeroing out half of the high and half of the low lane. To still get results that are compatible with implementations on other platforms, we create similar gaps in the stream of random values used to construct $\mathbf{F}$, ensuring that the same random elements are still used for the same coefficients.

Parallel SHAKE-128 and cSHAKE-128. As will be apparent in the next section, many cycles are spent computing the Keccak permutation (as part of either SHAKE-128 or cSHAKE-128). Some of the main culprits[17] are the commitments, the blinding of responses and the expansion of **F**. While the Keccak permutation does not lend itself nicely to internal parallelization, it is trivially possible to compute four instances in parallel in a 256 bit vector register. This allows us to seriously speed up the computations of the many commits and blindings, as these are all fully independent and can be grouped together across rounds. Deriving **F** can be parallelized by splitting it in four domain-separated cSHAKE-128 calls operating on the same seed, as was alluded to in Section 6.1.

Benchmarks. All things considered, evaluating the $\mathcal{MQ}$ function horizontally in batches of three turns out to give the fastest results, measuring in at 17 558 cycles per evaluation. Evaluating vertically is slightly more expensive, at 18 598 cycles per evaluation. The cost for evaluating the polar form is not significantly different, differing by approximately a hundred cycles from regular $\mathcal{MQ}$. Generating the monomial terms $x_iy_j + x_jy_i$ is somewhat more costly, but this is countered by the fact that the linear terms cancel out.

To generate a signature, we spend 21 305 472 cycles. Of this, 15 420 520 cycles can be attributed to evaluating $\mathcal{MQ}$, and 43 954 to AES-CTR. The remainder is almost entirely accounted for by the various calls to SHAKE-128 and cSHAKE-128 for the commitments, blindings and randomness expansion. In particular, expanding **F** costs 1 120 782 cycles. Note, however, that if many signatures are to be generated, this expansion only needs to be done once and **F** can be kept in memory across subsequent signatures.

Verification costs 15 492 686 cycles, and key generation costs 1 157 112. Remark that key generation is dominated by expansion of **F**. Systems that require often re-keying may opt to reconsider the diversification of **F**, and instead re-use the same random system in a similar way as batch signature generation can. This would asymptotically reduce the average cost of key generation all the way down to essentially a single $\mathcal{MQ}$ evaluation, which costs only 21 412 cycles.

The keys of SOFIA-4-128 are very small by nature, with the secret key consisting of only a single 32 byte seed, and the 64 byte public key being made up of a seed and a single $\mathcal{MQ}$ output.

The natural candidate for comparison is MQDSS-31-64 [CHR+16], the de facto predecessor of this work. While MQDSS has a proof in the ROM, we focus further comparison on post-quantum schemes that have proofs in the QROM or standard model. See Table 3, below; as mentioned in the introduction, we include SPHINCS-256 [BHH+15] (standard model), Picnic-10-38 [CDG+17] (QROM) and TESLA-768 [ABBD15] (QROM).

[17] Hashing the transcript in parallel is a bit more involved. While it could be done using constructions like ParallelHash [BDPV11], we omit this for the sake of simplicity.

	$\|\sigma\|$	$\|\mathrm{pk}\|$, $\|\mathrm{sk}\|$	keygen	signing	verification
	(bytes)	(bytes)	(cycles)	(cycles)	(cycles)
SOFIA-4-128 [a]	126 176	64 32	1 157 112	21 305 472	15 492 686
MQDSS-31-64[a]	40 952	72 64	1 826 612	8 510 616	5 752 612
SPHINCS-256[b]	41 000	1056 1088	3 237 260	51 636 372	1 451 004
Picnic-10-38[c,d]	195 458	64 32	≈36 000	≈112 716k	≈58 680 000
TESLA-768[a]	2 336	4128k 3216k	≈172 800m[e]	2 232 906	863 790

[a] Benchmarked on an Intel Core-i7-4770K (Haswell).

[b] Benchmarked on an Intel Xeon E3-1275 (Haswell).

[c] Benchmarked on an Intel Core-i7-4790 (Haswell).

[d] Converted from milliseconds at 3.6GHz.

[e] Using a measurement from [CDG+17], as [ABBD15] does not report key generation.

Table 3. Benchmark overview

References

ABBD15. Erdem Alkim, Nina Bindel, Johannes Buchmann, and Özgür Dagdelen. TESLA: tightly-secure efficient signatures from standard lattices. Cryptology ePrint Archive, Report 2015/755, 2015. http://eprint.iacr.org/2015/755/, version 20161117:055833.

ARU14. Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *FOCS 2014*, pages 474–483, 2014. http://eprint.iacr.org/2014/296.

BCC+10. Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225 of *LNCS*, pages 203–218. Springer, 2010. https://eprint.iacr.org/2010/313.

BDPV11. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak reference, 2011. http://keccak.noekeon.org/.

BFP12. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In Joris van der Hoeven and Mark van Hoeij, editors, *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation – ISSAC '12*, pages 67–74. ACM, 2012. https://hal.inria.fr/hal-00776070/document.

BFS15. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, 70:49–70, 2015. https://hal.inria.fr/hal-01064519/document.

BFSS13. Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, 29(1):53–75, 2013. www-polsys.lip6.fr/~jcf/Papers/BFSS12.pdf.

BHH$^+$15. Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In Marc Fischlin and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 368–397. Springer, 2015. http://cryptojedi.org/papers/#sphincs.

CDG$^+$17. Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. Cryptology ePrint Archive, Report 2017/279, 2017. http://eprint.iacr.org/2017/279/.

CHR$^+$16. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, volume 10032 of *LNCS*, pages 135–165. Springer, 2016. http://eprint.iacr.org/2016/708.

CKPS00. Nicolas Courtois, Er Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, 2000. www.iacr.org/archive/eurocrypt2000/1807/18070398-new.pdf.

Cou01. Nicolas T. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 402–421. Springer, 2001. https://eprint.iacr.org/2001/058.

DFG13. Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The Fiat–Shamir transformation in a quantum world. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 62–81. Springer, 2013. https://eprint.iacr.org/2013/245.

DGV$^+$16. Özgür Dagdelen, David Galindo, Pascal Véron, Sidi Mohamed El Yousfi Alaoui, and Pierre-Louis Cayrel. Extended security arguments for signature schemes. *Designs, Codes and Cryptography*, 78(2):441–461, 2016.

DHYC06. Jintai Ding, Lei Hu, Bo-Yin Yang, and Jiun-Ming Chen. Note on design criteria for rainbow-type multivariates. Cryptology ePrint Archive, Report 2006/307, 2006. https://eprint.iacr.org/2006/307.

Die04. Claus Diem. The XL-algorithm and a conjecture from commutative algebra. In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 323–337. Springer, 2004. https://www.iacr.org/archive/asiacrypt2004/33290320/33290320.pdf.

Fau99. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139:61–88, 1999. http://www-polsys.lip6.fr/~jcf/Papers/F99a.pdf.

Fau02. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation – ISSAC '02*, pages 75–83. ACM, 2002. http://www-polsys.lip6.fr/~jcf/Papers/F02a.pdf.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *Advances in Cryptology*

– *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, 2005. https://www.iacr.org/archive/crypto2005/36210148/36210148.pdf.

FLP08. Jean-Charles Faugère, Françoise Levy-dit-Vehel, and Ludovic Perret. Cryptanalysis of MinRank. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *LNCS*, pages 280–296. Springer, 2008. http://www-polsys.lip6.fr/~jcf/Papers/crypto08.pdf.

Gal14. François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation – ISSAC '14*, pages 296–303. ACM, 2014. https://arxiv.org/pdf/1401.7714.pdf.

GJ79. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

GLS98. Mark Giesbrecht, A. Lobo, and B. David Saunders. Certifying inconsistency of sparse linear systems. In *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation – ISSAC '98*, pages 113–119, 1998. https://cs.uwaterloo.ca/~mwg/files/incons.pdf.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Digital%20Signatures/A_Digital_Signature_Scheme_Secure_Against_Adaptive_Chosen-Message_Attack.pdf.

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing – STOC '96*, pages 212–219. ACM, 1996. https://arxiv.org/pdf/quant-ph/9605043v3.pdf.

JV17. Antoine Joux and Vanessa Vitse. A crossbred algorithm for solving boolean polynomial systems. Cryptology ePrint Archive, Report 2017/372, 2017. http://eprint.iacr.org/2017/372.

Pat96. Jacques Patarin. Hidden field equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Ueli Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *LNCS*, pages 33–48. Springer, 1996. http://www.minrank.org/hfe.pdf.

PS96. David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *LNCS*, pages 387–398. Springer, 1996. https://www.di.ens.fr/~pointche/Documents/Papers/1996_eurocrypt.pdf.

SSH11. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *LNCS*, pages 706–723. Springer, 2011. https://www.iacr.org/archive/crypto2011/68410703/68410703.pdf.

Tho13. Enrico Thomae. *About the Security of Multivariate Quadratic Public Key Schemes*. PhD thesis, Ruhr-University Bochum, Germany, 2013. https://www.iacr.org/phds/116_EnricoThomae_AboutSecurityMultivariateQuadr.pdf.

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 755–784. Springer, 2015. http://eprint.iacr.org/2014/587.

WS16. Bas Westerbaan and Peter Schwabe. Solving binary $\mathcal{MQ}$ with grover's algorithm. In Claude Carlet, Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Advanced Cryptography Engineering*, volume 10076 of *LNCS*. Springer, 2016. Document ID: 40eb0e1841618b99ae343ffa073d6c1e, http://cryptojedi.org/papers/#mqgrover.

YC04. Bo-Yin Yang and Jiun-Ming Chen. Theoretical analysis of XL over small fields. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, volume 3108 of *LNCS*, pages 277–288. Springer, 2004. http://www.iis.sinica.edu.tw/papers/byyang/2386-F.pdf.

YC05. Bo-Yin Yang and Jiun-Ming Chen. All in the XL family: Theory and practice. In Choon sik Park and Seongtaek Chee, editors, *Information Security and Cryptology – ICISC 2004*, pages 67–86. Springer, 2005. http://by.iis.sinica.edu.tw/by-publ/recent/xxl.pdf.

YCY13. Jenny Yuan-Chun Yeh, Chen-Mou Cheng, and Bo-Yin Yang. Operating Degrees for XL vs. F4/F5 for Generic $\mathcal{MQ}$ with Number of Equations Linear in That of Variables. In Marc Fischlin and Stefan Katzenbeisser, editors, *Number Theory and Cryptography: Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday*, pages 19–33. Springer, 2013. http://www.iis.sinica.edu.tw/papers/byyang/17377-F.pdf.

Zha16. Mark Zhandry. A note on quantum-secure PRPs. Cryptology ePrint Archive, Report 2016/1076, 2016. http://eprint.iacr.org/2016/1076.