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

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

[1] National Taiwan University / Academia Sinica, Taipei, Taiwan
[2] Technische Universiteit Eindhoven, Eindhoven, The Netherlands
[3] Radboud University, Nijmegen, The Netherlands
[4] "Ss. Cyril and Methodius" University, Skopje, R. Macedonia

2018-02-01

Tenerife

$\mathcal{MQ}$-based signatures

- Important candidate for post-quantum signatures
- Several submissions to NIST
  - DualModeMS [FPR17], GeMSS [CFMR+17], Gui [PCY+15, DCP+17a], HiMQ-3 [SPK17], LUOV [BPSV17], MQDSS [CHR+16, CHR+17], Rainbow [DS05, DCP+17b]
- Traditionally small signatures, larger keys
  - (except DualModeMS, LUOV, MQDSS)

$\mathcal{MQ}$-based signatures

- Important candidate for post-quantum signatures
- Several submissions to NIST
  - DualModeMS [FPR17], GeMSS [CFMR+17], Gui [PCY+15, DCP+17a], HiMQ-3 [SPK17], LUOV [BPSV17], MQDSS [CHR+16, CHR+17], Rainbow [DS05, DCP+17b]
- Traditionally small signatures, larger keys
  - (except DualModeMS, LUOV, MQDSS)
- Typically based on $\mathcal{MQ}$ but also related problems (e.g. IP)
  - MQDSS: (lossy) ROM reduction to $\mathcal{MQ}$

$\mathcal{MQ}$-based signatures

- Important candidate for post-quantum signatures
- Several submissions to NIST
  - DualModeMS [FPR17], GeMSS [CFMR+17], Gui [PCY+15, DCP+17a], HiMQ-3 [SPK17], LUOV [BPSV17], MQDSS [CHR+16, CHR+17], Rainbow [DS05, DCP+17b]
- Traditionally small signatures, larger keys
  - (except DualModeMS, LUOV, MQDSS)

- Typically based on $\mathcal{MQ}$ but also related problems (e.g. IP)
  - MQDSS: (lossy) ROM reduction to $\mathcal{MQ}$

- SOFIA: continue in line of MQDSS
  - Transform an $\mathcal{MQ}$-based IDS

Why not Fiat-Shamir?

- Non-tight proof in the ROM
- No proof in the QROM
  - Forking lemma $\Rightarrow$ rewinding adversary

Why not Fiat-Shamir?

- Non-tight proof in the ROM
- No proof in the QROM
  - Forking lemma $\Rightarrow$ rewinding adversary

- .. at the time of writing
- Lots of ongoing work!

Why not Fiat-Shamir?

- Non-tight proof in the ROM
- No proof in the QROM
  - Forking lemma $\Rightarrow$ rewinding adversary

- .. at the time of writing
- Lots of ongoing work!
- [KLP17]: tight Fiat-Shamir in the ROM
  - But similar issues in the QROM
- [KLS17]: Fiat-Shamir in QROM
  - Requires changing the IDS and parameters

This work

1. Extend Unruh's transform [Unr15] to 5-pass IDS
   - Specifically q2-IDS [CHR+16]

This work

1. Extend Unruh's transform [Unr15] to 5-pass IDS
   - Specifically q2-IDS [CHR+16]
2. Prove EU-CMA security in QROM
   - Via a (tight) proof in ROM

This work

1. Extend Unruh's transform [Unr15] to 5-pass IDS
   - Specifically $q2$-IDS [CHR+16]
2. Prove EU-CMA security in QROM
   - Via a (tight) proof in ROM
3. Instantiate and tweak for specific IDS [SSH11]

This work

1. Extend Unruh's transform [Unr15] to 5-pass IDS
   - Specifically $q2$-IDS [CHR+16]
2. Prove EU-CMA security in QROM
   - Via a (tight) proof in ROM
3. Instantiate and tweak for specific IDS [SSH11]
4. Parameterize to achieve 128-bit post-quantum
   - SOFIA-4-128

This work

1. Extend Unruh's transform [Unr15] to 5-pass IDS
   - Specifically q2-IDS [CHR+16]
2. Prove EU-CMA security in QROM
   - Via a (tight) proof in ROM
3. Instantiate and tweak for specific IDS [SSH11]
4. Parameterize to achieve 128-bit post-quantum
   - SOFIA-4-128
5. Implement and compare on AVX2

Canonical Identification Schemes

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

Informally:

1. Prover commits to some (randomized) value derived from sk
2. Verifier picks a challenge ‘ch’
3. Prover computes response ‘resp’
4. Verifier checks if response matches challenge

Security of the IDS

- Passively secure IDS

Soundness: the probability that an adversary can convince is 'small'

Honest-Verifier Zero-Knowledge: simulator can 'fake' transcripts

Special soundness: two 'similar' transcripts $\Rightarrow$ secret exposed

Security of the IDS

- Passively secure IDS

Soundness: the probability that an adversary can convince is 'small'

- Adversary $\mathcal{A}$ can 'guess right': soundness error κ

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

Honest-Verifier Zero-Knowledge: simulator can 'fake' transcripts

- Shows transcripts do not leak the secret

Special soundness: two 'similar' transcripts $\Rightarrow$ secret exposed

- Proof relies on constructing an 'extractor'

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges
  3. Apply length-preserving hash $\Rightarrow$ "blind" responses

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]

- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges
  3. Apply length-preserving hash $\Rightarrow$ "blind" responses
  4. Sample challenge vector

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges
  3. Apply length-preserving hash $\Rightarrow$ "blind" responses
  4. Sample challenge vector
  5. Reveal one response per commit

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges
  3. Apply length-preserving hash $\Rightarrow$ "blind" responses
  4. Sample challenge vector
  5. Reveal one response per commit
- In the proof, "blinding" is an invertible permutation
  - Adversary must have known several transcripts
  - Unblinding makes them available to extractor

Unruh's transform [Unr15]

- Based on Fischlin's transform [Fis05]
- Informally:
  1. Generate transcripts for a commit
  2. Iterate for multiple challenges
  3. Apply length-preserving hash $\Rightarrow$ "blind" responses
  4. Sample challenge vector
  5. Reveal one response per commit
- In the proof, "blinding" is an invertible permutation
  - Adversary must have known several transcripts
  - Unblinding makes them available to extractor
- Parallelize r rounds to decrease error
- Extra parameter: prepare for t challenges

Canonical Identification Schemes

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

5-pass q2 Identification Schemes

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow_R \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow_R \mathbb{F}_q$
$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \alpha)$	$\xrightarrow{\mathsf{resp}_1}$	
	$\xleftarrow{\mathsf{ch}_2}$	$\mathsf{ch}_2 \leftarrow_R \{0, 1\}$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \alpha, \mathsf{resp}_1, \mathsf{ch}_2)$	$\xrightarrow{\mathsf{resp}_2}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \alpha, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

5-pass q2 Identification Schemes

$\mathcal{P}$		$\mathcal{V}$
$\mathsf{com} \leftarrow_R \mathcal{P}_0(\mathsf{sk})$	$\xrightarrow{\mathsf{com}}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow_R \mathbb{F}_q$
$\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \alpha)$	$\xrightarrow{\mathsf{resp}_1}$	
	$\xleftarrow{\mathsf{ch}_2}$	$\mathsf{ch}_2 \leftarrow_R \{0, 1\}$
$\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \alpha, \mathsf{resp}_1, \mathsf{ch}_2)$	$\xrightarrow{\mathsf{resp}_2}$	
		$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \alpha, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$

- Unruh's transform: resp_2 for both $\mathsf{ch}_2 \in \{0, 1\}$, per α

$\mathcal{MQ}$ problem

The function family $\mathcal{MQ}(n, m, \mathbb{F}_q)$:

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})), \text{ where } f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$$
$$\text{for } a_{i,j}^{(s)},\ b_i^{(s)} \in \mathbb{F}_q, s \in \{1, \ldots, m\}$$

$\mathcal{MQ}$ problem

The function family $\mathcal{MQ}(n, m, \mathbb{F}_q)$:

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})), \text{ where } f_s(\mathbf{x}) = \textstyle\sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$$
$$\text{for } a_{i,j}^{(s)},\ b_i^{(s)} \in \mathbb{F}_q, s \in \{1, \ldots, m\}$$

Problem: For given $\mathbf{y} \in \mathbb{F}_q^m$, find $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{x}) = \mathbf{y}$.

$\mathcal{MQ}$ problem

The function family $\mathcal{MQ}(n, m, \mathbb{F}_q)$:

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})), \text{ where } f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$$
$$\text{for } a_{i,j}^{(s)},\ b_i^{(s)} \in \mathbb{F}_q, s \in \{1, \ldots, m\}$$

Problem: For given $\mathbf{y} \in \mathbb{F}_q^m$, find $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{x}) = \mathbf{y}$.

i.e., solve the system of equations:

$$\begin{aligned} y_1 &= a_{1,1}^{(1)} x_1 x_1 + a_{1,2}^{(1)} x_1 x_2 + \ldots + a_{n,n}^{(1)} x_n x_n + b_1^{(1)} x_1 + \ldots + b_n^{(1)} x_n \\ &\vdots \\ y_m &= a_{1,1}^{(m)} x_1 x_1 + a_{1,2}^{(m)} x_1 x_2 + \ldots + a_{n,n}^{(m)} x_n x_n + b_1^{(m)} x_1 + \ldots + b_n^{(m)} x_n \end{aligned}$$

$\mathcal{MQ}$ problem: numerical example

- Example parameters: $n = m = 3$, $\mathbb{F}_q = \mathbb{F}_5$

$\mathcal{MQ}$ problem: numerical example

- Example parameters: $n = m = 3$, $\mathbb{F}_q = \mathbb{F}_5$
- Random system of functions $\mathbf{F}$:

$$y_1 = 4x_1x_1 + 3x_1x_2 + 0x_1x_3 + x_2x_2 + 2x_2x_3 + x_3x_3 + 0x_1 + 2x_2 + 2x_3$$
$$y_2 = x_1x_1 + 2x_1x_2 + x_1x_3 + 0x_2x_2 + 3x_2x_3 + 4x_3x_3 + 0x_1 + 3x_2 + 2x_3$$
$$y_3 = 0x_1x_1 + x_1x_2 + 4x_1x_3 + 3x_2x_2 + 0x_2x_3 + x_3x_3 + 4x_1 + x_2 + 0x_3$$

$\mathcal{MQ}$ problem: numerical example

- Example parameters: $n = m = 3$, $\mathbb{F}_q = \mathbb{F}_5$
- Random system of functions $\mathbf{F}$:

$$y_1 = 4x_1x_1 + 3x_1x_2 + 0x_1x_3 + x_2x_2 + 2x_2x_3 + x_3x_3 + 0x_1 + 2x_2 + 2x_3$$
$$y_2 = x_1x_1 + 2x_1x_2 + x_1x_3 + 0x_2x_2 + 3x_2x_3 + 4x_3x_3 + 0x_1 + 3x_2 + 2x_3$$
$$y_3 = 0x_1x_1 + x_1x_2 + 4x_1x_3 + 3x_2x_2 + 0x_2x_3 + x_3x_3 + 4x_1 + x_2 + 0x_3$$

- 'Secret' input $\mathbf{x} = (1, 4, 3)$

$\mathcal{MQ}$ problem: numerical example

- Example parameters: $n = m = 3$, $\mathbb{F}_q = \mathbb{F}_5$
- Random system of functions $\mathbf{F}$:

$$y_1 = 4x_1x_1 + 3x_1x_2 + 0x_1x_3 + x_2x_2 + 2x_2x_3 + x_3x_3 + 0x_1 + 2x_2 + 2x_3$$
$$y_2 = x_1x_1 + 2x_1x_2 + x_1x_3 + 0x_2x_2 + 3x_2x_3 + 4x_3x_3 + 0x_1 + 3x_2 + 2x_3$$
$$y_3 = 0x_1x_1 + x_1x_2 + 4x_1x_3 + 3x_2x_2 + 0x_2x_3 + x_3x_3 + 4x_1 + x_2 + 0x_3$$

- 'Secret' input $\mathbf{x} = (1, 4, 3)$

$$y_1 = 4 \cdot 1 \cdot 1 + 3 \cdot 1 \cdot 4 + 4 \cdot 4 + 2 \cdot 4 \cdot 3 + 3 \cdot 3 + 2 \cdot 4 + 2 \cdot 3$$
$$y_2 = 1 \cdot 1 + 2 \cdot 1 \cdot 4 + 1 \cdot 3 + 3 \cdot 4 \cdot 3 + 4 \cdot 3 \cdot 3 + 3 \cdot 4 + 2 \cdot 3$$
$$y_3 = 1 \cdot 4 + 4 \cdot 1 \cdot 3 + 3 \cdot 4 \cdot 4 + 3 \cdot 3 + 4 \cdot 1 + 4$$

$\mathcal{MQ}$ problem: numerical example

- Example parameters: $n = m = 3$, $\mathbb{F}_q = \mathbb{F}_5$
- Random system of functions $\mathbf{F}$:

$$y_1 = 4x_1x_1 + 3x_1x_2 + 0x_1x_3 + x_2x_2 + 2x_2x_3 + x_3x_3 + 0x_1 + 2x_2 + 2x_3$$
$$y_2 = x_1x_1 + 2x_1x_2 + x_1x_3 + 0x_2x_2 + 3x_2x_3 + 4x_3x_3 + 0x_1 + 3x_2 + 2x_3$$
$$y_3 = 0x_1x_1 + x_1x_2 + 4x_1x_3 + 3x_2x_2 + 0x_2x_3 + x_3x_3 + 4x_1 + x_2 + 0x_3$$

- 'Secret' input $\mathbf{x} = (1, 4, 3)$

$$y_1 = 4 \cdot 1 \cdot 1 + 3 \cdot 1 \cdot 4 + 4 \cdot 4 + 2 \cdot 4 \cdot 3 + 3 \cdot 3 + 2 \cdot 4 + 2 \cdot 3 = 79 \equiv 4$$
$$y_2 = 1 \cdot 1 + 2 \cdot 1 \cdot 4 + 1 \cdot 3 + 3 \cdot 4 \cdot 3 + 4 \cdot 3 \cdot 3 + 3 \cdot 4 + 2 \cdot 3 = 102 \equiv 2$$
$$y_3 = 1 \cdot 4 + 4 \cdot 1 \cdot 3 + 3 \cdot 4 \cdot 4 + 3 \cdot 3 + 4 \cdot 1 + 4 = 81 \equiv 1$$

- 'Public' output $\mathbf{y} = (4, 2, 1)$

Sakumoto-Shirai-Hiwatari 5-pass IDS [SSH11]

$\mathcal{P} : (\mathbf{F}, \mathbf{v}, \mathbf{s})$		$\mathcal{V} : (\mathbf{F}, \mathbf{v})$
$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$		
$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$		
$c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$		
$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$	$\xrightarrow{(c_0, c_1)}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow_R \mathbb{F}_q$
$\mathbf{t}_1 \leftarrow \alpha \mathbf{r}_0 - \mathbf{t}_0$		
$\mathbf{e}_1 \leftarrow \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$	$\xrightarrow{\text{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$	
	$\xleftarrow{\text{ch}_2}$	$\text{ch}_2 \leftarrow_R \{0, 1\}$
If $\text{ch}_2 = 0$, $\text{resp}_2 \leftarrow \mathbf{r}_0$		
Else $\text{resp}_2 \leftarrow \mathbf{r}_1$	$\xrightarrow{\text{resp}_2}$	
		If $\text{ch}_2 = 0$, Parse $\text{resp}_2 = \mathbf{r}_0$, check
		$c_0 \stackrel{?}{=} Com(\mathbf{r}_0, \alpha \mathbf{r}_0 - \mathbf{t}_1, \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1)$
		Else Parse $\text{resp}_2 = \mathbf{r}_1$, check
		$c_1 \stackrel{?}{=} Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$

Sakumoto-Shirai-Hiwatari 5-pass IDS [SSH11]

$\mathcal{P} : (\mathbf{F}, \mathbf{v}, \mathbf{s})$ — $\mathcal{V} : (\mathbf{F}, \mathbf{v})$

Prover:

$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$

$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$

$c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$

$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$

$\xrightarrow{(c_0, c_1)}$

Verifier: $\alpha \leftarrow_R \mathbb{F}_q$

$\xleftarrow{\alpha}$

Prover:

$\mathbf{t}_1 \leftarrow \alpha \mathbf{r}_0 - \mathbf{t}_0$

$\mathbf{e}_1 \leftarrow \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$

$\xrightarrow{\mathsf{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$

Verifier: $\mathsf{ch}_2 \leftarrow_R \{0, 1\}$

$\xleftarrow{\mathsf{ch}_2}$

Prover:

If $\mathsf{ch}_2 = 0$, $\mathsf{resp}_2 \leftarrow \mathbf{r}_0$

Else $\mathsf{resp}_2 \leftarrow \mathbf{r}_1$

$\xrightarrow{\mathsf{resp}_2}$

Verifier:

If $\mathsf{ch}_2 = 0$, Parse $\mathsf{resp}_2 = \mathbf{r}_0$, check

$c_0 \stackrel{?}{=} Com(\mathbf{r}_0, \alpha \mathbf{r}_0 - \mathbf{t}_1, \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1)$

Else Parse $\mathsf{resp}_2 = \mathbf{r}_1$, check

$c_1 \stackrel{?}{=} Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$

Sakumoto-Shirai-Hiwatari IDS [SSH11]

- Key technique: cut-and-choose for $\mathcal{MQ}$
  - Analogously, consider DLP: $s = r_0 + r_1 \Rightarrow g^s = g^{r_0} \cdot g^{r_1}$

Sakumoto-Shirai-Hiwatari IDS [SSH11]

- Key technique: cut-and-choose for $\mathcal{MQ}$
  - Analogously, consider DLP: $s = r_0 + r_1 \Rightarrow g^s = g^{r_0} \cdot g^{r_1}$

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$

Sakumoto-Shirai-Hiwatari IDS [SSH11]

- Key technique: cut-and-choose for $\mathcal{MQ}$
  - Analogously, consider DLP: $s = r_0 + r_1 \Rightarrow g^s = g^{r_0} \cdot g^{r_1}$

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$

  - For $g_s \in \mathbf{G}$: $g_s(\mathbf{x}, \mathbf{y}) = \sum_{i,j} a_{i,j}^{(s)}(x_i y_j + x_j y_i)$
    - Recall: $f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$

Sakumoto-Shirai-Hiwatari IDS [SSH11]

- Key technique: cut-and-choose for $\mathcal{MQ}$
  - Analogously, consider DLP: $s = r_0 + r_1 \Rightarrow g^s = g^{r_0} \cdot g^{r_1}$

- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$

  - For $g_s \in \mathbf{G}$: $g_s(\mathbf{x}, \mathbf{y}) = \sum_{i,j} a_{i,j}^{(s)}(x_i y_j + x_j y_i)$
    - Recall: $f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$

  - See [SSH11] for details
  - Takeaway: evaluating $\mathbf{G} \approx$ evaluating $\mathbf{F}$

Sakumoto-Shirai-Hiwatari IDS [SSH11]

- Key technique: cut-and-choose for $\mathcal{MQ}$
  - Analogously, consider DLP: $s = r_0 + r_1 \Rightarrow g^s = g^{r_0} \cdot g^{r_1}$
- Bilinear map $\mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{F}(\mathbf{x} + \mathbf{y}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y})$
  - Split $\mathbf{s}$ and $\mathbf{F}(\mathbf{s})$ into $\mathbf{r}_0, \mathbf{r}_1$ and $\mathbf{F}(\mathbf{r}_0), \mathbf{F}(\mathbf{r}_1)$
    - Since then $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1 \Rightarrow \mathbf{F}(\mathbf{s}) = \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1)$
  - Split $\mathbf{r}_0$ and $\mathbf{F}(\mathbf{r}_0)$ further into $\mathbf{t}_0, \mathbf{t}_1$ resp. $\mathbf{e}_0, \mathbf{e}_1$
  - For $g_s \in \mathbf{G}$: $g_s(\mathbf{x}, \mathbf{y}) = \sum_{i,j} a_{i,j}^{(s)}(x_i y_j + x_j y_i)$
    - Recall: $f_s(\mathbf{x}) = \sum_{i,j} a_{i,j}^{(s)} x_i x_j + \sum_i b_i^{(s)} x_i$
  - See [SSH11] for details
  - Takeaway: evaluating $\mathbf{G} \approx$ evaluating $\mathbf{F}$
- Result: reveal either $\mathbf{r}_0$ or $\mathbf{r}_1$, and $(\mathbf{t}_0, \mathbf{e}_0)$ or $(\mathbf{t}_1, \mathbf{e}_1)$

SOFIA

Key generation:

- Sample seeds, expand $\mathbf{F}$, evaluate $\mathbf{v} = \mathbf{F}(\mathbf{s})$
  - Identical to MQDSS

SOFIA

Key generation:

- Sample seeds, expand $\mathbf{F}$, evaluate $\mathbf{v} = \mathbf{F}(\mathbf{s})$
  - Identical to MQDSS

Signing:

- Run the transformed IDS r times in parallel
  - Commit to randomness; $r \times \mathbf{G}$
  - Respond to t challenges $\alpha \in \mathbb{F}_q$; $r \times t \times \mathbf{F}$
- Hash all (blinded) responses to set of indices
- Unblind indicated responses

SOFIA

Key generation:

- Sample seeds, expand $\mathbf{F}$, evaluate $\mathbf{v} = \mathbf{F}(\mathbf{s})$
  - Identical to MQDSS

Signing:

- Run the transformed IDS r times in parallel
  - Commit to randomness; $r \times \mathbf{G}$
  - Respond to t challenges $\alpha \in \mathbb{F}_q$; $r \times t \times \mathbf{F}$
- Hash all (blinded) responses to set of indices
- Unblind indicated responses

Verification:

- Reconstruct indices
- Verify revealed responses
- Verify that commitments match responses; $r \times \mathbf{F}$, $\backsim \frac{1}{2} r \times \mathbf{G}$

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses
- Fix challenge space to $|\mathsf{ChS}_1| = t$

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses
- Fix challenge space to $|\mathsf{ChS}_1| = t$
- Unlink α and ch_2

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses
- Fix challenge space to $|\mathsf{ChS}_1| = t$
- Unlink α and ch_2
- Omit commitments [SSH11]

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses
- Fix challenge space to $|\mathsf{ChS}_1| = t$
- Unlink α and ch_2
- Omit commitments [SSH11]
- Self-randomizing commitments

Optimizations

Many similarities to e.g. Picnic [CDG+17]

- Exclude redundant blinded responses
- Fix challenge space to $|\mathsf{ChS}_1| = t$
- Unlink α and ch_2
- Omit commitments [SSH11]
- Self-randomizing commitments

What doesn't help:

- Opening for multiple α
- Committing to multiple $\mathbf{t}_0$

Parameter choice

- 128 bits post-quantum security
- Focus on signature size

Parameter choice

- 128 bits post-quantum security
- Focus on signature size
- Candidates: $\mathcal{MQ}(128, \mathbb{F}_4)$, $\mathcal{MQ}(96, \mathbb{F}_7)$, $\mathcal{MQ}(72, \mathbb{F}_{16})$
  - . . . and even $\mathbb{F}_5$, $\mathbb{F}_8$

Parameter choice

- 128 bits post-quantum security
- Focus on signature size
- Candidates: $\underline{\mathcal{MQ}(128, \mathbb{F}_4)}$, $\mathcal{MQ}(96, \mathbb{F}_7)$, $\mathcal{MQ}(72, \mathbb{F}_{16})$
  - ... and even $\mathbb{F}_5$, $\mathbb{F}_8$

Parameter choice

- 128 bits post-quantum security
- Focus on signature size
- Candidates: $\underline{\mathcal{MQ}(128, \mathbb{F}_4)}$, $\mathcal{MQ}(96, \mathbb{F}_7)$, $\mathcal{MQ}(72, \mathbb{F}_{16})$
  - ... and even $\mathbb{F}_5$, $\mathbb{F}_8$
- Analyzed using Hybrid approach and BooleanSolve
  - Instantiated with Grover search
  - At least 2^{117} operations

Parameter choice

- 128 bits post-quantum security
- Focus on signature size
- Candidates: $\underline{\mathcal{MQ}(128, \mathbb{F}_4)}$, $\mathcal{MQ}(96, \mathbb{F}_7)$, $\mathcal{MQ}(72, \mathbb{F}_{16})$
  - ... and even $\mathbb{F}_5$, $\mathbb{F}_8$
- Analyzed using Hybrid approach and BooleanSolve
  - Instantiated with Grover search
  - At least 2^{117} operations
- $t = 3$, $r = 438$ (since $2^{-(r \log \frac{2t}{t+1})/2} < 2^{-128}$)
- XOFs, hashes, PRGs: SHAKE, cSHAKE, (AES)

Implementation

- Evaluating $\mathcal{MQ}$
- XOFs

Implementation

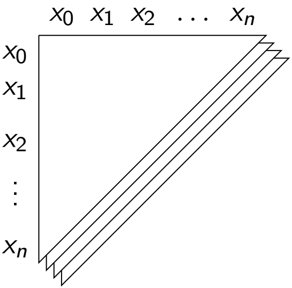
- Evaluating $\mathcal{MQ}$
  - 438 rounds, 2x per round
  - Pairwise multiply 128x $\in \mathbb{F}_4$
  - Multiply by coefficients from **F**, $\in \mathbb{F}_4$
  - Accumulate
- XOFs
  - Blinding commitments
  - Expanding **F**: 262 KiB
  - External parallelism and cSHAKE

Evaluating $\mathcal{MQ}$

- From $\mathbf{F}(\mathbf{x})$ to $\mathbf{x}$ is hard
- From $\mathbf{x}$ to $\mathbf{F}(\mathbf{x})$ should be easy

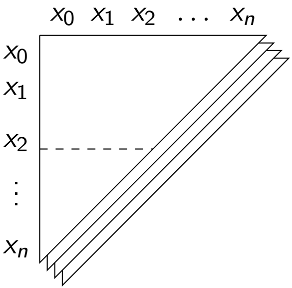
Evaluating $\mathcal{MQ}$

- From $\mathbf{F}(\mathbf{x})$ to $\mathbf{x}$ is hard
- From $\mathbf{x}$ to $\mathbf{F}(\mathbf{x})$ should be fast



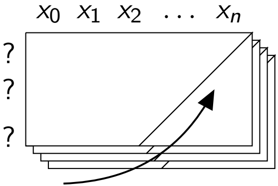
Evaluating $\mathcal{MQ}$

- From $\mathbf{F}(\mathbf{x})$ to $\mathbf{x}$ is hard
- From $\mathbf{x}$ to $\mathbf{F}(\mathbf{x})$ should be fast



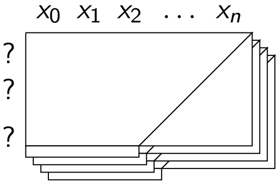
Evaluating $\mathcal{MQ}$

- From $\mathbf{F}(\mathbf{x})$ to $\mathbf{x}$ is hard
- From $\mathbf{x}$ to $\mathbf{F}(\mathbf{x})$ should be fast



Evaluating $\mathcal{MQ}$

- From $\mathbf{F}(\mathbf{x})$ to $\mathbf{x}$ is hard
- From $\mathbf{x}$ to $\mathbf{F}(\mathbf{x})$ should be fast



Evaluating $\mathcal{MQ}$

- 128x $\mathbb{F}_4$
- Bitsliced: two lanes in AVX2 register
- Each lane: 16 bytes, `vpshufb`
- Quadratic terms: ‘scheduling scripts’ similar to MQDSS

Evaluating $\mathcal{MQ}$

- 128x $\mathbb{F}_4$
- Bitsliced: two lanes in AVX2 register
- Each lane: 16 bytes, `vpshufb`
- Quadratic terms: 'scheduling scripts' similar to MQDSS
- Pre-set two register: $[\mathbf{x}_{high} \oplus \mathbf{x}_{low} | \mathbf{x}_{low}]$ and $[\mathbf{x}_{high} | \mathbf{x}_{high}]$

Evaluating $\mathcal{MQ}$

- 128x $\mathbb{F}_4$
- Bitsliced: two lanes in AVX2 register
- Each lane: 16 bytes, `vpshufb`
- Quadratic terms: ‘scheduling scripts’ similar to MQDSS
- Pre-set two register: $[\mathbf{x}_{high} \oplus \mathbf{x}_{low} | \mathbf{x}_{low}]$ and $[\mathbf{x}_{high} | \mathbf{x}_{high}]$

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

- `vpand`, `vpand`, `vpermq`, `vpxor`

Evaluating $\mathcal{MQ}$

- 'Vertically:' broadcast monomial, multiply with **F**
  - $a_{1,1}^{(1)}x_1x_1, a_{1,1}^{(\mathbf{2})}x_1x_1, a_{1,1}^{(\mathbf{3})}x_1x_1, a_{1,1}^{(4)}x_1x_1, \ldots$
- 'Horizontally:' iterate over output elements, `popcnt`
  - $a_{1,1}^{(1)}x_1x_1, a_{1,\mathbf{2}}^{(1)}x_1x_{\mathbf{2}}, a_{1,\mathbf{3}}^{(1)}x_1x_{\mathbf{3}}, \ldots a_{2,1}^{(1)}x_2x_1, a_{2,2}^{(1)}x_2x_2, \ldots$

Evaluating $\mathcal{MQ}$

- 'Vertically:' broadcast monomial, multiply with **F**
  - $a_{1,1}^{(1)}x_1x_1, a_{1,1}^{(\mathbf{2})}x_1x_1, a_{1,1}^{(\mathbf{3})}x_1x_1, a_{1,1}^{(4)}x_1x_1, \ldots$
- 'Horizontally:' iterate over output elements, `popcnt`
  - $a_{1,1}^{(1)}x_1x_1, a_{1,\mathbf{2}}^{(1)}x_1x_{\mathbf{2}}, a_{1,\mathbf{3}}^{(1)}x_1x_{\mathbf{3}}, \ldots a_{2,1}^{(1)}x_2x_1, a_{2,2}^{(1)}x_2x_2, \ldots$

- Horizontal: more loads, but internal parallelism

Evaluating $\mathcal{MQ}$

- 'Vertically:' broadcast monomial, multiply with $\mathbf{F}$
  - $a_{1,1}^{(1)}x_1x_1, a_{1,1}^{(\mathbf{2})}x_1x_1, a_{1,1}^{(\mathbf{3})}x_1x_1, a_{1,1}^{(4)}x_1x_1, \ldots$
- 'Horizontally:' iterate over output elements, `popcnt`
  - $a_{1,1}^{(1)}x_1x_1, a_{1,\mathbf{2}}^{(1)}x_1x_{\mathbf{2}}, a_{1,\mathbf{3}}^{(1)}x_1x_{\mathbf{3}}, \ldots a_{2,1}^{(1)}x_2x_1, a_{2,2}^{(1)}x_2x_2, \ldots$

- Horizontal: more loads, but internal parallelism

- Both cases: delay reductions in $\mathbb{F}_4$
  - $[\hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{low}]$ and $[\hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{low}]$

Evaluating $\mathcal{MQ}$

- 'Vertically:' broadcast monomial, multiply with **F**
  - $a_{1,1}^{(1)}x_1x_1, a_{1,1}^{(\mathbf{2})}x_1x_1, a_{1,1}^{(\mathbf{3})}x_1x_1, a_{1,1}^{(4)}x_1x_1, \ldots$
- 'Horizontally:' iterate over output elements, `popcnt`
  - $a_{1,1}^{(1)}x_1x_1, a_{1,\mathbf{2}}^{(1)}x_1x_\mathbf{2}, a_{1,\mathbf{3}}^{(1)}x_1x_\mathbf{3}, \ldots a_{2,1}^{(1)}x_2x_1, a_{2,2}^{(1)}x_2x_2, \ldots$

- Horizontal: more loads, but internal parallelism

- Both cases: delay reductions in $\mathbb{F}_4$
  - $[\hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{low}]$ and $[\hat{\mathbf{x}}_{low} \wedge \mathbf{f}_{high} | \hat{\mathbf{x}}_{high} \wedge \mathbf{f}_{low}]$

- Both cases: external parallelism over constant **F**

- Horizontal in batches of 3, avg. 17 558 cycles per $\mathcal{MQ}$

SOFIA-4-128 vs MQDSS-31-64

a.k.a. the price of QROM

- Signature size: 123 KiB (MQDSS: 40 KiB)
- 64 bytes pk, 32 bytes sk (MQDSS: 72 B, 64 B)

SOFIA-4-128 vs MQDSS-31-64

a.k.a. the price of QROM

- Signature size: 123 KiB (MQDSS: 40 KiB)
- 64 bytes pk, 32 bytes sk (MQDSS: 72 B, 64 B)

- Key generation 1.16 M cycles (MQDSS: 1.18 M)
- Signing 21.31 M cycles (MQDSS: 8.51 M)
  - ∽75% $\mathcal{MQ}$
  - ∽25% SHAKE
- Verification 15.49 M cycles (MQDSS: 5.75 M)

(Intel Haswell, Core-i7-4770K, AVX2)

Conclusions and comparisons

- Conservative $\mathcal{MQ}$ in the QROM
- Small keys, large signatures, not too slow

Conclusions and comparisons

- Conservative $\mathcal{MQ}$ in the QROM
- Small keys, large signatures, not too slow
- Significantly bigger than SPHINCS-256
  - And thus SPHINCS$^+$
- Smaller & faster than Picnic-10-38
  - $\backsim$ as big as Picnic-L5-FS

Conclusions and comparisons

- Conservative $\mathcal{MQ}$ in the QROM
- Small keys, large signatures, not too slow

- Significantly bigger than SPHINCS-256
  - And thus SPHINCS$^+$
- Smaller & faster than Picnic-10-38
  - $\backsim$ as big as Picnic-L5-FS
- Much bigger/slower than lattices, e.g. Dilithium, qTESLA
  - .. but much faster (& smaller keys) than TESLA-1,-2

Conclusions and comparisons

- Conservative $\mathcal{MQ}$ in the QROM
- Small keys, large signatures, not too slow

- Significantly bigger than SPHINCS-256
  - And thus SPHINCS$^+$
- Smaller & faster than Picnic-10-38
  - $\backsim$ as big as Picnic-L5-FS
- Much bigger/slower than lattices, e.g. Dilithium, qTESLA
  - .. but much faster (& smaller keys) than TESLA-1,-2

- C and AVX2 code available (public domain): `https://joostrijneveld.nl/papers/sofia`

References I

Ward Beullens, Bart Preneel, Alan Szepieniec, and Frederik Vercauteren.
LUOV.
Submission to NIST's post-quantum crypto standardization project, 2017.

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

A. Casanova, Jean-Charles Faugère, Gilles Macario-Rat, Jacques Patarin, Ludovic Perret, and J. Ryckeghem.
GeMSS.
Submission to NIST's post-quantum crypto standardization project, 2017.

References II

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

Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe.
MQDSS.
Submission to NIST's post-quantum crypto standardization project, 2017.

Jintai Ding, Ming-Shen Chen, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang.
Gui.
Submission to NIST's post-quantum crypto standardization project, 2017.

References III

Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, and Bo-Yin Yang.

Rainbow.

Submission to NIST's post-quantum crypto standardization project, 2017.

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.

References IV

Marc Fischlin.

Communication-efficient non-interactive proofs of knowledge with online extractors.

In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, 2005.

https://www.iacr.org/archive/crypto2005/36210148/36210148.pdf.

Jean-Charles Faugère, Ludovic Perret, and J. Ryckeghem.

DualModeMS.

Submission to NIST's post-quantum crypto standardization project, 2017.

Eike Kiltz, Julian Loss, and Jiaxin Pan.

Tightly-secure signatures from five-move identification protocols.

In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, pages 68–94, Cham, 2017. Springer International Publishing.

References V

Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner.
A concrete treatment of fiat-shamir signatures in the quantum random-oracle model.
Cryptology ePrint Archive, Report 2017/916, 2017.
https://eprint.iacr.org/2017/916.

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.

Kyung-Ah Shim, Cheol-Min Park, and Aeyoung Kim.
HiMQ-3.
Submission to NIST's post-quantum crypto standardization project, 2017.

References VI

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.

Dominique Unruh.

Non-interactive zero-knowledge proofs in the quantum random oracle model.

In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 755–784. Springer, 2015.

http://eprint.iacr.org/2014/587.