From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures

Ming-Shing Chen[1] and Andreas Hülsing[2] and Joost Rijneveld[3] and Simona Samardjiska[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. This paper presents MQDSS, the first signature scheme with a security reduction based on the problem of solving a multivariate system of quadratic equations ($\mathcal{MQ}$ problem). In order to construct this scheme we give a new security reduction for the Fiat-Shamir transform from a large class of 5-pass identification schemes and show that a previous attempt from the literature to obtain such a proof does not achieve the desired goal. We give concrete parameters for MQDSS and provide a detailed security analysis showing that the resulting instantiation MQDSS-31-64 achieves 128 bits of post-quantum security. Finally, we describe an optimized implementation of MQDSS-31-64 for recent Intel processors with full protection against timing attacks and report benchmarks of this implementation.

Keywords: post-quantum cryptography, Fiat-Shamir, 5-pass identification scheme, vectorized implementation.

1 Introduction

Already since 1997, when Shor published a polynomial-time quantum algorithm for factoring and discrete logarithms, it is known that an attacker equipped with a sufficiently large quantum computer will be able to break essentially all public-key cryptography in use today. More recently, various statements by

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: 36edf88b815b75e85fae8684c05ec336. Date: December 1, 2016

physicists and quantum engineers indicate that they may be able to build such a large quantum computer within the next few decades. For example, IBM's Mark Ketchen said in 2012 *"I'm thinking like it's 15 [years] or a little more. It's within reach. It's within our lifetime. It's going to happen."*. In May this year, IBM gave access to their 5-qubit quantum computer to researchers and announced that they are expecting to scale up to 50–100 qubits within one decade [IBM16].

It is still a matter of debate *when* and even *if* we will see a large quantum computer that can efficiently break, for example, RSA-4096 or 256-bit elliptic-curve crypto. However, it becomes more and more clear that cryptography aiming at long-term security can no longer discard the *possibility* of attacks by large quantum computers in the foreseeable future. Consequently, NSA recently updated their Suite B to explicitly emphasize the importance of a migration to *post-quantum* algorithms [NSA] and NIST announced a call for submissions to a post-quantum competition [NIS16]. Submissions to this competition will be accepted for post-quantum public-key encryption, key exchange, and digital signature. The results presented in this paper fall into the last of these three categories: post-quantum digital signature schemes.

Most experts agree that the most conservative choice for post-quantum signatures are hash-based signatures with tight reductions in the standard model to properties like second-preimage resistance of an underlying cryptographic hash function. Unfortunately, the most efficient hash-based schemes are stateful, a property that makes their use prohibitive in many scenarios [MKF+16]. A reasonably efficient stateless construction called SPHINCS was presented at Eurocrypt 2015 [BHH+15]; however, eliminating the state in this scheme comes at the cost of decreased speed and increased signature size.

The second direction of research for post-quantum signatures are lattice-based schemes. Various schemes have been proposed with different security and performance properties. The best performance is achieved by BLISS [DDLL13] (improved in [Duc14]) whose security reduction relies on the hardness of R-SIS and NTRU, and is non-tight. Furthermore, the performance is achieved at the cost of being vulnerable against cache-attacks as demonstrated in [GHLY16]. A more conservative approach is the signature scheme proposed by Bai and Galbraith in [BG14] with improvements to performance and security in [DBG+15], [ABBD15] and [ABB+16]. The security reduction to LWE in [ABBD15] is tight; a variant using the (more efficient) ideal-lattice setting was presented in [ABB+16]. However, these schemes either come with enormous key and signature sizes (e.g. sizes in [ABBD15] are in the order of megabytes), or sizes are reduced at the cost of switching to assumptions on lattices with additional structure like NTRU, Ring-SIS, or Ring-LWE.

The third large class of post-quantum signature algorithms is based on the hardness of solving large systems of multivariate quadratic equations, the so-called $\mathcal{MQ}$ problem. For random instances this problem is NP-complete [GJ79]. However, all schemes in this class that have been proposed with actual parameters for practical use share two properties that often raise concerns about their security: First, their security arguments are rather ad-hoc; there is no reduc-

tion to the hardness of $\mathcal{MQ}$. The reason for this is the second property, namely that these systems require a hidden structure in the system of equations; this implies that their security inherently also relies on the hardness of the so-called isomorphism-of-polynomials (IP) problem [Pat96] (or, more precisely, the Extended IP problem [DHYC06] or the similar IP with partial knowledge [Tho13] problem). Time has shown that IP in many of the proposed schemes actually relies on the MinRank problem [Cou01,FLP08], and unfortunately, more than often, on an easy instance of this problem. Therefore, many proposed schemes have been broken not by targeting $\mathcal{MQ}$, but by targeting IP (and thus exploiting the structure in the system of equations). Examples of broken schemes include Oil-and-Vinegar [Pat97] (broken in [KS98]), SFLASH [CGP] (broken in [DFSS07]), MQQ-Sig [GØJ+11] (broken in [FGP+15]), (Enhanced) TTS [YCC04a,YC05b] (broken in [TW12]), and Enhanced STS [TGTF10] (broken in [TW12]). There are essentially only two proposals from the "$\mathcal{MQ}$ +IP" class of schemes that are still standing: HFEv$^-$ variants [PCG01,PCY+15] and Unbalanced Oil-and-Vinegar (UOV)variants [KPG99,DS05]. The literature does not, to the best of our knowledge, describe any instantiation of those schemes with parameters that achieve a conservative *post-quantum* security level.

Contributions of this paper. Obviously what one would want in the realm of $\mathcal{MQ}$-based signatures is a scheme that has a tight reduction to $\mathcal{MQ}$ in the quantum-random-oracle model (QROM) or even better in the standard model, and has small key and signatures sizes and fast signing and verification algorithms when instantiated with parameters that offer 128 bits of post-quantum security. In this paper we make a major step towards such a scheme. Specifically, we present a signature system with a reduction to $\mathcal{MQ}$, a set of parameters that achieves 128 bits of post-quantum security according to our careful post-quantum security analysis, and an optimized implementation of this scheme.

This does not mean that our proposal is going quite all the way to the desired scheme sketched above: our reduction is non-tight and in the ROM. Furthermore, at the 128-bit post-quantum security level, the signature size is 40 952 bytes, which is comparable to SPHINCS [BHH+15], but larger than what lattice-based schemes or $\mathcal{MQ}$ +IP schemes achieve. However, the scheme excels in key sizes: it needs only 72 bytes for public keys and 64 bytes for private keys.

The basic idea of our construction is to apply a Fiat-Shamir transform to the $\mathcal{MQ}$-based 5-pass identification scheme (IDS) that was presented by Sakumoto, Shirai, and Hiwatari at Crypto 2011 [SSH11]. In principle, this idea is not new; it already appeared in a 2012 paper by El Yousfi Alaoui, Dagdelen, Véron, Galindo, and Cayrel [EDV+12]. In their paper they use the 5-pass IDS from [SSH11] as one example of a scheme with a property they call "n-soundness". According to their proof in the ROM, this property of an IDS guarantees that it can be used in a Fiat-Shamir transform to obtain an existentially unforgeable signature scheme. They give such a transform using the IDS from [SSH11, Section 4.2].

One might think that choosing suitable parameters for precisely this transform (and implementing the scheme with those parameters) produces the results we are advertising in this paper. However, we show that not only is the construc-

tion from [EDV+12, Section 4.2] insecure (because it ignores the requirement of an exponentially large challenge space), but also that the proof based on the n-soundness property does not apply to a corrected Fiat-Shamir transform of the 5-pass IDS from [SSH11]. The reason is that the n-soundness property does not hold for this IDS. More than that, we show that any $(2n+1)$-pass scheme for which the n-soundness property holds can trivially be transformed into a 3-pass scheme. This observation essentially renders the results of [EDV+12] vacuous, as the declared contribution of that paper is to present *"the first transformation which gives generic security statements for SS derived from* $(2n+1)$*-pass IS"*.

To solve these issues, we present a new proof in the ROM for Fiat-Shamir transforms of a large class of 5-pass IDS, including the 5-pass scheme from [SSH11]. This proof is of independent interest; it applies also, for example, to the IDS from [CVE10], [Ste93] and (with minor modifications) to [PP03]. Equipped with this result, we fix the signature scheme from [EDV+12] and instantiate the scheme with parameters for the 128-bit post-quantum security level. We call this signature scheme MQDSS and the concrete instatiation with the proposed parameters MQDSS-31-64. Our optimized implementation of MQDSS-31-64 for Intel Haswell processors takes 8 510 616 cycles for signing and 5 752 612 cycles for verification; key generation takes 1 826 612 cycles. These cycle counts include full protection against timing attacks.

A missed reference. After finishing this work, we were made aware that the authors of [EDV+12] published an updated journal version of their paper [DGV+16]. In this updated version, the authors give a new definition of n-soundness, adapt their forking lemma, and fix the presented signature scheme constructions to respect the requirement of exponentially large challenge spaces. However, it turns out that even the updated proof in [DGV+16] does not cover security of the proposed $\mathcal{MQ}$-based signature scheme (and neither of the code-based signature scheme proposed in the same paper). Nevertheless, the signature schemes proposed in [DGV+16] can be proven secure using our results without any modifications.

More specifically, the authors of [DGV+16] give a new definition of the n-soundness notion that depends on n, the number of communication steps of the IDS underlying a generic signature scheme. The new n-soundness notion requires that it is possible to extract the secret key of a generic $2n+1$-signature scheme, if 2^n valid signatures are given that follow a certain pattern. Generic $2n+1$-signature schemes are signature schemes obtained from $2n+1$-pass identification schemes using a generalized Fiat-Shamir transform. Then they adapt their forking lemma accordingly, showing that one can indeed obtain 2^n such signatures from rewinding an adversary against the signature scheme.

Now, one might think our new forking lemma is just a special instance of this very generic forking lemma. However, our forking lemma goes one step further and considers the problem that all 5-pass IDS that we are aware of (including the IDS considered in [DGV+16]) actually come with a significant soundness error. This requires to turn the original IDS into a new IDS with negligible soundness error by parallel execution of many rounds of the IDS before it is transformed

into a signature scheme. Now, an integral part of our proof is to show that one can obtain 2^n (in our case 4) transcripts of the original IDS, following a given pattern on the challenges, from the 2^n signatures (See Section 4 for more details). The authors of [DGV+16] treat this point as part of proving that a specific generic signature scheme fulfills their new n-soundness notion. However, their argument in both proofs (for the code-based and the $\mathcal{MQ}$-based scheme) is flawed. Namely, they argue that one could consider the transformation of a single instance of the IDS into a generic signature scheme and then use their forking lemma on this instance. This is incorrect as in this case the challenge spaces are not exponentially large which is a necessary requirement for the application of the forking lemma. Consequently, their specific signature scheme constructions are not covered by their security reduction.

Organization of this paper. We start with some preliminaries in Section 2. In Section 3, we recall the 5-pass IDS as introduced in [SSH11]. We present our theoretical results in Section 4: We discuss the problems with the result from [EDV+12] and resolve them by providing a new proof. We present a description of the transformed 5-pass signature scheme and give a security reduction for it in Section 5. In Section 6 we finally present a concrete instantiation and implementation thereof.

Availability of the software. We place all software described in this paper into the public domain to maximize reusability of our results. The software is available online at `https://joostrijneveld.nl/papers/mqdss`.

Acknowledgements. The authors would like to thank Marc Fischlin for helpful discussions, the anonymous reviewers for valuable comments, Wen-Ding Li for his contributions to the software, and Arno Mittelbach for the cryptocode package.

2 Preliminaries

In the following we provide basic definitions used throughout this work.

Digital signatures. The main target of this work are digital signature schemes. These are defined as follows.

Definition 2.1 (Digital signature scheme). *A digital signature scheme* Dss *is a triplet of polynomial time algorithms* $\mathit{Dss} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ *defined as:*

- *The key generation algorithm* KGen *is a probabilistic algorithm that on input* 1^k*, where* k *is a security parameter, 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.*

For correctness of a Dss, we require that for all $k \in \mathbb{N}$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^k)$, 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. By $\mathsf{Dss}(1^k)$ we denote a signature scheme with security parameter k.

Experiment $\mathsf{Exp}^{\mathsf{eu\text{-}cma}}_{\mathsf{Dss}(1^k)}(\mathcal{A})$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^k)$
$(M^\star, \sigma^\star) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$
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 adversary has only negligible success probability:

Definition 2.2 (EU-CMA). *Let $k \in \mathbb{N}$, Dss a digital signature scheme as defined above. We call Dss EU-CMA-secure if for all $Q_s, t = \mathrm{poly}(k)$ the maximum success probability $\mathrm{InSec}^{\mathsf{eu\text{-}cma}}\left(\mathsf{Dss}(1^k); t, Q_s\right)$ of all possibly probabilistic classical adversaries $\mathcal{A}$ running in time $\leq t$, making at most Q_s queries to Sign in the above experiment, is negligible in k:*

$$\mathrm{InSec}^{\mathsf{eu\text{-}cma}}\left(\mathsf{Dss}(1^k); t, Q_s\right) \stackrel{def}{=} \max_{\mathcal{A}}\{\mathrm{Succ}^{\mathsf{eu\text{-}cma}}_{\mathsf{Dss}(1^k)}(\mathcal{A})\} = \mathrm{negl}(k).$$

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

Definition 2.3 (Identification scheme). *An identification scheme consists of three probabilistic, polynomial-time algorithms $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ such that:*

- *the key generation algorithm KGen is a probabilistic algorithm that on input 1^k, where k is a security parameter, 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".*

For correctness of the scheme we require that for all $k \in \mathbb{N}$ and all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^k)$ 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 only concerned with passively secure identification schemes. We define security in terms of two properties: soundness and honest-verifier zero-knowledge.

Definition 2.4 (Soundness (with soundness error κ)). *Let $k \in \mathbb{N}$, $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ an identification scheme. We say that IDS is sound with soundness error κ if for every PPT adversary $\mathcal{A}$,*

$$\mathsf{Pr}\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^k) \\ \langle \mathcal{A}(1^k, \mathsf{pk}), \mathcal{V}(\mathsf{pk})\rangle = 1\end{matrix}\right] \leq \kappa + \mathrm{negl}(k)\,.$$

Of course, the goal is to obtain an IDS with negligible soundness error. This can be achieved by running r rounds of the protocol for an r that fulfills $\kappa^r = \mathrm{negl}(k)$.

For the following definition we need the notion of a transcript. A transcript of an execution of an identification scheme IDS refers to all the messages exchanged between $\mathcal{P}$ and $\mathcal{V}$ and is denoted by $\mathsf{trans}(\langle \mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk})\rangle)$.

Definition 2.5 ((statistical) Honest-verifier zero-knowledge). *Let $k \in \mathbb{N}$, $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ an identification scheme. 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:*

$$\{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^k) : (\mathsf{sk}, \mathsf{pk}, \mathsf{trans}(\langle \mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk})\rangle))\}$$
$$\{(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^k) : (\mathsf{sk}, \mathsf{pk}, \mathcal{S}(\mathsf{pk}))\}\,.$$

3 Sakumoto et al. 5-pass IDS scheme

In [SSH11], Sakumoto et al. proposed two new identification schemes, a 3-pass and a 5-pass IDS, based on the intractability of the $\mathcal{MQ}$ problem. They showed that assuming existence of a non-interactive commitment scheme that is statistically hiding and computationally binding, their schemes are statistical zero knowledge and argument of knowledge, respectively. They further showed that the parallel composition of their protocols is secure against impersonation under passive attack. Let us quickly recall the basics of the construction.

Let $\mathbf{x} = (x_1, \ldots, x_n)$ and let $\mathcal{MQ}(n, m, \mathbb{F}_q)$ denote the family of vectorial functions $\mathbf{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ of degree 2 over $\mathbb{F}_q$: $\mathcal{MQ}(n, m, \mathbb{F}_q) = \{\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) | f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i, s \in \{1, \ldots, m\}\}$. The function $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$ is called the polar form of the function $\mathbf{F}$. The $\mathcal{MQ}$ problem $\mathcal{MQ}(\mathbf{F}, \mathbf{v})$ is defined as follows:

Given $\mathbf{v} \in \mathbb{F}_q^m$ find, if any, $\mathbf{s} \in \mathbb{F}_q^n$ such that $\boldsymbol{F}(\mathbf{s}) = \mathbf{v}$.

The decisional version of this problem is NP-complete [GJ79]. It is widely believed that the $\mathcal{MQ}$ problem is intractable, i.e., that there exists no PPT adversary 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.

The novelty of the approach of Sakumoto et al. [SSH11] is that 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]. The key for this is the introduction of a technique to split the secret using the polar form $\mathbf{G}(\mathbf{x}, \mathbf{y})$ of a system of polynomials $\mathbf{F}(\mathbf{x})$.

In essence, with their technique, the secret $\mathbf{s}$ is split into $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, and the public $\mathbf{v} = \mathbf{F}(\mathbf{s})$ can be represented as $\mathbf{v} = \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1) + \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1)$. In order for the polar form not to depend on both shares of the secret, $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ are further split as $\alpha\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{t}_1$ and $\alpha\mathbf{F}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. Now, due to the linearity of the polar form it holds that $\alpha\mathbf{v} = (\mathbf{e}_1 + \alpha\mathbf{F}(\mathbf{r}_1) + \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1)) + (\mathbf{e}_0 + \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1))$, and from only one of the two summands, represented by $(\mathbf{r}_1, \mathbf{t}_1, \mathbf{e}_1)$ and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, nothing can be learned about the secret $\mathbf{s}$. The 5-pass IDS is given in Figure 1 where $(\mathsf{pk}, \mathsf{sk}) = (\mathbf{v}, \mathbf{s}) \leftarrow \mathsf{KGen}(\mathbf{1}^{\mathbf{k}})$.

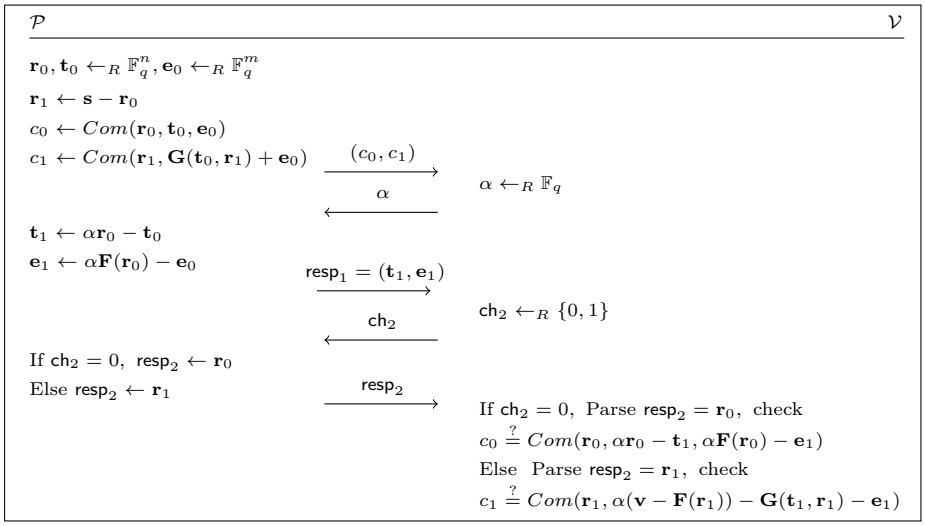

Fig. 1. Sakumoto et al. 5-pass IDS

Sakumoto et al. [SSH11] proved that their 5-pass scheme is statistically zero knowledge when the commitment scheme Com is statistically hiding which implies (honest-verifier) zero knowledge. Here we prove the soundness property of the scheme[5].

[5] Sakumoto et al. [SSH11] also sketched a proof that their 5-pass protocol is argument of knowledge when Com is computationally binding. Our security arguments rely on the weaker notion of soundness, therefore we include an appropriate proof.

Theorem 3.1. *The 5-pass identification scheme of Sakumoto et al. [SSH11] is sound with soundness error* $\frac{1}{2} + \frac{1}{2q}$ *when the commitment scheme Com is computationally binding.*

Proof. Before we prove the theorem statement, we first present a lower bound on the cheating probability as it helps to understand the scheme. We show that there exists an adversary $\mathcal{C}$, the cheater, that can cheat with probability $\frac{1}{2} + \frac{1}{2q}$. This cheater $\mathcal{C}$ simply follows the protocol using some random $\mathbf{s}'$ with a little difference: It guesses α and manipulates the second part of the commitment (c_0, c_1). The reason this works is that for $\mathsf{ch}_2 = 0$, nothing is checked using the verification key $\mathbf{v}$. Hence, in this case the cheater can always win. This already gives a success probability of $1/2$. Furthermore, manipulating c_1 does not influence the success probability in the case where $\mathsf{ch}_2 = 0$, but may increase the success probability in case $\mathsf{ch}_2 = 1$.

More formally, for the public $\mathsf{pk} = \mathbf{v}$, the cheater $\mathcal{C}$ chooses $\alpha^* \in \mathbb{F}_q$ as a prediction of what the verifier $\mathcal{V}$ will use in the protocol later on. Then $\mathcal{C}$ chooses $\mathbf{s}', \mathbf{r}_0, \mathbf{t}_0 \in \mathbb{F}_q^n$, $\mathbf{e}_0 \in \mathbb{F}_q^m$ at random, and computes $\mathbf{r}_1 \leftarrow \mathbf{s}' - \mathbf{r}_0$, and $\mathbf{t}_1 \leftarrow \alpha^* \mathbf{r}_0 - \mathbf{t}_0$. Now $\mathcal{C}$ computes the commitment (c_0, c_1) as:

- $c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$,
- $c_1 \leftarrow Com(\mathbf{r}_1, \alpha^*(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \alpha^* \mathbf{F}(\mathbf{r}_0) + \mathbf{e}_0)$.

and computes $\mathbf{e}_1 \leftarrow \alpha^* \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$. Then it follows the protocol as is.

Now, when $\mathsf{ch}_2 = 0$, $\mathcal{C}$ always wins, regardless of what α the verifier has chosen. When $\mathsf{ch}_2 = 1$, $\mathcal{C}$ wins when $\alpha = \alpha^*$, i.e. the first challenge was correctly guessed, which happens with probability $1/q$.

So, $\frac{1}{2} + \frac{1}{2q}$ is a lower bound on the success probability of an adversary. What we want to show now is that there cannot exist a cheater that wins with significantly higher success probability as long as the $\mathcal{MQ}$ problem is hard and the used commitment is computationally binding.

Towards a contradiction, suppose there exists a malicious PPT cheater $\mathcal{C}$ such that it holds that

$$\epsilon := \mathsf{Pr}[\langle \mathcal{C}(1^k, \mathbf{v}), \mathcal{V}(\mathbf{v}) \rangle = 1] - (\frac{1}{2} + \frac{1}{2q}) = \frac{1}{P(k)}.$$

for some polynomial function $P(k)$. We show that this implies that there exists a PPT adversary $\mathcal{A}$ with access to $\mathcal{C}$ that can either break the binding property of Com or can solve the $\mathcal{MQ}$ problem $\mathcal{MQ}(\mathbf{F}, \mathbf{v})$.

$\mathcal{A}$ can achieve this if she can obtain four accepting transcripts from $\mathcal{C}$ with same internal random tape, equation system $\mathbf{F}$, and public key $\mathbf{v}$, such that for two different α there are two transcripts for each α with different ch_2. This is done by rewinding $\mathcal{C}$ and feeding it with all possible combinations of $\alpha \in [0, q-1]$ and $ch_2 \in \{0, 1\}$. That way we obtain $2q$ different transcripts. Now, per assumption $\mathcal{C}$ produces an accepting transcript with probability $\frac{1}{2} + \frac{1}{2q} + \epsilon$. Hence, with non-negligible probability ϵ we get at least $q + 2$ accepting transcripts. A simple counting argument gives that there has to be a set of four transcripts fulfilling the

above conditions. Let these transcripts be $((c_0, c_1), \alpha^{(i)}, (\mathbf{t}_1^{(i)}, \mathbf{e}_1^{(i)}), \mathsf{ch}_2^{(i)}, \mathsf{resp}_2^{(i)})$, where $\alpha^{(1)} = \alpha^{(2)} \neq \alpha^{(3)} = \alpha^{(4)}$, $\mathbf{t}_1^{(1)} = \mathbf{t}_1^{(2)} \neq \mathbf{t}_1^{(3)} = \mathbf{t}_1^{(4)}$, $\mathbf{e}_1^{(1)} = \mathbf{e}_1^{(2)} \neq \mathbf{e}_1^{(3)} = \mathbf{e}_1^{(4)}$, $\mathsf{ch}_2^{(1)} = \mathsf{ch}_2^{(3)} = 0$, $\mathsf{ch}_2^{(2)} = \mathsf{ch}_2^{(4)} = 1$, $\mathsf{resp}_2^{(1)} = \mathbf{r}_0^{(1)}$, $\mathsf{resp}_2^{(3)} = \mathbf{r}_0^{(3)}$, $\mathsf{resp}_2^{(2)} = \mathbf{r}_1^{(2)}$, $\mathsf{resp}_2^{(4)} = \mathbf{r}_1^{(4)}$. Since the commitment (c_0, c_1) is the same in all four transcripts, we have

$$\begin{aligned} &Com(\mathbf{r}_0^{(1)}, \alpha^{(1)}\mathbf{r}_0^{(1)} - \mathbf{t}_1^{(1)}, \alpha^{(1)}\mathbf{F}(\mathbf{r}_0^{(1)}) - \mathbf{e}_1^{(1)}) = \\ &Com(\mathbf{r}_0^{(3)}, \alpha^{(3)}\mathbf{r}_0^{(3)} - \mathbf{t}_1^{(3)}, \alpha^{(3)}\mathbf{F}(\mathbf{r}_0^{(3)}) - \mathbf{e}_1^{(3)}) \end{aligned} \tag{1}$$

$$\begin{aligned} &Com(\mathbf{r}_1^{(2)}, \alpha^{(2)}(\mathbf{v} - \mathbf{F}(\mathbf{r}_1^{(2)})) - \mathbf{G}(\mathbf{t}_1^{(2)}, \mathbf{r}_1^{(2)}) - \mathbf{e}_1^{(2)}) = \\ &Com(\mathbf{r}_1^{(4)}, \alpha^{(4)}(\mathbf{v} - \mathbf{F}(\mathbf{r}_1^{(4)})) - \mathbf{G}(\mathbf{t}_1^{(4)}, \mathbf{r}_1^{(4)}) - \mathbf{e}_1^{(4)}) \end{aligned} \tag{2}$$

If any of the arguments of Com on the left-hand side is different from the one on the right-hand side in (1) or in (2), then we get two different openings of Com, which breaks its computationally binding property.

If they are the same in both (1) and (2), then from (1):

$(\alpha^{(1)} - \alpha^{(3)})\mathbf{r}_0^{(1)} = \mathbf{t}_1^{(1)} - \mathbf{t}_1^{(3)}$ and $(\alpha^{(1)} - \alpha^{(3)})\mathbf{F}(\mathbf{r}_0^{(1)}) = \mathbf{e}_1^{(1)} - \mathbf{e}_1^{(3)}$,

and from (2): $(\alpha^{(2)} - \alpha^{(4)})(\mathbf{v} - \mathbf{F}(\mathbf{r}_1^{(2)})) = \mathbf{G}(\mathbf{t}_1^{(2)} - \mathbf{t}_1^{(4)}, \mathbf{r}_1^{(2)}) + \mathbf{e}_1^{(2)} - \mathbf{e}_1^{(4)}$.

Combining the two,

$$(\alpha^{(2)} - \alpha^{(4)})(\mathbf{v} - \mathbf{F}(\mathbf{r}_1^{(2)})) = (\alpha^{(2)} - \alpha^{(4)})\mathbf{G}(\mathbf{r}_0^{(1)}, \mathbf{r}_1^{(2)}) + (\alpha^{(2)} - \alpha^{(4)})\mathbf{F}(\mathbf{r}_0^{(1)}),$$

and since $\alpha^{(2)} \neq \alpha^{(4)}$ we get $\mathbf{v} = \mathbf{F}(\mathbf{r}_1^{(2)}) + \mathbf{G}(\mathbf{r}_0^{(1)}, \mathbf{r}_1^{(2)}) + \mathbf{F}(\mathbf{r}_0^{(1)})$, i.e, $\mathbf{r}_0^{(1)} + \mathbf{r}_1^{(2)}$ is a solution to the given $\mathcal{MQ}$ problem. □

We will look into the inner workings of the IDS in more detail in Section 5, where we also introduce the related 3-pass scheme.

4 Fiat-Shamir for 5-pass identification schemes

For several intractability assumptions, the most efficient IDS are five pass, i.e. IDS where a transcript consists of five messages. Here, efficiency refers to the size of all communication of sufficient rounds to make the soundness error negligible. This becomes especially relevant when one wants to turn an IDS into a signature scheme as it is closely related to the signature size of the resulting scheme.

In [EDV+12], the authors present a Fiat-Shamir style transform for $(2n + 1)$-pass IDS fulfilling a certain kind of canonical structure. To provide some intuition, a five pass IDS is called canonical in the above sense if $\mathcal{P}$ starts with a commitment com_1, $\mathcal{V}$ replies with a challenge ch_1, $\mathcal{P}$ sends a first response resp_1, $\mathcal{V}$ replies with a second challenge ch_2 and finally $\mathcal{P}$ returns a second response resp_2. Based on this transcript, $\mathcal{V}$ then accepts or rejects. The authors of [EDV+12] also present a security reduction for signature schemes derived from such IDS using a security property of the IDS which they call *special n-soundness*. Intuitively, this property says that given two transcripts that agree

on all messages but the last challenge and possibly the last response, one can extract a valid secret key.

In this section we first show that any $(2n+1)$-pass IDS that fulfills the requirements of the security reduction in [EDV+12] can be converted into a 3-pass IDS by letting $\mathcal{P}$ choose all but the last challenge uniformly at random himself. The main reason this is possible is the *special n-soundness*. On the other hand, we argue that existing 5-pass schemes in the literature do not fulfill *special n-soundness* and prove it for the 5-pass $\mathcal{MQ}$-IDS from [SSH11]. Hence, they can neither be turned into 3-pass schemes, nor does the security reduction from [EDV+12] apply. Afterwards we give a security reduction for a less generic class of 5-pass IDS which covers many 5-pass IDS, including [CVE10], [Ste93] and [PP03]. In particular, it covers the 5-pass $\mathcal{MQ}$ scheme from [SSH11].

4.1 The El Yousfi et al. proof

Before we can make any statement about IDS that fall into the case of [EDV+12] we have to define the target of our analysis. A canonical $(2n+1)$-pass IDS is an IDS where the prover and the verifier exchange n challenges and replies. More formally:

Definition 4.1 (Canonical $(2n+1)$-pass identification schemes). *Let $k \in \mathbb{N}$, $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ a $(2n+1)$-pass identification scheme with n challenge spaces $\mathsf{C}_j, 0 < j \leq n$. We call IDS a canonical $(2n+1)$-pass identification scheme if the prover can be split into $n+1$ subroutines $\mathcal{P} = (\mathcal{P}_0, \ldots, \mathcal{P}_n)$ and the verifier into $n+1$ subroutines $\mathcal{V} = (\mathsf{ChS}_1, \ldots, \mathsf{ChS}_n, \mathsf{Vf})$ such that*

- *$\mathcal{P}_0(\mathsf{sk})$ computes the initial commitment com sent as the first message.*
- *$\mathsf{ChS}_j, j \leq n$ computes the j-th challenge message $\mathsf{ch}_j \leftarrow_R \mathsf{C}_j$, sampling a random element from the j-th challenge space.*
- *$\mathcal{P}_i(\mathsf{sk}, \mathsf{trans}_{2i}), 0 < i \leq n$ computes the i-th response of the prover given access to the secret key and trans_{2i}, the transcript so far, containing the first $2i$ messages.*
- *$\mathsf{Vf}(\mathsf{pk}, \mathsf{trans})$, upon access to the public key and the whole transcript outputs $\mathcal{V}$'s final decision.*

Figure 2 describes a canonical 5-pass IDS. The definition implies that a canonical $(2n+1)$-pass IDS is *public coin*. The public coin property just says that the challenges are sampled from the respective challenge spaces using the uniform distribution.

El Yousfi et al. propose a generalized Fiat-Shamir transform that turns a canonical $(2n+1)$-pass IDS into a digital signature scheme. The algorithms of the obtained signature scheme make use of the IDS algorithms as follows. The key generation is just the IDS key generation. The signature algorithm simulates an execution of the IDS, replacing challenge ch_j by the output of a hash function (that maps into C_j) that takes as input the concatenation of the message to be signed and all $2(j-1)+1$ messages that have been exchanged so far. The signature just contains the messages sent by $\mathcal{P}$. The verification algorithm uses the

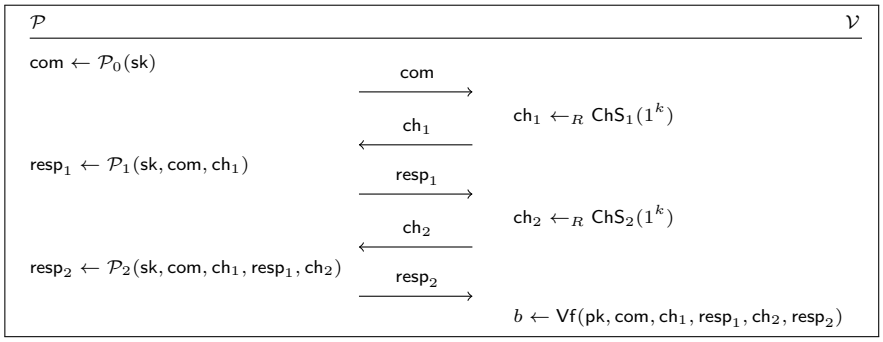

Fig. 2. Canonical 5-pass IDS

signature and the message to be signed to generate a full transcript, recomputing the challenges using the hash function. Then the verification algorithm runs Vf on the public key and the computed transcript and outputs its result.

El Yousfi et al. give a security reduction for the resulting signature scheme if the used IDS is honest-verifier zero-knowledge and fulfills *special n-soundness* defined below. The latter is a generalization of special soundness. Intuitively, special n-soundness says that given two transcripts that agree up to the second-to-last response but disagree on the last challenge, one can extract the secret key.

Definition 4.2 (Special n-soundness). *A canonical $(2n+1)$-pass IDS is said to fulfill special n-soundness if there exists a PPT algorithm $\mathcal{E}$, called the extractor, that given two accepting transcripts* $\mathsf{trans} = (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \ldots, \mathsf{resp}_{n-1}, \mathsf{ch}_n, \mathsf{resp}_n)$ *and* $\mathsf{trans}' = (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \ldots, \mathsf{resp}_{n-1}, \mathsf{ch}'_n, \mathsf{resp}'_n)$ *with* $\mathsf{ch}_n \neq \mathsf{ch}'_n$ *as well as the corresponding public key* pk*, outputs a matching secret key* sk *for* pk *with non-negligible success probability.*

The common special soundness for canonical (3-pass) IDS is hence just special 1-soundness. Please note that El Yousfi et al. define special n-soundness for the resulting signature scheme which in turn requires the used IDS to provide special n-soundness. We decided to follow the more common approach, defining the soundness properties for the IDS.

From $(2n+1)$ to three passes. We now show that every canonical $(2n+1)$-pass IDS that fulfills special n-soundness can be turned into a canonical 3-pass IDS fulfilling special soundness.

Theorem 4.3. *Let* $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ *be a canonical $(2n+1)$-pass IDS that fulfills special n-soundness. Then, the following 3-pass IDS* $\mathsf{IDS}' = (\mathsf{KGen}, \mathcal{P}', \mathcal{V}')$ *is canonical and fulfills special soundness.*

IDS' *is obtained from* IDS *by just moving* $\mathsf{ChS}_j, 0 < j < n$*, (i.e. all but the last challenge generation algorithm) from $\mathcal{V}$ to $\mathcal{P}$: $\mathcal{P}'$ computes* $\mathsf{com}' = (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \ldots, \mathsf{resp}_{n-1}, \mathsf{ch}_{n-1})$ *using* $\mathcal{P}_0, \ldots, \mathcal{P}_{n-1}$ *and* $\mathsf{ChS}_1, \ldots, \mathsf{ChS}_{n-1}$.

After $\mathcal{P}'$ *sent* com', $\mathcal{V}'$ *replies with* $\mathsf{ch}'_1 \leftarrow \mathsf{ChS}_n(1^k)$. $\mathcal{P}'$ *computes* $\mathsf{resp}'_1 \leftarrow \mathcal{P}_n(\mathsf{sk}, \mathsf{trans}_{2n})$ *and* $\mathcal{V}'$ *verifies the transcript using* Vf.

Proof. Clearly, IDS' is a canonical 3-pass IDS. It remains to prove that it is honest-verifier zero-knowledge and that it fulfills special soundness. The latter is straight forward as two transcripts for IDS', that fulfill the conditions in the soundness definition, can be turned into two transcripts for IDS fulfilling the conditions in the n-soundness definition, splitting $\mathsf{com}' = (\mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \ldots, \mathsf{resp}_{n-1}, \mathsf{ch}_{n-1})$ into its parts. Consequently, we can use any extractor for IDS as an extractor for IDS' running in the same time and having the exact same success probability.

Showing honest-verifier zero-knowledge is similarly straight forward. A simulator $\mathcal{S}'$ for IDS' can be obtained from any simulator $\mathcal{S}$ for IDS. $\mathcal{S}'$ just runs $\mathcal{S}$ to obtain a transcript and regroups the messages to produce a valid transcript for IDS'. Again, $\mathcal{S}'$ runs in essentially the same time as $\mathcal{S}$ and achieves the exact same statistical distance. □

The Sakumoto et al 5-pass IDS does not fulfill special n-soundness. The above result raises the question whether this property was overlooked and we can turn all the 5-pass schemes in the literature into 3-pass schemes. This would have the benefit that we could use the classical Fiat-Shamir transform to turn the resulting schemes into signature schemes.

Sadly, this is not the case. The reason is that the extractors for those IDS need more than two transcripts. For example, the extractor for the 5-pass IDS from [SSH11] needs four transcripts such that they all agree on com. The transcripts have to form two pairs such that in a pair the transcripts agree on ch_1 but not on ch_2 and the two pairs disagree on ch_1. The proof given by El Yousfi et al. is flawed. The authors miss that the two secret shares $\mathbf{r}_0$ and $\mathbf{r}_1$ obtained from two different transcripts do not have to be shares of a valid secret key. We now give a formal proof.

Theorem 4.4. *The 5-pass identification scheme from [SSH11] does not fulfill special n-soundness if the computational $\mathcal{MQ}$-problem is hard.*

Proof. We prove this by showing that there exist pairs of transcripts, fulfilling the special n-soundness criteria that can be generated by an adversary without knowledge of the secret key simulating just two executions of the protocol. As a key pair for the $\mathcal{MQ}$-IDS is a random instance of the $\mathcal{MQ}$ problem, special n-soundness of the 5-pass $\mathcal{MQ}$-IDS would imply that the $\mathcal{MQ}$ problem can be solved in probabilistic polynomial time.

Towards a contradiction, assume there exists a PPT extractor $\mathcal{E}$ against the 5-pass $\mathcal{MQ}$-IDS that fulfills Definition 4.2. We show how to build a PPT solver $\mathcal{A}$ for the $\mathcal{MQ}$ problem. Given an instance of the $\mathcal{MQ}$ problem $\mathbf{v}$, $\mathcal{A}$ sets $\mathsf{pk} = \mathbf{v}$ which is a valid public key for the $\mathcal{MQ}$-IDS. Next, $\mathcal{A}$ computes two transcripts as follows. $\mathcal{A}$ samples a random $\alpha \in \mathbb{F}_q$ and random $\mathbf{s}, \mathbf{r}_0, \mathbf{t}_0 \in \mathbb{F}_q^n$, $\mathbf{e}_0 \in \mathbb{F}_q^m$, and computes $\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$, and $\mathbf{t}_1 \leftarrow \alpha\mathbf{r}_0 - \mathbf{t}_0$. Then $\mathcal{A}$ simulates two successful

protocol executions, one for $\mathsf{ch}_2 = 0$, one for $\mathsf{ch}_2 = 1$. To do so, $\mathcal{A}$ impersonates $\mathcal{P}$ and replaces the first challenge with α and the second with 0 in the first run and 1 in the second run. In addition, $\mathcal{A}$ uses the knowledge of α to compute the commitments as:

- $c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$,
- $c_1 \leftarrow Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \alpha\mathbf{F}(\mathbf{r}_0) + \mathbf{e}_0)$.

Then $\mathcal{A}$ computes $\mathbf{e}_1 \leftarrow \alpha\mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$ and sets the second commitment in both runs to $(\mathbf{t}_1, \mathbf{e}_1)$. For $\mathsf{ch}_2 = 0$, $\mathcal{A}$ sets $\mathsf{resp} = \mathbf{r}_0$, and for $\mathsf{ch}_2 = 1$, $\mathcal{A}$ sets $\mathsf{resp} = \mathbf{r}_1$.

Now, the first transcript (when $\mathsf{ch}_2 = 0$) is valid, since $\mathbf{t}_0 = \alpha\mathbf{r}_0 - \mathbf{t}_1$ and $\mathbf{e}_0 = \alpha\mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1$. The second transcript (when $\mathsf{ch}_2 = 1$) is also valid as a straight forward calculation shows. Finally, $\mathcal{A}$ feeds the transcripts to $\mathcal{E}$ and outputs whatever $\mathcal{E}$ outputs. $\mathcal{A}$ has the same success probability as $\mathcal{E}$ and runs in essentially the same time. As $\mathcal{E}$ is a PPT algorithm per assumption, this contradicts the hardness of the computational $\mathcal{MQ}$ problem. □

Clearly, we can also use $\mathcal{A}$ to deal with a parallel execution of many rounds of the scheme. A similar situation arises for all the 5-pass IDS schemes that we found in the literature.

4.2 A Fiat-Shamir transform for most $(2n + 1)$-pass IDS

By now we have established that we are currently lacking security arguments for signature schemes derived from $(2n + 1)$-pass IDS. We now show how to fix this issue for most $(2n + 1)$-pass IDS in the literature. As most of these IDS are 5-pass schemes that follow a certain structure, we restrict ourselves to these cases. There are some generalizations that are straight-forward and possible to deal with, but they massively complicate accessibility of our statements.

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

Definition 4.5 ($q2$-Identification scheme). *Let $k \in \mathbb{N}$. A $q2$-Identification scheme $\mathsf{IDS}(1^k)$ 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 negligible (in k), where the probability is taken over the random choice of the input and the used randomness.*

To keep the security reduction below somewhat generic, we also need a property that defines when an extractor exists for a $q2$-IDS. As we have seen special n-soundness is not applicable. Hence, we give a less generic definition.

Definition 4.6 ($q2$-Extractor). *We say that a $q2$-Identification scheme $\mathsf{IDS}(1^k)$ has a $q2$-*extractor *if there exists a PPT algorithm $\mathcal{E}$, the extractor, that given a public key pk and four transcripts $\mathsf{trans}^{(i)} = (\mathsf{com}, \mathsf{ch}_1^{(i)}, \mathsf{resp}_1^{(i)}, \mathsf{ch}_2^{(i)}, \mathsf{resp}_2^{(i)})$, $i \in \{1, 2, 3, 4\}$, with*

$$\begin{aligned} \mathsf{ch}_1^{(1)} = \mathsf{ch}_1^{(2)} \neq \mathsf{ch}_1^{(3)} = \mathsf{ch}_1^{(4)}, \\ \mathsf{ch}_2^{(1)} = \mathsf{ch}_2^{(3)} \neq \mathsf{ch}_2^{(2)} = \mathsf{ch}_2^{(4)}, \end{aligned} \tag{3}$$

valid with respect to pk*, outputs a matching secret key* sk *for* pk *with non-negligible success probability (in* k*).*

In what follows, let $\mathsf{IDS}^r = (\mathsf{KGen}, \mathcal{P}^r, \mathcal{V}^r)$ be the parallel composition of r rounds of the identification scheme $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$. As the schemes we are concerned with only achieve a constant soundness error, the construction below uses a polynomial number of rounds to obtain an IDS with negligible soundness error as intermediate step. We denote the transcript of the j-th round by $\mathsf{trans}_j = (\mathsf{com}_j, \mathsf{ch}_{1,j}, \mathsf{resp}_{1,j}, \mathsf{ch}_{2,j}, \mathsf{resp}_{2,j})$.

Construction 4.7 (Fiat-Shamir transform for $q2$-IDS). *Let* $k \in \mathbb{N}$ *the security parameter,* $\mathsf{IDS} = (\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ *a* $q2$*-Identification scheme that achieves soundness with soundness error* κ*. Select* r*, the number of (parallel) rounds of* IDS*, such that* $\kappa^r = \mathrm{negl}(k)$*, and that the challenge spaces of the composition* IDS^r*,* $\mathsf{C}_1^r, \mathsf{C}_2^r$ *have exponential size in* k*. Moreover, select cryptographic hash functions* $H_1 : \{0,1\}^* \to \mathsf{C}_1^r$ *and* $H_2 : \{0,1\}^* \to \mathsf{C}_2^r$*. The* $q2$*-signature scheme* $q2\text{-}\mathsf{Dss}(1^k)$ *derived from* IDS *is the triplet of algorithms* $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vf})$ *with:*

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(1^k)$,
- $\sigma = (\sigma_0, \sigma_1, \sigma_2) \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$ *where* $\sigma_0 = \mathsf{com} \leftarrow \mathcal{P}_0^r(\mathsf{sk})$*,* $h_1 = H_1(m, \sigma_0)$*,* $\sigma_1 = \mathsf{resp}_1 \leftarrow \mathcal{P}_1^r(\mathsf{sk}, \sigma_0, h_1)$*,* $h_2 = H_2(m, \sigma_0, h_1, \sigma_1)$*, and* $\sigma_2 = \mathsf{resp}_2 \leftarrow \mathcal{P}_2^r(\mathsf{sk}, \sigma_0, h_1, \sigma_1, h_2)$*.*
- $\mathsf{Vf}(\mathsf{pk}, m, \sigma)$ *parses* $\sigma = (\sigma_0, \sigma_1, \sigma_2)$*, computes the values* $h_1 = H_1(m, \sigma_0)$*,* $h_2 = H_2(m, \sigma_0, h_1, \sigma_1)$ *as above and outputs* $\mathcal{V}^r(\mathsf{pk}, \sigma_0, h_1, \sigma_1, h_2, \sigma_2)$*.*

Correctness of the scheme follows immediately from the correctness of IDS.

4.3 Security of $q2$-signature schemes.

We now give a security reduction for the above transform in the random oracle model assuming that the underlying $q2$-IDS is honest-verifier zero-knowledge, achieves soundness with constant soundness error, and has a $q2$-extractor. More specifically, we prove the following theorem:

Theorem 4.8 (EU-CMA security of $q2$-signature schemes). *Let* $k \in \mathbb{N}$*,* $\mathsf{IDS}(1^k)$ *a* $q2$*-IDS that is honest-verifier zero-knowledge, achieves soundness with constant soundness error* κ *and has a* $q2$*-extractor. Then* $q2\text{-}\mathsf{Dss}(1^k)$*, the* $q2$*-signature scheme derived applying Construction 4.7 is existentially unforgeable under adaptive chosen message attacks.*

In the following, we model the functions H_1 and H_2 as independent random oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. To proof Theorem 4.8, we proceed in several steps. Our proof builds on techniques introduced by Pointcheval and Stern [PS96]. As the reduction is far from being tight, we refrain from doing an exact proof as it does not buy us anything but a complicated statement. We first recall an important tool from [PS96] called the splitting lemma.

Lemma 4.9 (Splitting lemma [PS96]). *Let $A \subset X \times Y$, such that $\Pr[A(x,y)] \geqslant \epsilon$. Then, there exists $\Omega \subset X$, such that*

$$\Pr[x \in \Omega] \geqslant \epsilon/2,$$
$$\Pr[A(a,y) | a \in \Omega] \geqslant \epsilon/2.$$

We now present a forking lemma for $q2$-signature schemes. The lemma shows that we can obtain four valid signatures which contain four valid transcripts of the underlying IDS, given a successful key-only adversary. Moreover, these four traces fulfill a certain requirement on the challenges (here the related parts of the hash function outputs) that we need later.

Lemma 4.10 (Forking lemma for $q2$-signature schemes). *Let $k \in \mathbb{N}$, $\mathsf{Dss}(1^k)$ a $q2$-signature scheme with security parameter k. If there exists a PPT adversary $\mathcal{A}$ that can output a valid signature message pair (m, σ) with non-negligible success probability, given only the public key as input, then, with non-negligible probability, rewinding $\mathcal{A}$ a polynomial number of times (with same randomness) but different oracles, outputs 4 valid signature message pairs $(m, \sigma = (\sigma_0, \sigma_1^{(i)}, \sigma_2^{(i)})$, $i \in \{1,2,3,4\}$, such that for the associated hash values it holds that*

$$\begin{aligned} h_{1,j}^{(1)} = h_{1,j}^{(2)} \neq h_{1,j}^{(3)} = h_{1,j}^{(4)}, \\ h_{2,j}^{(1)} = h_{2,j}^{(3)} \neq h_{2,j}^{(2)} = h_{2,j}^{(4)}, \end{aligned} \tag{4}$$

for some round $j \in \{1, \ldots, r\}$.

Proof. To prove the Lemma we need to show that we can rewind $\mathcal{A}$ three times and the probability that $\mathcal{A}$ succeeds in forging a (different) signature in all four runs is non-negligible. Moreover, we have to show that the signatures have the additional property claimed in the Lemma, again with non-negligible probability.

Let $\omega \in R_w$ be $\mathcal{A}$'s random tape with R_w the set of allowable random tapes. During the attack $\mathcal{A}$ may ask polynomially many queries (in the security parameter k) $Q_1(k)$ and $Q_2(k)$ to the random oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Let $q_{1,1}$, $q_{1,2}, \ldots,$ q_{1,Q_1} and $q_{2,1}$, $q_{2,2}, \ldots,$ q_{2,Q_2} be the queries to $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. Moreover, let $(r_{1,1}, r_{1,2}, \ldots, r_{1,Q_1}) \in (\mathsf{C}_1^r)^{Q_1}$ and $(r_{2,1}, r_{2,2}, \ldots, r_{2,Q_2}) \in (\mathsf{C}_2^r)^{Q_2}$ the corresponding answers of the oracles.

Towards proving the first point, we assume that $\mathcal{A}$ also outputs h_1, h_2 with the signature and a signature is considered invalid if those do not match the responses of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. This assumption is without loss of generality as we can construct such $\mathcal{A}$ from any $\mathcal{A}'$ that does not output h_1, h_2. $\mathcal{A}$ just runs $\mathcal{A}'$ and given the result queries $\mathcal{O}_1$ and $\mathcal{O}_2$ for h_1, h_2 and outputs everything. Clearly $\mathcal{A}$ succeeds with the same success probability as $\mathcal{A}'$ and runs in essentially the same time, making just one more query to each RO.

Denote by F the event that $\mathcal{A}$ outputs a valid message signature pair $(m, \sigma^{(1)} = (\sigma_0, \sigma_1^{(1)}, \sigma_2^{(1)}))$ with the associated hash values $h_1^{(1)}, h_2^{(1)}$. Per assumption, this event occurs with non-negligible probability, i.e., $\mathsf{Pr}[\mathsf{F}] = \frac{1}{P(k)}$, for some polynomial $P(k)$. In addition, F implies $h_1^{(1)} = \mathcal{O}_1(m, \sigma_0)$ and $h_2^{(1)} = \mathcal{O}_2(m, \sigma_0, h_1^{(1)}, \sigma_1^{(1)})$.

As $h_1^{(1)}, h_2^{(1)}$ are chosen uniformly at random from exponentially large sets $\mathsf{C}_1^r, \mathsf{C}_2^r$, the probability that $\mathcal{A}$ did not query $\mathcal{O}_1$ for $h_1^{(1)}$ and $\mathcal{O}_2$ for $h_2^{(1)}$ is negligible. Hence, there exists a polynomial P' such that the event F' that F occurs and $\mathcal{A}$ queried $\mathcal{O}_1$ for $h_1^{(1)}$ and $\mathcal{O}_2$ for $h_2^{(1)}$ has probability $\mathsf{Pr}[\mathsf{F}'] = \frac{1}{P'(k)}$.

For the moment only consider the second oracle. As of the previous equation, there exists at least one $\beta \leqslant Q_2$ such that

$$\mathsf{Pr}[\mathsf{F}' \wedge q_{2,\beta} = (m, \sigma_0, h_1^{(1)}, \sigma_1^{(1)})] \geqslant \frac{1}{Q_2(k)P'(k)}$$

where the probability is taken over the random coins of $\mathcal{A}$ and $\mathcal{O}_2$. Informally, the following steps just show that the success of an algorithm with non-negligible success probability cannot be conditioned on an event that occurs only with negligible probability (i.e. the outcome of the $q_{2,\beta}$ query landing in some negligible subset).

Let $\mathcal{B} = \{(\omega, r_{2,1}, r_{2,2}, \ldots, r_{2,Q_2}) | \omega \in R_w \wedge (r_{2,1}, r_{2,2}, \ldots, r_{2,Q_2}) \in (\mathsf{C}_2^r)^{Q_2} \wedge \mathsf{F}' \wedge q_{2,\beta} = (m, \sigma_0, h_1^{(1)}, \sigma_1^{(1)})\}$, i.e., the set of random tapes and oracle responses such that $\mathsf{F}' \wedge q_{2,\beta} = (m, \sigma_0, h_1^{(1)}, \sigma_1^{(1)})$. This implies that there exists a non-negligible set of "good" random tapes $\Omega_\beta \subseteq R_\omega$ for which $\mathcal{A}$ can provide a valid signature and $q_{2,\beta}$ is the oracle query fixing $h_2^{(1)}$. Applying the splitting lemma, we get that

$$\begin{aligned} \Pr[w \in \Omega_\beta] &\geqslant \frac{1}{2Q_2(k)P'(k)} \\ \Pr[(\omega, r_{2,1}, r_{2,2}, \ldots, r_{2,Q_2}) \in \mathcal{B} | w \in \Omega_\beta] &\geqslant \frac{1}{2Q_2(k)P'(k)} \end{aligned}$$

Applying the same reasoning again we can derive from the later probability being non-negligible that there exists a non-negligible subset $\Omega_{\beta,\omega}$ of the "good" oracle responses $(r_{2,1}, r_{2,2}, \ldots, r_{2,\beta-1})$ such that $(\omega, r_{2,1}, r_{2,2}, \ldots, r_{2,Q_2}) \in \mathcal{B}$. Applying the splitting lemma again, we get

$$\begin{aligned} \Pr[(r_{2,1}, \ldots, r_{2,\beta-1}) \in \Omega_{\beta,\omega}] &\geqslant \frac{1}{4Q_2(k)P'(k)} \\ \Pr[(\omega, r_{2,1}, \ldots, r_{2,Q_2}) \in \mathcal{B} | (r_{2,1}, \ldots, r_{2,\beta-1}) \in \Omega_{\beta,\omega})] &\geqslant \frac{1}{4Q_2(k)P'(k)} \end{aligned}$$

This means that rewinding $\mathcal{A}$ to the point where it made query $q_{2,\beta}$ and running it with new, random $r'_{2,\beta}, \ldots, r'_{2,Q_2}$ has a non-negligible probability of $\mathcal{A}$ outputting another valid signature. Therefore, we can use $\mathcal{A}$ to find two valid signature message pairs with associated hash values $(m, \sigma = (\sigma_0, \sigma_1^{(1)}, \sigma_2^{(1)}), h_1^{(1)}, h_2^{(1)})$, $(m, \sigma^{(2)} = (\sigma_0, \sigma_1^{(2)}, \sigma_2^{(2)}), h_1^{(2)}, h_2^{(2)})$, with $h_2^{(1)} \neq h_2^{(2)}$ and such that $(\sigma_0, h_1^{(1)}, \sigma_1^{(1)}) = (\sigma_0, h_1^{(2)}, \sigma_1^{(2)})$, with non-negligible probability.

We now rewind the adversary again using exactly the same technique as above but now considering the queries to $\mathcal{O}_1$ and its responses. In the replay we change

the responses of $\mathcal{O}_1$ to obtain a third signature that differs from the previously obtained ones in the first associated hash value. It can be shown that with non-negligible probability $\mathcal{A}$ will output a third signature on m, $\sigma^{(3)} = (\sigma_0, \sigma_1^{(3)}, \sigma_2^{(3)})$, with associated hash values $(h_1^{(3)}, h_2^{(3)})$ such that $h_1^{(3)} \neq h_1^{(2)} = h_1^{(1)}$.

Finally, we rewind the adversary a third time, keeping the responses of $\mathcal{O}_1$ from the last rewind and focusing on $\mathcal{O}_2$ again. Again, with non-negligible probability $\mathcal{A}$ will produce yet another signature on m, $\sigma^{(4)} = (\sigma_0, \sigma_1^{(4)}, \sigma_2^{(4)})$ with associated hash values $(h_1^{(4)}, h_2^{(4)})$ such that $h_1^{(4)} = h_1^{(3)}$ and $h_2^{(4)} \neq h_2^{(3)}$.

Summing up, rewinding the adversary three times, we can find four valid signatures $\sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)}$ with the above property on the associated hash values with non-negligible success probability $\dfrac{1}{P(k)}$ for some polynomial $P(k)$. Let us denote this event by $\mathcal{E}_\sigma$. So we have that

$$\Pr[\mathcal{E}_\sigma] \geqslant \frac{1}{P(k)}.$$

What remains is to show that the obtained signatures satisfy the particular structure from the lemma (Equation 4) with non-negligible probability.

Let $\mathcal{H}$ be the event that there exists a $j \in \{1, \ldots, r\}$ such that (4) is satisfied. We have that

$$\mathsf{Pr}[\mathcal{E}_\sigma \wedge \mathcal{H}] = \mathsf{Pr}[\mathcal{E}_\sigma] - \mathsf{Pr}[\neg\mathcal{H} \wedge \mathcal{E}_\sigma] = \mathsf{Pr}[\mathcal{E}_\sigma] - \mathsf{Pr}[\neg\mathcal{H}|\mathcal{E}_\sigma]\mathsf{Pr}[\mathcal{E}_\sigma] \geqslant \frac{1}{P(k)} - \mathsf{Pr}[\neg\mathcal{H}|\mathcal{E}_\sigma]$$

We will now give a statistical argument why $\mathsf{Pr}[\neg\mathcal{H}|\mathcal{E}_\sigma]$ is negligible.

As argued above, the hash values associated with the signatures must be outcomes of the RO queries of $\mathcal{A}$. During its first run, $\mathcal{A}$ can choose the first hash value $h_1^{(1)}$ from his Q_1 queries to $\mathcal{O}_1$ and the second hash value $h_2^{(1)}$ from his Q_2 queries to $\mathcal{O}_2$. The total number of possible combinations is Q_1Q_2. The hash values associated with the second signature are $h_1^{(2)} = h_1^{(1)}$ (as $\mathcal{E}_\sigma$) and $h_2^{(2)}$. So, the first hash value is fixed and the second is chosen from a set of no more than Q_2 responses from $\mathcal{O}_2$. Following the same arguments, the hash pair associated with the third signature is chosen from a set of size Q_1Q_2 and the one associated with the fourth signature from a set of size Q_2. The oracle outputs are uniformly distributed within C_1^r and C_2^r, respectively. Hence, the set of all possible combinations of hash values that $\mathcal{A}$ could output has size

$$\lambda(k) \leq Q_1Q_2 \cdot Q_2 \cdot Q_1Q_2 \cdot Q_2,$$

which is a polynomial in k as Q_1 and Q_2 are.

Recall C_1 has size q and C_2 size 2. The probability that the required pattern did not occur in the four-tupel of challenges derived from random hash values for one internal round j is

$$\mathsf{Pr}[\neg\mathcal{H}_j] = 1 - \mathsf{Pr}[\mathcal{H}_j] = 1 - \frac{q-1}{2^2 q} = \frac{3q+1}{4q}.$$

The last follows from the fact that out of all 2^4q^2 4-tuples

$$((\alpha_1, \beta_1), (\alpha_1, \beta_2), (\alpha_2, \beta_3), (\alpha_2, \beta_4)) \in (\mathsf{C}_1 \times \mathsf{C}_2)^4$$

exactly $2^2q(q-1)$ satisfy $\alpha_1 \neq \alpha_2$, $\beta_1 \neq \beta_2$, $\beta_3 \neq \beta_4$. Hence, the probability that a random four-tuple of hash values does not have a single internal round that satisfies (4) and hence fulfills $\neg\mathcal{H}$ is

$$\mathsf{Pr}[\neg\mathcal{H}] = \left(\frac{3q+1}{4q}\right)^r = \mathrm{negl}(k)\,.$$

According to Construction 4.7, the number of rounds r must be super-logarithmic (in k), to fulfill C_2^r being exponentially large (in k). Hence, the above is negligible for random hash values.

Finally, we just have to combine the two results. The adversary can at most choose out of a polynomially bounded number of four-tuples of hash pairs. Each of these four-tuples has a negligible probability of fulfilling $\neg\mathcal{H}$. Hence, the probability that all the possible combinations of query responses even contain a four-tuple that does not fulfill $\mathcal{H}$ is negligible. So,

$$\mathsf{Pr}[\neg\mathcal{H}|\mathcal{E}_\sigma] = \mathrm{negl}(k)\,,$$

and hence, the conditions from the lemma are satisfied with non-negligible probability. □

With Lemma 4.10 we can already establish unforgeability under key only attacks:

Corollary 4.11 (Key-only attack resistance). *Let $k \in \mathbb{N}$, $\mathsf{IDS}(1^k)$ a $q2$-IDS that achieves soundness with constant soundness error κ and has a $q2$-extractor. Then $q2$-$\mathsf{Dss}(1^k)$, the $q2$-signature scheme derived applying Construction 4.7 is unforgeable under key-only attacks.*

A straight forward application of Lemma 4.10 allows to generate the four traces needed to apply the $q2$-extractor. The obtained secret key can then be used to violate soundness.

For EU-CMA security, we still have to deal with signature queries. The following lemma shows that a reduction can produce valid responses to the adversarial signature queries if the identification scheme is honest-verifier zero-knowledge.

Lemma 4.12. *Let $k \in \mathbb{N}$ the security parameter, $\mathsf{IDS}(1^k)$ a $q2$-IDS that is honest-verifier zero-knowledge. Then any PPT adversary $\mathcal{B}$ against the EU-CMA-security of $q2$-$\mathsf{Dss}(1^k)$, the $q2$-signature scheme derived by applying Construction 4.7, can be turned into a key-only adversary $\mathcal{A}$ with the properties described in Lemma 4.10. $\mathcal{A}$ runs in polynomial time and succeeds with essentially the same success probability as $\mathcal{B}$.*

Proof. By construction. We show how to construct an oracle machine $\mathcal{A}^{\mathcal{B},\mathcal{S},\mathcal{O}_1,\mathcal{O}_2}$ that has access to $\mathcal{B}$, an honest-verifier zero-knowledge simulator $\mathcal{S}$, and random

oracles $\mathcal{O}_1, \mathcal{O}_2$. $\mathcal{A}$ produces a valid signature for $q2\text{-}\mathsf{Dss}(1^k)$ given only a public key running in time polynomial in k and achieving essentially the same success probability (up to a negligible difference) as $\mathcal{B}$.

Upon input of public key pk, $\mathcal{A}$ runs $\mathcal{B}^{\mathcal{O}'_1,\mathcal{O}'_2,\mathsf{Sign}}(\mathsf{pk})$ simulating the random oracles (ROs) $\mathcal{O}'_1, \mathcal{O}'_2$, as well as the signing oracle Sign towards $\mathcal{B}$. When $\mathcal{B}$ outputs a forgery (m^*, σ^*), $\mathcal{A}$ just forwards it.

To simulate the ROs, $\mathcal{A}$ keeps two initially empty tables of query-response pairs, one per oracle. Whenever $\mathcal{B}$ queries $\mathcal{O}'_b$, $\mathcal{A}$ first checks if the table for $\mathcal{O}'_b$ already contains a pair for this query. If such a pair exists, $\mathcal{A}$ just returns the stored response. Otherwise, $\mathcal{A}$ forwards the query to its own $\mathcal{O}_b$.

As IDS is honest-verifier zero-knowledge there exists a PPT simulator $\mathcal{S}$ that upon input of a IDS public key generates a valid transcript that is indistinguishable of the transcripts generated by honest protocol executions. Whenever $\mathcal{B}$ queries the signature oracle with message m, $\mathcal{A}$ runs $\mathcal{S}$ r times, to obtain r valid transcripts. $\mathcal{A}$ combines the transcripts to obtain a valid signature with associated hashes $\sigma = ((\sigma_0, \sigma_1, \sigma_2), h_1, h_2)$. Before outputting σ, $\mathcal{A}$ checks if the table for $\mathcal{O}'_1$ already contains an entry for query (m, σ_0). If so, $\mathcal{A}$ aborts. Otherwise, $\mathcal{A}$ adds the pair $((m, \sigma_0), h_1)$. Then, $\mathcal{A}$ checks the second table for query $(m, \sigma_0, h_1, \sigma_1)$. Again, $\mathcal{A}$ aborts if it finds such an entry and adds $((m, \sigma_0, h_1, \sigma_1), h_2)$, otherwise.

The probability that $\mathcal{A}$ aborts is negligible in k. When answering signature queries, $\mathcal{A}$ verifies that certain queries were not made before. Both queries contain σ_1 which takes any given value only with negligible probability. On the other hand, the total number of queries that $\mathcal{B}$ makes to all its oracles is polynomially bounded. Hence, the probability that one of the two queries was already made before is negligible. If $\mathcal{A}$ does not abort, it perfectly simulates all oracles towards $\mathcal{B}$. Hence, $\mathcal{B}$ – and thereby $\mathcal{A}$ – succeeds with the same probability as in the real $\mathsf{EU\text{-}CMA}$ game in this case. Hence, $\mathcal{A}$ succeeds with essentially the same probability as $\mathcal{B}$. □

We now got everything we need to prove Theorem 4.8. The proof is a straight forward application of the previous two lemmas.

Proof (of Theorem 4.8). Towards a contradiction, assume that there exists a PPT adversary $\mathcal{B}$ against the $\mathsf{EU\text{-}CMA}$-security of $q2\text{-}\mathsf{Dss}$ succeeding with non-negligible probability. We show how to construct a PPT impersonator $\mathcal{C}$ breaking the soundness of IDS. Applying Lemma 4.12, $\mathcal{C}$ can construct a PPT key-only forger $\mathcal{A}$, with essentially the same success probability as $\mathcal{B}$. Given a public key for IDS (which is a valid $q2\text{-}\mathsf{Dss}$ public key) $\mathcal{C}$ runs $\mathcal{A}$ as described in Lemma 4.10. That way $\mathcal{C}$ can use $\mathcal{A}$ to obtain four signatures that per (4) lead four transcripts as required by the $q2$-extractor $\mathcal{E}$. Running $\mathcal{E}$, $\mathcal{C}$ can extract a valid secret key that allows to impersonate $\mathcal{P}$ with success probability 1.

$\mathcal{C}$ just runs $\mathcal{A}$ and $\mathcal{E}$, two PPT algorithms. Consequently, $\mathcal{C}$ runs in polynomial time. Also, $\mathcal{A}$ and $\mathcal{E}$ both have non-negligible success probability implying that also $\mathcal{C}$ succeeds with non-negligible probability. □

5 Our proposal

In the previous sections, we gave security arguments for a Fiat-Shamir transform of 5-pass IDS that contain two challenges , from $\{0, \ldots, q-1\}$ and $\{0, 1\}$ respectively, where $q \in \mathbb{Z}^*$. In this section we apply the transform to the 5-pass IDS from [SSH11] (see Section 3). Before discussing the 5-pass scheme, which we dub MQDSS, we first briefly examine the signature scheme obtained by applying the traditional Fiat-Shamir transform to the 3-pass IDS in [SSH11], to obtain a baseline. Then we give a generic description of MQDSS and prove it secure.

The IDS requires an $\mathcal{MQ}$ system $\mathbf{F}$ as input, potentially system-wide. We could simply select one function $\mathbf{F}$ and define it as a system parameter for all users. Instead, we choose to derive it from a unique seed that is included in each public key. This increases the size of pk by k bits, and adds some cost for seed expansion when signing and verifying. However, selecting a single system-wide $\mathbf{F}$ might allow an attacker to focus their efforts on a single $\mathbf{F}$ for all users, and would require whoever selects this system parameter to convince all users of its randomness (which is not trivial [BCC+14]). For consistency with literature, we still occasionally refer to $\mathbf{F}$ as the 'system parameter'.

Note that the signing procedure described below is slightly more involved than is suggested by Construction 4.7. Where the transformed construction operates directly on the message m, we first apply what is effectively a randomized hash function. As discussed in [HK06], this extra step provides resilience against collisions in the hash function at only little extra cost. A similar construction appears e.g. in SPHINCS [BHH+15]. The digest (and thus the signature) is still derived from m and sk deterministically.

5.1 Establishing a baseline using the 3-pass scheme over $\mathbb{F}_2$

In the interest of brevity, we will not go into the details of the derived signature scheme here – instead, we refer to Appendix A.

For the 3-pass scheme, we select $n = m = 256$ over $\mathbb{F}_2$. This results in signatures of 54.81 KB, and a key pair of 64 bytes per key. We ran benchmarks on a single 3.5GHz core of an Intel Core i7-4770K CPU, measuring 118 088 992 cycles for signature generation, 8 066 324 cycles for key generation and 82 650 156 cycles for signature verification (or 33.7 ms, 2.30 ms and 23.6 ms, respectively).

5.2 The 5-pass scheme over $\mathbb{F}_{31}$

As can be seen from the results above, the plain 3-pass scheme over $\mathbb{F}_2$ is quite inefficient, both in terms of signature size and signing speed. This is a direct consequence of the large number of variables and equations required to achieve 128 bits of post-quantum security using $\mathcal{MQ}$ over $\mathbb{F}_2$, as well as the high number of rounds required (see Appendix A for an analysis). Using a 5-pass scheme over $\mathbb{F}_{31}$ allows for a smaller n and m, as well as a smaller number of rounds. One might wonder why we do not consider different fields for the 3-pass scenario, instead. This turns out to be suboptimal: contrary to the 5-pass scheme, this

does not result in a knowledge error reduction, but does increase the transcript size per round.

The MQDSS **signature scheme.** We now explicitly construct the functions KGen, Sign and Vf in accordance with Definition 2.1. Specific values for the parameters that achieve 128 bit post-quantum security are given in the next section. We start by presenting the parameters of the scheme in general.

Parameters. MQDSS is parameterized by a security parameter $k \in \mathbb{N}$, and $m, n \in \mathbb{N}$ such that the security level of the $\mathcal{MQ}$ instance $\mathcal{MQ}(n, m, \mathbb{F}_2) \geq k$. The latter fix the description length of the equation system $\mathbf{F}$, $F_{len} = m \cdot \frac{n \cdot (n+1)}{2}$.

- Cryptographic hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$, $H_1 : \{0,1\}^{2k} \to {\mathbb{F}_{31}}^r$, and $H_2 : \{0,1\}^{2k} \to \{0,1\}^r$.
- two string commitment functions $Com_0 : {\mathbb{F}_{31}}^n \times {\mathbb{F}_{31}}^n \times {\mathbb{F}_{31}}^m \to \{0,1\}^k$ and $Com_1 : {\mathbb{F}_{31}}^n \times {\mathbb{F}_{31}}^m \to \{0,1\}^k$,
- pseudo-random generators $G_{\mathcal{S}_F} : \{0,1\}^k \to {\mathbb{F}_{31}}^{F_{len}}$, $G_{SK} : \{0,1\}^k \to {\mathbb{F}_{31}}^n$, and $G_c : \{0,1\}^{2k} \to {\mathbb{F}_{31}}^{r \cdot (2n+m)}$.

Key generation. Given the security parameter k, we randomly sample a secret key of k bits $SK \leftarrow_R \{0,1\}^k$ as well as a seed $\mathcal{S}_F \leftarrow_R \{0,1\}^k$. We then select a pseudorandom $\mathcal{MQ}$ system $\mathbf{F}$ from $\mathcal{MQ}(n, m, \mathbb{F}_{31})$ by expanding $\mathcal{S}_F$. In total, we must generate $F_{len} = m \cdot (\frac{n \cdot (n+1)}{2} + n)$ elements for $\mathbf{F}$, to use as coefficients for both the quadratic and the linear monomials. We use the pseudorandom generator $G_{\mathcal{S}_F}$ for this.

In order to compute the public key, we want to use the secret key as input for the $\mathcal{MQ}$ function defined by $\mathbf{F}$. As SK is a k-bit string rather than a sequence of n elements from $\mathbb{F}_{31}$, we instead use it as a seed for a pseudorandom generator as well, deriving $SK_{\mathbb{F}_{31}} = G_{SK}(SK)$. It is then possible to compute $\boldsymbol{PK_v} = \mathbf{F}(SK_{\mathbb{F}_{31}})$. The secret key $\mathsf{sk} = (SK, \mathcal{S}_F)$ and the public key $\mathsf{pk} = (\mathcal{S}_F, \boldsymbol{PK_v})$ require $2 \cdot k$ and $k + 5 \cdot m$ bits respectively, assuming 5 bits per $\mathbb{F}_{31}$ element.

Signing. The signature algorithm takes as input a message $m \in \{0,1\}^*$ and a secret key $\mathsf{sk} = (SK, \mathcal{S}_F)$. Similarly as in the key generation, we derive $\mathbf{F} = G_{\mathcal{S}_F}(\mathcal{S}_F)$. Then, we derive a message-dependent random value $R = \mathcal{H}(SK \parallel m)$, where "$\parallel$" is string concatenation. Using this random value R, we compute the randomized message digest $D = \mathcal{H}(R \parallel m)$. The value R must be included in the signature, so that a verifier can derive the same randomized digest.

As mentioned in Definition 2.4, the core of the derived signature scheme essentially consists of iterations of the IDS. We refer to the number of required iterations to achieve the security level k as r (note that this should not be confused with $\mathbf{r}_0$ and $\mathbf{r}_1$, which are vectors of elements of $\mathbb{F}_{31}$).

Given SK and D, we now compute $G_c(SK, D)$ to produce $(\mathbf{r}_{(0,0)}, \ldots, \mathbf{r}_{(0,r)}, \mathbf{t}_{(0,0)}, \ldots, \mathbf{t}_{(0,r)}, \mathbf{e}_{(0,0)}, \ldots, \mathbf{e}_{(0,r)})$. Using these values, we compute $c_{(0,i)}$ and $c_{(1,i)}$ for each round i, as defined in the IDS. Recall that $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$, and that Com_0 and Com_1 are string commitment functions:

$$c_{(0,i)} = Com_0(\mathbf{r}_{(0,i)}, \mathbf{t}_{(0,i)}, \mathbf{e}_{(0,i)}) \text{ and } c_{(1,i)} = Com_1(\mathbf{r}_{(1,i)}, \mathbf{G}(\mathbf{t}_{(0,i)}, \mathbf{r}_{(1,i)}) + \mathbf{e}_{(0,i)}).$$

As mentioned in [SSH11], it is not necessary to include all $2r$ commitments in the transcript. Instead, we include a digest over the concatenation of all commitments $\sigma_0 = \mathcal{H}(c_{(0,0)} \| c_{(1,0)} \| \ldots \| c_{(0,r-1)} \| c_{(1,r-1)})$. We derive the challenges[6] $\alpha_i \in \mathbb{F}_{31}$ (for $0 \leq i < r$) by applying H_1 to $h_1 = (D, \sigma_0)$. Using these α_i, the vectors $\mathbf{t}_{(1,i)} = \alpha_i \cdot \mathbf{r}_{(0,i)} - \mathbf{t}_{(0,i)}$ and $\mathbf{e}_{(1,i)} = \alpha_i \cdot \mathbf{F}(\mathbf{r}_{(0,i)}) - \mathbf{e}_{(0,i)}$ can be computed.

Let $\sigma_1 = (\mathbf{t}_{(1,0)} \| \mathbf{e}_{(1,0)} \| \ldots \| \mathbf{t}_{(1,r-1)} \| \mathbf{e}_{(1,r-1)})$. We compute h_2 by applying H_2 to the tuple $(D, \sigma_0, h_1, \sigma_1)$ and use it as r binary challenges $\mathsf{ch}_{2,i} \in \{0, 1\}$.

Now we define $\sigma_2 = (\mathbf{r}_{(\mathsf{ch}_{2,i},i)}, \ldots, \mathbf{r}_{(\mathsf{ch}_{2,i},r-1)}, c_{1-\mathsf{ch}_{2,i}}, \ldots, c_{1-\mathsf{ch}_{2,r-1}})$. Note that here we also need to include the challenges $c_{1-\mathsf{ch}_{2,i}}$ that the verifier cannot recompute. We then output $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$ as the signature. At 5 bits per $\mathbb{F}_{31}$ element, the size of the signature is $(2 + r) \cdot k + 5 \cdot r \cdot (2 \cdot n + m)$ bits.

Verification. The verification algorithm takes as input the message m, the signature $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$ and the public key $PK = (\mathcal{S}_F, \boldsymbol{PK}_{\boldsymbol{v}})$. As above, we use R and m to compute D, and derive $\mathbf{F}$ from $\mathcal{S}_F$ using $G_{\mathcal{S}_F}$. As the signature contains σ_0, we can compose h_1 and, consequentially, the challenge values α_i for all r rounds by using H_1. Similarly, the values $\mathsf{ch}_{2,i}$ are computed by applying H_2 to $(D, \sigma_0, h_1, \sigma_1)$. For each round i, the verifier extracts vectors $\mathbf{t}_i$ and $\mathbf{e}_i$ (which are always $\mathbf{t}_{(1,i)}$ and $\mathbf{e}_{(1,i)}$) from σ_1 and $\mathbf{r}_i$ from σ_2. Depending on $\mathsf{ch}_{2,i}$, half of the commitments can now be computed:

$$\begin{aligned} &\text{if } \mathsf{ch}_{2,i} = 0 \qquad c_{(0,i)} = Com_0(\mathbf{r}_i, \alpha \cdot \mathbf{r}_i - \mathbf{t}_i, \alpha \cdot \mathbf{F}(\mathbf{r}_i) - \mathbf{e}_i) \\ &\text{if } \mathsf{ch}_{2,i} = 1 \qquad c_{(1,i)} = Com_1(\mathbf{r}_i, \alpha \cdot (\boldsymbol{PK}_{\boldsymbol{v}} - \mathbf{F}(\mathbf{r}_i)) - \mathbf{G}(\mathbf{t}_i, \mathbf{r}_i) - \mathbf{e}_i) \end{aligned}$$

Extracting the missing commitments $c_{(1-\mathsf{ch}_{2,i},i)}$ from σ_2, the verifier now computes $\sigma_0' = \mathcal{H}(c_{(0,0)} \| c_{(1,0)} \ldots \| c_{(0,r-1)} \| c_{(1,r-1)})$. For verification to succeed, $\sigma_0' = \sigma_0$ should hold.

5.3 Security of MQDSS

We now give a security reduction for MQDSS in the ROM. As our results from the last section are non-tight we only prove an asymptotic statement. While this does not suffice to make any statement about the security of a specific parameter choice, it provides evidence that the general approach leads a secure scheme. Also, the reduction is in the ROM, not in the QROM, thereby limiting applicability in the post-quantum setting. As already mentioned in the introduction, we consider it important future work to strengthen this statement.

In the remainder of this subsection we prove the following theorem.

Theorem 5.1. MQDSS *is EU-CMA-secure in the random oracle model, if*

- *the search version of the $\mathcal{MQ}$ problem is intractable,*
- *the hash functions $\mathcal{H}$, H_1, and H_2 are modeled as random oracles,*
- *the commitment functions Com_0 and Com_1 are computationally binding, computationally hiding, and the probability that their output takes a given value is negligible in the security parameter,*

[6] Note that the concatenation of all α_i was previously referred to as ch_1.

– *the pseudorandom generator $G_{\mathcal{S}_F}$ is modeled as random oracle, and*
– *the pseudorandom generators, G_{SK}, and G_c have outputs computationally indistinguishable from random.*

To prove this theorem we would like to apply Theorem 4.8. However, Theorem 4.8 was formulated for a slightly more generic construction. The point is that we apply an optimization originally proposed in [Ste96]. So, in our actual proposal, the parallel composition of the IDS is slightly different as, instead of the commitments, only the hash of their concatenation is sent. Also, the last message now contains the remaining commitments.

While we could have treated this case in Section 4, it would have limited the general applicability of the result, as the above optimization is only applicable to schemes with a certain, less generic, structure. However, it is straightforward to redo the proofs from Section 4 for the optimized scheme. When modeling the hash function used to compress the commitments as RO, the arguments are exactly the same with one exception. The proof of Lemma 4.12 uses that the commitment scheme – and thereby the first signature element σ_1 – only takes a given value with negligible probability. Now this statement follows from the same property of the commitment scheme and the randomness of the RO. Altogether this leads to the following corollary:

Corollary 5.2 (EU-CMA security of $q2$-signature schemes). *Let $k \in \mathbb{N}$, $\mathsf{IDS}(1^k)$ a $q2$-IDS that is honest-verifier zero-knowledge, achieves soundness with constant soundness error κ and has a $q2$-extractor. Then opt-$q2$-$\mathsf{Dss}(1^k)$, the optimized $q2$-signature scheme derived by applying Construction 4.7 and the optimization explained above, is existentially unforgeable under adaptive chosen message attacks.*

Based on this corollary we can now prove the above theorem.

Proof (of Theorem 5.1). Towards a contradiction, assume there exists an adversary $\mathcal{A}$ that wins the EU-CMA game against MQDSS with non-negligible success probability. We show that this implies the existence of an oracle machine $\mathcal{M}^{\mathcal{A}}$ that solves the $\mathcal{MQ}$ problem, breaks a property of one of the commitment schemes, or distinguishes the outputs of one of the pseudorandom generators from random. We first define a series of games and argue that the difference in success probability of $\mathcal{A}$ between these games is negligible. We assume that $\mathcal{M}$ runs $\mathcal{A}$ in these games.

Game 0: Is the EU-CMA game for MQDSS.
Game 1: Is Game 0 with the difference that $\mathcal{M}$ replaces the outputs of G_{SK} by random bit strings.
Game 2: Is Game 1 with the difference that $\mathcal{M}$ replaces the outputs of G_c by random bit strings.
Game 3: Is Game 2 with the difference that $\mathcal{M}$ takes as additional input a random equation system $\mathbf{F}$. $\mathcal{M}$ simulates $G_{\mathcal{S}_F}$ towards $\mathcal{A}$, programming $G_{\mathcal{S}_F}$ such that it returns the coefficients representing $\mathbf{F}$ upon input of $\mathcal{S}_F$ and uniformly random values on any other input.

Per assumption, $\mathcal{A}$ wins Game 0 with non-negligible success probability. Let's call this ϵ. If the difference in $\mathcal{A}$'s success probability playing Game 0 or Game 1 was non-negligible, we could use $\mathcal{A}$ to distinguish the outputs of G_{SK} from random. The same argument applies for the difference between Game 1 and Game 2, and G_c. Finally, the output distribution of $G_{\mathcal{S}_F}$ in Game 3 is the same as in previous games. Hence, there is no difference for $\mathcal{A}$ between Game 2 and Game 3. Accordingly, $\mathcal{A}$'s success probability in these two games is equal.

Now, Game 3 is exactly the EU-CMA game for the optimized $q2$ signature scheme that is derived from $\mathcal{MQ}$-IDS, the 5-pass IDS from [SSH11]. We obtain the necessary contradiction if we can apply Corollary 5.2. For this, it just remains to be shown that $\mathcal{MQ}$-IDS is a $q2$-IDS that is honest-verifier zero-knowledge, achieves soundness with constant soundness error κ and has a $q2$-extractor. Clearly, $\mathcal{MQ}$-IDS is a $q2$-IDS under the given assumptions on the commitment schemes. Sakumoto et al. [SSH11] show that $\mathcal{MQ}$-IDS is honest-verifier zero-knowledge. Theorem 3.1 shows that $\mathcal{MQ}$-IDS achieves soundness with constant soundness error $\kappa = \frac{q+1}{2q}$. Finally, the proof of Theorem 3.1 provides a construction of a $q2$-extractor. □

6 Instantiating the scheme

In this section, we provide a concrete instance of MQDSS. We discuss a suitable set of parameters to achieve the desired security level, discuss an optimized software implementation, and present benchmark results.

Parameter choice and security analysis. For the 5-pass scheme, the soundness error κ is affected by the size of q. This motivates a field choice larger than $\mathbb{F}_2$ in order to reduce the number of rounds required. From an implementation point of view, it is beneficial to select a small prime, allowing very cheap multiplications as well as comparatively cheap field reductions. We choose $\mathbb{F}_{31}$ with the intention of storing it in a 16 bit value – the benefits of which become clear in the next subsection, where we discuss the required reductions.

We now consider the choice of $\mathcal{MQ}(n, m, \mathbb{F}_{31})$, i.e. the parameters n and m. There are several known generic classical algorithms for solving systems of quadratic equations over finite fields, such as the F4 algorithm [Fau99] and the F5 algorithm [Fau02,BFS15] using Gröbner basis techniques, the Hybrid Approach [BFP09,BFP12] that is a variant of the F5 algorithm, or the XL algorithm [CKPS00,Die04] and variants [YC05a].

Currently, for fields $\mathbb{F}_q$ where $q \geqslant 4$, the best known technique for solving overdetermined systems of equations over $\mathbb{F}_q$ is combining equation solvers with exhaustive search. The Hybrid Approach [BFP09,BFP12] and the FXL variant of XL [YC05a] use this paradigm. Here we will analyze the complexity using the Hybrid approach. Note that the complexity for the XL family of algorithms is similar [YCY13].

Roughly speaking, for an optimization parameter ℓ, using the Hybrid approach one first fixes ℓ among the n variables, and then computes q^ℓ Gröbner bases of the smaller systems in $n - \ell$ variables. Hence, the improvement over the

plain F5 algorithm comes from the proper choice of the parameter ℓ. It has been shown in [BFP12] that the best trade-off is achieved when the parameter ℓ is proportional to the number of variables n, i.e when $\ell = \tau n$.

Let $2 \leqslant \omega \leqslant 3$ be the linear algebra constant. The complexity of computing a Gröbner basis of a system of m equations in n variables, $m \geqslant n$, using the F5 algorithm is given by

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

where $d_{reg}(n, m)$ is the degree of regularity of the system which can be approximated as

$$d_{reg}(n, m) \approx (\frac{m}{n} - \frac{1}{2} - \sqrt{\frac{m}{n}(\frac{m}{n} - 1)}) + \mathcal{O}\left(n^{1/3}\right).$$

For a fixed $0 < \tau < 1$, the complexity of the Hybrid approach is

$$C_{Hyb}(n, m, \tau, d_{reg}(n(1-\tau), m)) = q^{\tau n} \cdot C_{F5}(n(1-\tau), m, d_{reg,\tau}(n(1-\tau), m)).$$

It is well known (and can be seen from the complexity above) that the F5 algorithm as well as the Hybrid approach perform better when the number of equations is bigger than the number of variables, so from this point of view there is no incentive in choosing $m > n$. On the other hand, if $m < n$, then we can simply fix $n - m$ variables and reduce the problem to a smaller one, with m variables. Therefore, in terms of classical security the best choice is $m = n$.

Following the analysis from [BFP09,BFP12], we calculated the best trade-off for τ for the family of functions $\mathcal{MQ}(n, n, \mathbb{F}_{31})$, when $\omega = 2.3$. Asymptotically, $\tau \to 0.16$, although for smaller values of n (e.g $n = 32$) we find $\tau = 0.13$.

Since our goal is classical security of at least 128 bits, we need to choose $n \geq 51$, so that for any choice of the linear algebra constant $2 \leqslant \omega \leqslant 3$ the Hybrid approach would need at least 2^{128} operations. Note that if we set the more realistic value of $\omega = 2.3$, the minimum is $n = 45$.

For implementation reasons, we choose $n = 64$. In particular, a multiple of 16 suggests efficient register usage for vectorized implementations. In this case, for $\omega = 2.3$, the complexity of the Hybrid approach is $\approx 2^{177}$ and the best result is obtained for $\tau = 0.14$, which translates to fixing 9 variables in the system.

Regarding post-quantum security, at the moment there is no dedicated quantum algorithm for solving systems of quadratic equations. Instead, we can use Grover's search algorithm [Gro96] to directly attack the $\mathcal{MQ}$ problem, or use Grover's algorithm for the search part in a quantum implementation of the Hybrid method. Note that the later requires an efficient quantum implementation of the F5 algorithm, that we will assume provides no quantum speedup over the classical implementation.

Grover's algorithm searches for an item in a unordered list of size $N = 2^{\mathbf{n}}$ that satisfies a certain condition given in the form of a quantum black-box function $f : \{0, 1\}^{\mathbf{n}} \to \{0, 1\}$. If the condition is satisfied for the i-th item, then $f(i) = 1$,

otherwise $f(i) = 0$. The complexity of Grover's algorithm is $\mathcal{O}(\sqrt{N/M})$, where M is the number of items in the list that satisfy the condition, i.e. the algorithm provides a quadratic speed-up compared to classical search.

First we will consider a direct application of Grover's algorithm on the $\mathcal{MQ}$ problem in question. In this case, f should provide an answer whether a given n-tuple $\mathbf{x}$ from $\mathbb{F}_{31}^n$ satisfies the system of equations $\mathbf{F}(\mathbf{x}) = \mathbf{v}$. Since the domain is not Boolean, we need to convert it one, so we get a domain of size $n \log 31$.

To estimate the complexity of the algorithm, we need the number of solutions M to the given system of equations. Determining the exact M requires exponential time [Val79], but it was shown in [FB07] that the number of solutions of a system of n equations in n variables follows the Poisson distribution with parameter $\lambda = 1$. Therefore the expected value is 1. Furthermore, the probability that there are at least M solutions can be estimated as the tail probability of a Poisson random variable $P[X \geqslant M] \geqslant \frac{(e\lambda)^M}{e^\lambda M^M} = \frac{1}{e}(\frac{e}{M})^M$ which is negligible in M. In practice, we can safely assume that $M \leqslant 4$, since $P[M \geqslant 5] \geqslant 2^{-8}$. In total, Grover's algorithm takes $\mathcal{O}(2^{n \log 31/2}/4) \approx 2^{156}$ operations.

As said earlier, we can also use a quantum version of the Hybrid approach for $m = n$. In this case the complexity will be

$$C_{Hyb,quantum}(n, \tau, d_{reg}(n(1-\tau), n)) = \sqrt{\frac{q^{\tau n}}{M}} \cdot C_{F5}(n(1-\tau), n, d_{reg,\tau}(n(1-\tau), n)).$$

Taking again $M \leqslant 4$, the optimal value for the optimization parameter is $\tau = 0.39$, which means we should fix 25 variables in the system. Hence, the quantum version of the Hybrid method has a time complexity of $\approx 2^{139}$ operations.

To achieve EU-CMA for 128 bits of post-quantum security, we require that $k^r \leq 2^{-256}$, as an adversary could perform a preimage search to effectively control the challenges. As $\kappa = \frac{q+1}{2q}$ with $q = 31$, we need $r = 269$. To complete the scheme, we instantiate the functions $\mathcal{H}$, Com_0 and Com_1 with SHA3-256, and use SHAKE-128 for H_1, H_2, $G_{\mathcal{S}_F}$, G_c, and G_{SK} [BDPV11]. In order to convert between the output domain of SHAKE-128 and functions that map to vectors over $\mathbb{F}_{31}$, we simply reject and resample values that are not in $\mathbb{F}_{31}$ (effectively applying an instance of the second TSS08 construction from [WHCB13]).

We refer to this instance of the scheme as MQDSS-31-64.

Implementation. The central and most costly computation in this signature scheme is the evaluation of $\mathbf{F}$ (and, by corollary, $\mathbf{G}$). The signing procedure requires one evaluation of each for every round, and the verifier needs to compute either $\mathbf{F}$ (if $\mathsf{ch}_2 = 0$) or both $\mathbf{F}$ and $\mathbf{G}$ (if $\mathsf{ch}_2 = 1$), for each round. Other than these functions, the computational effort is made up of seed expansion, several hash function applications and a small number of additions and subtractions. For SHA3-256 and SHAKE-128, we rely on existing code from the Keccak Code Package [BDP+16]. Clearly, the focus for an optimized implementation should be on the $\mathcal{MQ}$ function. Previous work [CCC+09] has shown that modern CPUs offer interesting and valuable methods to efficiently implement this primitive, in particular by exploiting the high level of internal parallelism.

Compared to the binary 3-pass scheme, the implementation of the 5-pass scheme over $\mathbb{F}_{31}$ presents more challenges. As $\mathbb{F}_{31}$ does not have closure under regular integer multiplication and addition, results of computations need to be reduced to smaller representations. To avoid having to this too frequently, we generally represent field elements during computation as unsigned 16 bit values. During specific parts of the computation, we vary this representation as needed.

The evaluation of $\mathbf{F}$ can roughly be divided in two parts: the generation of all monomials, and computation of the resulting polynomials for known monomials. Generating the quadratic monomials based on the given linear monomials requires $n \cdot \frac{n+1}{2}$ multiplications. For the second part, we require $m \cdot (n + n \cdot \frac{n+1}{2})$ multiplications to multiply the coefficients of the system parameter with the quadratic monomials, as well as a number of additions to accumulate all results. As the second part is clearly more computationally intensive, the optimization of this part is our primary concern. We describe an approach for the monomial generation in Appendix B

To efficiently compute all polynomials for a given set of monomials, we keep all required data in registers to avoid the cost of register spilling throughout the computation. Given that $n = m = 64$, for this part of the computation we represent the 64 $\mathbb{F}_{31}$ input values as 8 bit values and the resulting 64 $\mathbb{F}_{31}$ elements as 16 bit values, costing us 2 and 4 `YMM` registers respectively. The coefficients of $\mathbf{F}$ can be represented as a column major matrix with every column containing all coefficients that correspond to a specific monomial, i.e. one for each output value. That would imply that every row of the matrix represents one polynomial of $\mathbf{F}$. In this representation, each result term is computed by accumulating the products of a row of coefficients with each monomial, which is exactly the same as computing the product of the matrix $\mathbf{F}$ and the vector containing all monomials. This allows us to efficiently accumulate the output terms, minimizing the required output registers.

In order to perform the required multiplications and additions as quickly as possible, we heavily rely on the AVX2 instruction `VPMADDUBSW`. In one instruction, this computes two 8 bit SIMD multiplications and a 16 bit SIMD addition. However, this instruction operates on 8 bit input values that are stored adjacently. This requires a slight variation on the representation of $\mathbf{F}$ described above: instead, we arrange the coefficients of $\mathbf{F}$ in a column major matrix with 16 bit elements, each corresponding to two concatenated monomials.

When arranging reductions, we must strike a careful balance between preventing overflow and not reducing more often than necessary. As we make extensive use of `VPMADDUBSW`, which takes both a signed and an unsigned operand to compute the quadratic monomials, we ensure that the input variables for the $\mathcal{MQ}$ function are unsigned values (in particular: $\{0, \ldots, 31\}$). For the coefficients in the system parameter $\mathbf{F}$, we can then freely assume the values are in $\{-15, \ldots, 15\}$, as these are the direct result of a pseudo-random generator. It turns out to be efficient to immediately reduce the quadratic monomials back to $\{0, \ldots, 31\}$ when they are computed. When we now multiply such a product with an element from the system parameter and add it to the accumulators, the

maximum value of each accumulator word will be at most[7] $64 \cdot 31 \cdot 15 = 29760$. As this does not exceed 32768, we only have to perform reductions on each individual accumulator at the very end.

One should note that [CCC+09] approaches this problem from a slightly different angle. In particular, they accumulate each individual output element sequentially, allowing them to keep the intermediate results in the 32 bit representation that is the output of their combined multiplication and addition instructions. This has the natural consequence of also avoiding early reductions.

Benchmark results. The MQDSS-31-64 implementation has been optimized for large Intel processors, supporting AVX2 instructions. Benchmarks were carried out on a single core of an Intel Core i7-4770K CPU, running at 3.5 GHz.

Signature and key sizes. The signature size of MQDSS-31-64 is considerably smaller than that of the 3-pass scheme. The obvious factor in this is the decreased ratio between the element size (which, in packed form, now require $64 \cdot 5 = 320$ bits each) and the number of rounds, resulting in a signature size of $2 \cdot 256 + 269 \cdot (256 + (5 \cdot 3 \cdot 64)) = 327\,616$ bits, or 40 952 bytes (39.99 KB). The shape of the keys does not change compared to 3-pass scheme, but since a vector of field elements now requires 320 bits, the public key is 72 bytes. The secret key remains 64 bytes.

Performance. As the $\mathcal{MQ}$ function is the most costly part of the computation, parameters are chosen in such a way that its performance is maximized. The required number of multiplications and additions (expressed as functions of n and m) does not change dramatically compared to the 3-pass baseline[8], but the actual values n and m are only a quarter of what they were. As the relation between n and m and the number of multiplications is quadratic for the monomials and cubic for the system parameter masking, and we see only a linear increase in the number of registers needed to operate on, the entire sequence of multiplications and additions becomes much cheaper. This especially impacts operations that involve the accumulators. As the representation allows us to keep reductions out of this innermost repeated loop, we perform (only) $\frac{67 \cdot 4}{2} + 4 = 136$ reductions[9] throughout the main computation and 66 when preparing quadratic monomials. As we were able to arrange the registers in such a way that they do not need to rotate across multiple registers, we greatly reduce the number of rotations required compared to the 3-pass scenario. Furthermore, we note that we use a total of $67 \cdot 16 \cdot 4 = 4288$ `VPMADDUBSW` instructions for the core computations.

For one iteration of the $\mathcal{MQ}$ function $\mathbf{F}$, we measure 6 616 cycles ($\mathbf{G}$ is slightly less costly, at 6 396 cycles). We measure a total of 8 510 616 cycles for the complete signature generation. Key generation costs 1 826 612 cycles, and verification consumes 5 752 612 cycles. On the given platform, that translates

[7] This follows from the fact that we combine 64 such monomials in two `YMM` registers.
[8] A slight difference is introduced by cancellation of the monomials in the $\mathbb{F}_2$ setting.
[9] This follows from the fact that we need a total of $\frac{64+64 \cdot 65}{2 \cdot 32} = 67$ `YMM` registers worth of space to store the monomials and perform 4 reductions after accumulating 2 `YMM` monomials.

to roughly 2.43 ms, 0.52 ms and 1.64 ms, respectively. Verification is expected to require on average $\frac{3}{2}$ calls to an $\mathcal{MQ}$ function per round, whereas signature generation always requires two. This explains the ratio; note that both signer and verifier incur additional costs besides the $\mathcal{MQ}$ functions, e.g. for seed expansion.

In order to compare these results to the state of the art, we consider the performance figures reported in [CCC+09]. In particular, we examine the Rainbow(31, 24, 20, 20) instance, as the 'public map' in this scheme is effectively the $\mathcal{MQ}$ function over $\mathbb{F}_{31}$ with $n = 64$, as used above. The number of equations differs (i.e. $m = 40$ as opposed to $m = 64$), but this can be approximated by normalizing linearly. In [CCC+09], the authors report a time measurement of $17.7\mu s$, which converts to 50 144 cycles on their 2.833 GHz Intel C2Q Q9550. After normalizing for m, this amounts to 80 230 cycles. Results from the eBACS benchmarking project further show that running the Rainbow verification function from [CCC+09] on a Haswell CPU requires approximately 46 520 cycles (and thus 74 432 after normalizing); verification is dominated by the public map. Using their (by now arguably outdated) SSE2-based code to evaluate a public map with $m = 64$ consumes 60 968 cycles on our Intel Core i7-4770K. All of these results provide confidence in the fact that our implementation, which makes extensive use of AVX2 instructions, is performing in line with expectations.

References

ABB+16. Sedat Akleylek, Nina Bindel, Johannes Buchmann, Juliane Krämer, and Giorgia Azzurra Marson. An efficient lattice-based signature scheme with provably secure instantiation. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology – AFRICACRYPT 2016*, volume 9646 of *LNCS*, pages 44–60. Springer, 2016. https://www.cdc.informatik.tu-darmstadt.de/fileadmin/user_upload/Group_CDC/An_Efficient_Lattice-Based_Signature_Scheme_with_Provably_Secure_Instantiation.pdf.

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/.

BCC+14. Daniel J. Bernstein, Tung Chou, Chitchanok Chuengsatiansup, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, and Christine van Vredendaal. How to manipulate curve standards: a white paper for the black hat. Cryptology ePrint Archive, Report 2014/571, 2014. https://eprint.iacr.org/2014/571.pdf.

BDP+16. Guido Bertoni, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Keccak Code Package, 2016. http://keccak.noekeon.org, Retrieved on April 13, 2016.

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

BFP09. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, pages 177–197, 2009. http://www-polsys.lip6.fr/~jcf/Papers/JMC2.pdf.

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, *ISSAC'12 – Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, 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.

BG14. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In Josh Benaloh, editor, *Topics in Cryptology – CT-RSA 2014*, volume 8366 of *LNCS*, pages 28–47. Springer, 2014. https://eprint.iacr.org/2013/838.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.

CCC+09. Anna Inn-Tung Chen, Ming-Shing Chen, Tien-Ren Chen, Chen-Mou Cheng, Jintai Ding, Eric Li-Hsiang Kuo, Frost Yu-Shuang Lee, and Bo-Yin Yang. SSE implementation of multivariate PKCs on modern x86 CPUs. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems – CHES 2009*, volume 5747 of *LNCS*, pages 33–48. Springer, 2009. https://www.iacr.org/archive/ches2009/57470031/57470031.pdf.

CGP. Nicolas Courtois, Louis Goubin, and Jacques Patarin. SFLASH, a fast asymmetric signature scheme for low-cost smartcards - primitive specification and supporting documentation. http://www.minrank.org/sflash-b-v2.pdf.

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.

CVE10. Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 171–186. Springer, 2010. https://hal-univ-tln.archives-ouvertes.fr/hal-00674249/document.

DBG+15. Özgür Dagdelen, Rachid El Bansarkhani, Florian Göpfert, Tim Güneysu, Tobias Oder, Thomas Pöppelmann, Ana Helena Sánchez, and Peter Schwabe. High-speed signatures from standard lattices. In Diego F. Aranha and Alfred Menezes, editors, *Progress in Cryptology – LATINCRYPT 2014*, volume 8895 of *LNCS*, pages 84–123. Springer, 2015. https://cryptojedi.org/papers/#lwesign.

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013. https://eprint.iacr.org/2013/383/.

DFSS07. Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical cryptanalysis of SFLASH. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *LNCS*, pages 1–12. Springer, 2007. https://eprint.iacr.org/2007/14.

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, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *LNCS*, pages 323–337. Springer, 2004. https://www.iacr.org/archive/asiacrypt2004/33290320/33290320.pdf.

DS05. Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, volume 3531 of *LNCS*, pages 164–175. Springer, 2005. https://www.semanticscholar.org/paper/Rainbow-a-New-Multivariable-Polynomial-Signature-Ding-Schmidt/7977afcdb8ec9c420935f7a1f8212c303f0ca7fb/pdf.

Duc14. Léo Ducas. Accelerating Bliss: the geometry of ternary polynomials. Cryptology ePrint Archive, Report 2014/874, 2014. http://eprint.iacr.org/2014/874/.

EDV+12. Sidi Mohamed El Yousfi Alaoui, Özgür Dagdelen, Pascal Véron, David Galindo, and Pierre-Louis Cayrel. Extended security arguments for signature schemes. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *Progress in Cryptology – AFRICACRYPT 2012*, volume 7374 of *LNCS*, pages 19–34. Springer, 2012. https://eprint.iacr.org/2012/068/.

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*, pages 75–83. ACM, 2002. http://www-polsys.lip6.fr/~jcf/Papers/F02a.pdf.

FB07. Giordano Fusco and Eric Bach. Phase transition of multivariate polynomial systems. In Jin-Yi Cai, Barry S. Cooper, and Hong Zhu, editors, *International Conference on Theory and Applications of Models of Computation – TAMC 2007*, volume 4484 of *LNCS*, pages 632–645. Springer, 2007. https://minds.wisconsin.edu/bitstream/handle/1793/60544/TR1588.pdf.

FGP+15. Jean-Charles Faugère, Danilo Gligoroski, Ludovic Perret, , Simona Samardjiska, and Enrico Thomae. A polynomial-time key-recovery attack on MQQ cryptosystems. In Jonathan Katz, editor, *Public-Key Cryptography – PKC 2015*, volume 9020 of *LNCS*, pages 150–174. Springer, 2015. https://eprint.iacr.org/2014/811.

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.

GHLY16. Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, Gauss, and Reload – a cache attack on the BLISS lattice-based signature scheme. Cryptology ePrint Archive, Report 2014/874, 2016. http://eprint.iacr.org/2016/300/.

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.

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.

GØJ+11. Danilo Gligoroski, Rune S. Ødegård, Rune E. Jensen, Ludovic Perret, Jean-Charles Faugère, Svein Johan Knapskog, and Smile Markovski. MQQ-SIG - an ultra-fast and provably CMA resistant digital signature scheme. In Liqun Chen, Moti Yung, and Liehuang Zhu, editors, *Trusted Systems*, volume 7222 of *LNCS*, pages 184–203. Springer, 2011. https://hal.inria.fr/hal-00778083/document.

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.

HK06. Shai Halevi and Hugo Krawczyk. Strengthening digital signatures via randomized hashing. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *LNCS*, pages 41–59. Springer, 2006. https://www.iacr.org/archive/crypto2006/41170039/41170039.pdf.

IBM16. IBM. IBM makes quantum computing available on IBM cloud to accelerate innovation, 2016. https://www-03.ibm.com/press/us/en/pressrelease/49661.wss.

KPG99. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *LNCS*, pages 206–222. Springer, 1999. http://www.goubin.fr/papers/OILLONG.PDF.

KS98. Aviad Kipnis and Adi Shamir. Cryptanalysis of the Oil & Vinegar signature scheme. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO '98*, volume 1462 of *LNCS*, pages 257–266. Springer,

1998. http://antoanthongtin.vn/Portals/0/UploadImages/kiennt2/KyYeu/DuLieuNuocNgoai/3.Advances%20in%20cryptology-Crypto%201998-LNCS%201462/14620257.PDF.

MKF+16. David McGrew, Panos Kampanakis, Scott Fluhrer, Stefan-Lukas Gazdag, Denis Butin, and Johannes Buchmann. State management for hash based signatures. Cryptology ePrint Archive, Report 2016/357, 2016. https://eprint.iacr.org/2016/357.

NIS16. NIST. Post-quantum cryptography: NIST's plan for the future, 2016. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/pqcrypto-2016-presentation.pdf.

NSA. NSA. NSA suite B cryptography. https://www.nsa.gov/ia/programs/suiteb_cryptography/, Updated on August 19, 2015.

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.

Pat97. Jacques Patarin. The Oil and Vinegar signature scheme. In *Dagstuhl Workshop on Cryptography*, 1997.

PCG01. Jacques Patarin, Nicolas Courtois, and Louis Goubin. QUARTZ, 128-bit long digital signatures. In David Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *LNCS*, pages 282–297. Springer, 2001. http://www.goubin.fr/papers/rsa2001b.pdf.

PCY+15. Albrecht Petzoldt, Ming-Shing Chen, Bo-Yin Yang, Chengdong Tao, and Jintai Ding. Design principles for HFEv- based multivariate signature schemes. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, volume 9452 of *LNCS*, pages 311–334. Springer, 2015. http://www.iis.sinica.edu.tw/papers/byyang/19342-F.pdf.

PP03. David Pointcheval and Guillaume Poupard. A new NP-complete problem and public-key identification. *Designs, Codes and Cryptography*, 28(1):5–31, 2003.

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.

Ste93. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO '93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993. https://www.di.ens.fr/~stern/data/St47.pdf.

Ste96. Jacques Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996. https://www.di.ens.fr/users/stern/data/St55b.pdf.

TGTF10. Shigeo Tsujii, Masahito Gotaishi, Kohtaro Tadaki, and Ryou Fujita. Proposal of a signature scheme based on STS trapdoor. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, volume 6061 of *LNCS*, pages 201–217. Springer, 2010. https://eprint.iacr.org/2010/118.

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.

TW12. Enrico Thomae and Christopher Wolf. Cryptanalysis of enhanced TTS, STS and all its variants, or: Why cross-terms are important. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *Progress in Cryptology – AFRICACRYPT 2012*, volume 7374 of *LNCS*, pages 188–202. Springer, 2012.

Val79. Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

WHCB13. Patrick Weiden, Andreas Hülsing, Daniel Cabarcas, and Johannes Buchmann. Instantiating treeless signature schemes. Cryptology ePrint Archive, Report 2013/065, 2013. http://eprint.iacr.org/2013/065/.

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.

YC05a. 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.

YC05b. Bo-Yin Yang and Jiun-Ming Chen. Building secure tame-like multivariate public-key cryptosystems: The new TTS. In Colin Boyd and Juan Manuel González Nieto, editors, *Information Security and Privacy*, volume 3574 of *LNCS*, pages 518–531. Springer, 2005. http://www.iis.sinica.edu.tw/papers/byyang/2381-F.pdf.

YCC04a. Bo-Yin Yang, Jiun-Ming Chen, and Yen-Hung Chen. TTS: High-speed signatures on a low-cost smart card. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *LNCS*, pages 371–385. Springer, 2004. https://www.iacr.org/archive/ches2004/31560371/31560371.pdf.

YCC04b. Bo-Yin Yang, Jiun-Ming Chen, and Nicolas Courtois. On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis. In Javier Lopez, Sihan Qing, and Eiji Okamoto, editors, *Information and Communications Security*, volume 3269 of *LNCS*, pages 401–413. Springer, 2004. http://www.iis.sinica.edu.tw/papers/byyang/2384-F.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*, pages 19–33. Springer, 2013. http://www.iis.sinica.edu.tw/papers/byyang/17377-F.pdf.

A The 3-pass scheme over $\mathbb{F}_2$

In this appendix, we discuss the result of applying the Fiat-Shamir transform to the 3-pass IDS introduced in [SSH11] (see Figure 3).

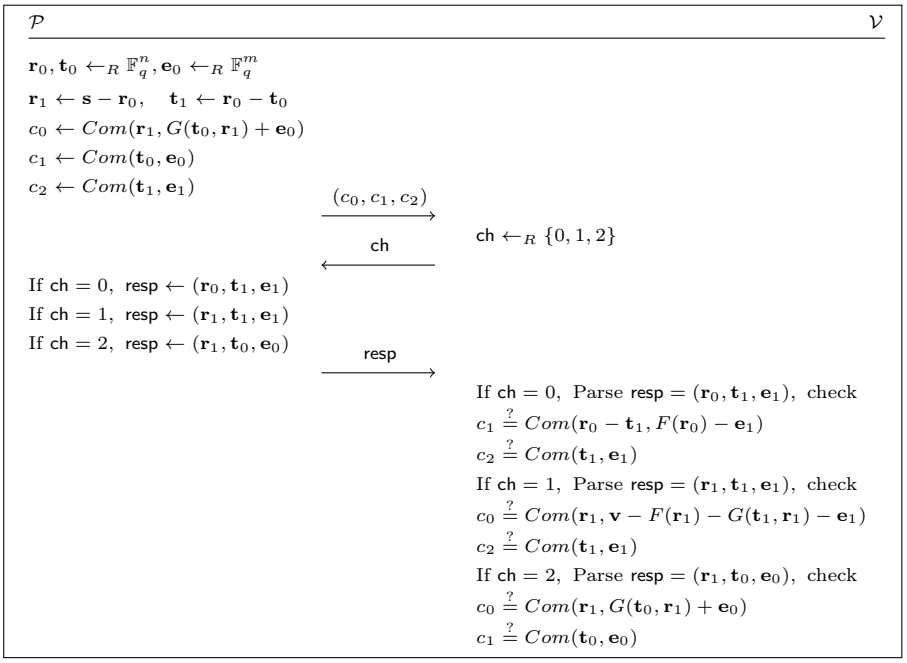

Fig. 3. Sakumoto et al. 3-pass IDS.

The signature scheme. As the resulting scheme is similar to the transformed 5-pass scheme (MQDSS) described in Section 5 in many ways, we occasionally refer back to that description to prevent needless redundancy and duplication.

Parameters. The scheme is parameterized by $k, m, n \in \mathbb{N}$ in much the same way as MQDSS. Whereas the 5-pass variant required several functions that ranged over $\mathbb{F}_{31}$, however, a slightly simpler setup suffices for the 3-pass scheme. We require the following functions:

- Cryptographic hash functions $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ and $H_1 : \{0,1\}^{2k} \to \{0,1,2\}^r$,
- a string commitment function $Com : {\mathbb{F}_2}^n \times {\mathbb{F}_2}^m \to \{0,1\}^k$,
- pseudo-random generators $G_F : \{0,1\}^k \to {\mathbb{F}_2}^{F_{len}}$ and $G_c : \{0,1\}^{2k} \to {\mathbb{F}_2}^{r \cdot (2n+m)}$.

Key generation. As before, we randomly sample $SK \leftarrow_R \{0,1\}^k$ and $\mathcal{S}_F \leftarrow_R \{0,1\}^k$. Given parameters $n, m \in \mathbb{N}$, we expand $\mathcal{S}_F$ to obtain $\mathbf{F} = G_{\mathcal{S}_F}(\mathcal{S}_F)$, a random system from $\mathcal{MQ}(n, m, \mathbb{F}_2)$. The equation system $\mathbf{F}$ is defined by F_{len} elements[10] from $\mathbb{F}_2$. We then apply $\mathbf{F}$ to SK to obtain the rest of the

[10] As we are computing over $\mathbb{F}_2$, we have $x_i \cdot x_i = x_i \wedge x_i = x_i$. This allows us to merge the linear monomial terms into the quadratics, reusing $n \cdot m$ field elements in $\mathbf{F}$.

public key, $\boldsymbol{PK_v} = \mathbf{F}(SK)$. The key generation algorithm outputs $(\mathsf{sk}, \mathsf{pk}) = ((SK, \mathcal{S}_F), (\mathcal{S}_F, \boldsymbol{PK_v}))$ as the key pair.

Signing. The signature algorithm takes as input a message $m \in \{0,1\}^*$ and a secret key $\mathsf{sk} = (SK, \mathcal{S}_F)$. The message-dependent randomness R and randomized message digest D are derived in the same way as was done for the 5-pass scheme, and $\mathbf{F} = G_{\mathcal{S}_F}(\mathcal{S}_F)$ as was done during key generation. As before, let r be the required number of rounds.

The pair (SK, D) is expanded using G_c to produce the values $(\mathbf{r}_{(0,0)}, \ldots, \mathbf{r}_{(0,r)}, \mathbf{t}_{(0,0)}, \ldots, \mathbf{t}_{(0,r)}, \mathbf{e}_{(0,0)}, \ldots, \mathbf{e}_{(0,r)})$. We then compute $c_{(0,i)}$, $c_{(1,i)}$ and $c_{(2,i)}$ using the string commitment function Com, as follows:

$$\begin{aligned} c_{(0,i)} &= Com(\mathbf{r}_{(1,i)}, \mathbf{G}(\mathbf{t}_{(0,i)}, \mathbf{r}_{(1,i)}) + \mathbf{e}_{(0,i)}) \\ c_{(1,i)} &= Com(\mathbf{t}_{(0,i)}, \mathbf{e}_{(0,i)}) \\ c_{(2,i)} &= Com(\mathbf{t}_{(1,i)}, \mathbf{e}_{(1,i)}) \end{aligned}$$

Furthermore, $\sigma_0 = \mathcal{H}(c_{(0,0)} \| c_{(1,0)} \| c_{(2,0)} \| \ldots \| c_{(0,r-1)} \| c_{(1,r-1)} \| c_{(2,r-1)})$.

We now derive the challenges h_1 from the pair (D, σ_0) using H_1. Then, for each of the rounds, we include the responses for each challenge ch_i in σ_1 (i.e. $(\mathbf{r}_{(0,i)}, \mathbf{t}_{(1,i)}, \mathbf{e}_{(1,i)})$, $(\mathbf{r}_{(1,i)}, \mathbf{t}_{(1,i)}, \mathbf{e}_{(1,i)})$ or $(\mathbf{r}_{(1,i)}, \mathbf{t}_{(0,i)}, \mathbf{e}_{(0,i)})$, respectively), as well as the one commitment the verifier cannot recompute: c_{ch_i}. The resulting signature is $\sigma = (R, \sigma_0, \sigma_1)$, for a total of $2 \cdot k + r \cdot (2 \cdot n + m + k)$ bits.

Verification. As above, the verifier uses R and m to compute D, deriving $\mathbf{F}$ from $\mathcal{S}_F$ (which is available in pk) using $G_{\mathcal{S}_F}$.

The verification algorithm takes as input the message m, the signature $\sigma = (R, \sigma_0, \sigma_1)$ and the public key $PK = (\mathcal{S}_F, \boldsymbol{PK_v})$. Again, we compute $\mathbf{F} = G_{\mathcal{S}_F}(\mathcal{S}_F)$ and the randomized digest $D = \mathcal{H}(R \parallel m)$. As was done in the signing procedure, the verifier can now derive ch_i for all r rounds using H_1 and the pair (D, σ_0). The verifier then extracts $(\mathbf{r}_i, \mathbf{t}_i, \mathbf{e}_i)$ from σ_1, and, depending on the values of ch_i, computes two thirds of the committed values as follows:

$$\begin{aligned} \text{if } \mathsf{ch}_i = 0 &\begin{cases} c_{(1,i)} = Com(\mathbf{r}_i - \mathbf{t}_i, \mathbf{F}(\mathbf{r}_i) - \mathbf{e}_i) \\ c_{(2,i)} = Com(\mathbf{t}_i, \mathbf{e}_i) \end{cases} \\ \text{if } \mathsf{ch}_i = 1 &\begin{cases} c_{(0,i)} = Com(\mathbf{r}_i, \boldsymbol{PK_v} - \mathbf{F}(\mathbf{r}_i) - \mathbf{G}(\mathbf{t}_i, \mathbf{r}_i) - \mathbf{e}_i) \\ c_{(2,i)} = Com(\mathbf{t}_i, \mathbf{e}_i) \end{cases} \\ \text{if } \mathsf{ch}_i = 2 &\begin{cases} c_{(0,i)} = Com(\mathbf{r}_i, \mathbf{G}(\mathbf{t}_i, \mathbf{r}_i) + \mathbf{e}_i) \\ c_{(1,i)} = Com(\mathbf{t}_i, \mathbf{e}_i) \end{cases} \end{aligned}$$

For each round, the other commitments $c_{(\mathsf{ch}_i, i)}$ can be extracted from σ_1, allowing the verifier to compute $\sigma_0' = \mathcal{H}(c_{(0,0)} \| c_{(1,0)} \| c_{(2,0)} \| \ldots \| c_{(0,r-1)} \| c_{(1,r-1)} \| c_{(2,r-1)})$. If $\sigma_0' = \sigma_0$, verification succeeds.

Parameter choice and security analysis. As in the case of the 5-pass scheme we motivate our choice of parameters both from security and from implementation point of view. First of all, the security arguments for $n = m$ are the same as for the 5-pass scheme. From an implementation point of view, choosing $n = m$ as a power of 2 provides various benefits. Most notably, the fact that we operate

over a binary field in combination with a number of elements that neatly fit typical registers greatly enhances the ease of using bitwise operations. Therefore, we choose $n = 256$.

In terms of classical security, we can use the known generic algorithms for solving systems of quadratic equations [Fau99,Fau02,CKPS00], or the recently proposed algorithm BooleanSolve [BFSS13], crafted specifically for the Boolean case. This algorithm performs similar to XL over $\mathbb{F}_2$. Indeed, the analysis in [YC04,YCC04b] and that of [BFSS13] both show that for large enough systems (i.e. where $n \geqslant 200$) it is possible to outperform exhaustive search in terms of time complexity. When $n = m$, the asymptotic complexity of the FXL algorithm is $\mathcal{O}(2^{0.875n})$ [YC04], and of the BooleanSolve algorithm it is $\mathcal{O}(2^{0.841n})$ in the deterministic variant and $\mathcal{O}(2^{0.792n})$ in the probabilistic variant [BFSS13]. Since we have $n = 256$, the probabilistic variant of BooleanSolve, for example, is expected to perform better than exhaustive search, but not bellow 2^{202} Boolean operations, obtained from the asymptotic estimate.

Using the same reasoning as for the 5-pass scheme we have that the expected number of solutions of the system is 1 and with great probability it is $\leqslant 4$. Under this assumption, the complexity of Grover's algorithm will be $\approx 2^{128}$ operations.

Similarly, as was mentioned in Section 6, we must have $\kappa^r \leq 2^{-256}$ in order to achieve EU-CMA for 128 bits of post-quantum security. Thus, for soundness error $\kappa = \frac{2}{3}$, we must perform $r = 438$ rounds.

As before, we choose SHA3-256 for the functions $\mathcal{H}$ and Com. For H_1, $G_{\mathcal{S}_F}$ and G_c, we select the SHAKE-128 extendable output function [BDPV11]. To map to the appropriate output domain, we reject and resample unusable output from H_1.

Implementation. As briefly mentioned above, we operate on $n = 256$ elements from $\mathbb{F}_2$, which means that an input vector $\mathbf{x}$ fits precisely in one 256 bit SIMD vector register. In essence, the computation of $\mathbf{F}$ comes down to computing the product of all unique pairs of elements in $\mathbf{x}$. Intuitively, one might visualize such a multiplication as the triangle depicted in Figure 4 (note that we do not incur any costs for register-wide gaps in the arrangement). As we are using vector registers, we can perform 256 such multiplications (which are bitwise AND operations in $\mathbb{F}_2$) in parallel. To do this, we rearrange the products in a way that makes them more easily computable – see Figure 5.

The careful reader will notice that the elements as arranged in Figure 5 can be generated by rotating the original vector and computing the bitwise AND with the original after each rotation. Thus we require only $\frac{n}{2} + 1$ vectorized AND operations (one of which is the last half-row). After computing a row of products, the result row can be multiplied with m bits from the system parameter for each of the output bits.

To compute $\mathbf{G}(\mathbf{x}, \mathbf{y})$, one can iterate through two instances of Figure 5 in parallel; one for each input. Computing $a_{i,j,l} \wedge ((x_i \wedge y_j) \oplus (x_j \wedge y_i))$ (where $a_{i,j,l}$ is an element of $\mathbf{F}$) is then a matter of computing a crosswise XOR before masking with system parameter bits. The first row can be skipped, as $(x_i \wedge y_i) \oplus (x_i \wedge y_i)$, is always 0.

&	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	-	11	12	13	14	15	16	17
2	-	-	22	23	24	25	26	27
3	-	-	-	33	34	35	36	37
4	-	-	-	-	44	45	46	47
5	-	-	-	-	-	55	56	57
6	-	-	-	-	-	-	66	67
7	-	-	-	-	-	-	-	77

Fig. 4. Intuitive AND of 8-bit vector.

&	0	1	2	3	4	5	6	7
	00	11	22	33	44	55	66	77
	10	21	32	43	54	65	76	07
	20	31	42	53	64	75	06	17
	30	41	52	63	74	05	16	27
	40	51	62	73	-	-	-	-

Fig. 5. Efficient AND of 8-bit vector.

Benchmark results. As mentioned in Section 5.1, benchmarks were performed on an Intel Core i7-4770K CPU at 3.5 GHz.

Signature and key sizes. For the 3-pass scheme, the signature size is fairly large because of the high number of rounds required to achieve 128 bit post-quantum security, as well as the large n and m. With each vector consuming 256 bits and each of the 438 rounds adding three vectors and a commitment hash, the total amounts to $2 \cdot 256 + 438 \cdot (3 \cdot 256 + 256) = 449\,024$ bits, or 54.81 KB. Both the secret key and the public key are 64 bytes.

Performance. As one would expect, the calls to the $\mathcal{MQ}$ function are the most time-consuming. Theoretically, we can calculate that we require $n \cdot \frac{n+1}{2}$ AND operations to compute all quadratic monomials in a call to $\mathbf{F}$. For each of the m output bits, the results are masked with bits from the system parameter, incurring another $m \cdot n \cdot \frac{n+1}{2}$ AND operations, as well as $m \cdot (n \cdot \frac{n+1}{2} - 1)$ XOR operations. Using vector instructions, the most optimal scenario would allow us to do 256 such operations in parallel. While this is almost achieved (although not fully for the half-row of monomials, and because the accumulators used to compute the final result need to be folded onto themselves at the end), this is offset greatly by the amount of loads and stores required to manage the 256 accumulators for the output results. Some additional costs are also incurred for having to rotate the vector register for every 256 monomials, at the cost of two shifts, a double-quadword permute and an OR operation. Costs of the application of the function $\mathbf{G}$ are similar, although distributed slightly differently: each 'binomial' costs two AND operations and a XOR operation, and two vectors need to be rotated, but as the symmetric monomials can be omitted, one entire iteration of updating accumulators can be spared.

For one iteration of the $\mathcal{MQ}$ function, we measure 122 564 cycles (again, $\mathbf{G}$ is marginally cheaper: 121 928 cycles), while we measure 118 088 992 cycles for the complete signature generation. Key generation comes in at 8 066 324 cycles, and verification costs 82 650 156 cycles. On the used CPU, that comes down to 33.7 ms, 2.30 ms and 23.6 ms, respectively. On average, verification should require $1\frac{1}{3}$ calls to an $\mathcal{MQ}$ function, varying with the challenge value.

B Generating quadratic monomials in the 5-pass scheme

For the initial part of the computation, we represent the monomials as 16 bit values (as this helps prevent reductions). In the 3-pass case, we rotated an `YMM` vector register to efficiently compute all quadratic monomials (see Appendix A, in particular Figure 4 and 5). While that was already expensive, rotating four `YMM` registers as if it were one 1024 bit value is typically much more costly. Additionally, storing both the original and rotated state would require many registers. Instead, one can arrange the products in such a way that the blocks of 16 elements do not need to be mixed, but can each be rotated individually and multiplied with the unrotated originals. This is especially beneficial when computing $\mathbf{G}$, in which we process 8 blocks of 256 bits. Two caveats appear in the first and last rows: duplicates need to be avoided when unrotated originals are combined in the first row, while the last (half-)row needs to be used to produce missing products by pairwise multiplying the high half of each register with every low half. See Figure 6 for an intuition of this arrangement with 4 registers containing 4 field elements. The software that is part of this work (see "Availability of the software" in Section 1) includes a script that generates this arrangement.

&	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	00	11	22	33	04	15	26	37	08	19	2A	3B	0C	1D	2E	3F
	10	21	32	03	14	25	36	07	18	29	3A	0B	1C	2D	3E	0F
	-	-	-	-	44	55	66	77	48	59	6A	7B	4C	5D	6E	7F
	50	61	72	43	54	65	76	47	58	69	7A	4B	5C	6D	7E	4F
	-	-	-	-	-	-	-	-	88	99	AA	BB	8C	9D	AE	BF
	90	A1	B2	83	94	A5	B6	87	98	A9	BA	8B	9C	AD	BE	8F
	-	-	-	-	-	-	-	-	-	-	-	-	CC	DD	EE	FF
	D0	E1	F2	C3	D4	E5	F6	C7	D8	E9	FA	CB	DC	ED	FE	CF
	02	13	-	-	42	53	-	-	82	93	-	-	C2	D3	-	-
	06	17	-	-	46	57	-	-	86	97	-	-	C6	D7	-	-
	0A	1B	-	-	4A	5B	-	-	8A	9B	-	-	CA	DB	-	-
	0E	1F	-	-	4E	5F	-	-	8E	9F	-	-	CE	DF	-	-

Fig. 6. Efficient arrangement of AND of 4 registers with 4 $\mathbb{F}_{31}$ elements each.