MQDSS signatures: construction

Ming-Shing Chen[1,2], Andreas Hülsing[3], **Joost Rijneveld**[4],
Simona Samardjiska[5], Peter Schwabe[4]

National Taiwan University[1] / Academia Sinica[2], Taipei, Taiwan
Eindhoven University of Technology, The Netherlands[3]
Radboud University, Nijmegen, The Netherlands[4]
"Ss. Cyril and Methodius" University, Skopje, Republic of Macedonia[5]

2017-03-24

Crypto Working Group

Post-quantum signatures

Problem: we want a post-quantum signature scheme

- Security arguments
- 'Acceptable' speed and size

Post-quantum signatures

Problem: we want a post-quantum signature scheme

- Security arguments
- 'Acceptable' speed and size

Solutions:

- Hash-based: SPHINCS [BHH+15], XMSS [BDH11, HRS16]
  - Slow or stateful
- Lattice-based: (Ring-)TESLA [ABB+16, ABB+15], BLISS [DDL+13], GLP [GLP12]
  - Large keys, or additional structure
- $\mathcal{MQ}$: ?
  - Unclear security: many broken (except HFEv-, UOV)

This work

- Transform class of 5-pass IDS to signature schemes
  - Extend Fiat-Shamir transform
- Prove an earlier attempt [EDV+12] vacuous
  - Amended in [DGV+16]
- Propose MQDSS
  - Obtained by performing transform
  - Hardness of $\mathcal{MQ}$
- Instantiate and implement as MQDSS-31-64

But also:

- Reduction in the ROM (not in QROM)
- No tight proof

This talk

- Transform class of 5-pass IDS to signature schemes
  - Extend Fiat-Shamir transform
- Prove an earlier attempt [EDV+12] vacuous
  - Amended in [DGV+16]
- Propose MQDSS
  - Obtained by performing transform
  - Hardness of $\mathcal{MQ}$
- Instantiate and implement as MQDSS-31-64

But also:

- Reduction in the ROM (not in QROM)
- No tight proof

Canonical Identification Schemes

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

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

Security of the IDS

- Passively secure IDS

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

- Shows knowledge of secret
- 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 that transcripts do not leak the secret

Fiat-Shamir transform

- First transform IDS with soundness error κ to negl(k)
  - Using parallel composition

Cheating prover

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

Cheating prover

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

$$
\begin{array}{lcl}
\mathcal{P}^* & & \mathcal{V} \\
\hline
\mathsf{ch}^* \leftarrow_R \mathsf{ChS}(1^k) & & \\
\mathsf{com}, \mathsf{resp} \leftarrow_R \mathcal{P}_0^*(\mathsf{sk}, \mathsf{ch}^*) & \xrightarrow{\quad \mathsf{com} \quad} & \\
 & & \mathsf{ch} \leftarrow_R \mathsf{ChS}(1^k) \\
 & \xleftarrow{\quad \mathsf{ch} \quad} & \\
\text{If } \mathsf{ch} \neq \mathsf{ch}^*, \textit{fail} & & \\
 & \xrightarrow{\quad \mathsf{resp} \quad} & \\
 & & b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})
\end{array}
$$

Parallel Canonical IDS

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
\mathsf{com}^{(1)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\mathsf{com}^{(2)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\vdots & & \\
\mathsf{com}^{(r)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & \xrightarrow{\forall_i\, \mathsf{com}^{(i)}} & \\
& & \mathsf{ch}^{(1)} \leftarrow_R \mathsf{ChS}(1^k) \\
& & \mathsf{ch}^{(2)} \leftarrow_R \mathsf{ChS}(1^k) \\
& & \vdots \\
& \xleftarrow{\forall_i\, \mathsf{ch}^{(i)}} & \mathsf{ch}^{(r)} \leftarrow_R \mathsf{ChS}(1^k) \\
\mathsf{resp}^{(1)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(1)}, \mathsf{ch}^{(1)}) & & \\
\mathsf{resp}^{(2)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(2)}, \mathsf{ch}^{(2)}) & & \\
\vdots & & \\
\mathsf{resp}^{(r)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(r)}, \mathsf{ch}^{(r)}) & \xrightarrow{\forall_i\, \mathsf{resp}^{(i)}} & \\
& & b \leftarrow \forall_i\, \mathsf{Vf}(\mathsf{pk}, \mathsf{com}^{(i)}, \mathsf{ch}^{(i)}, \mathsf{resp}^{(i)})
\end{array}
$$

Fiat-Shamir transform

- First transform IDS with soundness error κ to negl(k)
  - Using parallel composition
- Transform IDS into signature
- Non-interactive:
  - Signer is ‘prover’
  - Function $\mathcal{H}$ provides challenges
  - Transcript is signature

Parallel Canonical IDS

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
\mathsf{com}^{(1)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\mathsf{com}^{(2)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\vdots & & \\
\mathsf{com}^{(r)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & \xrightarrow{\forall_i\, \mathsf{com}^{(i)}} & \\
& & \mathsf{ch}^{(1)} \leftarrow_R \mathsf{ChS}(1^k) \\
& & \mathsf{ch}^{(2)} \leftarrow_R \mathsf{ChS}(1^k) \\
& & \vdots \\
& \xleftarrow{\forall_i\, \mathsf{ch}^{(i)}} & \mathsf{ch}^{(r)} \leftarrow_R \mathsf{ChS}(1^k) \\
\mathsf{resp}^{(1)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(1)}, \mathsf{ch}^{(1)}) & & \\
\mathsf{resp}^{(2)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(2)}, \mathsf{ch}^{(2)}) & & \\
\vdots & & \\
\mathsf{resp}^{(r)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(r)}, \mathsf{ch}^{(r)}) & \xrightarrow{\forall_i\, \mathsf{resp}^{(i)}} & \\
& & b \leftarrow \forall_i\, \mathsf{Vf}(\mathsf{pk}, \mathsf{com}^{(i)}, \mathsf{ch}^{(i)}, \mathsf{resp}^{(i)})
\end{array}
$$

Transformed IDS

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
\mathsf{com}^{(1)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\qquad \vdots & & \\
\mathsf{com}^{(r)} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\sigma_0 \leftarrow \mathsf{com}^{(1)}, \mathsf{com}^{(2)}, \ldots \mathsf{com}^{(r)} & & \\
\mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \ldots, \mathsf{ch}^{(r)} \leftarrow \mathcal{H}(\sigma_0, M) & & \\
\mathsf{resp}^{(1)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(1)}, \mathsf{ch}^{(1)}) & & \\
\qquad \vdots & & \\
\mathsf{resp}^{(r)} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}^{(r)}, \mathsf{ch}^{(r)}) & & \\
\sigma_1 \leftarrow \mathsf{resp}^{(1)}, \mathsf{resp}^{(2)}, \ldots \mathsf{resp}^{(r)} & \xrightarrow{m, \sigma_0, \sigma_1} & \\
& & \mathsf{com}^{(1)}, \mathsf{com}^{(2)}, \ldots, \mathsf{com}^{(r)} \leftarrow \sigma_0 \\
& & \mathsf{ch}^{(1)}, \mathsf{ch}^{(2)}, \ldots, \mathsf{ch}^{(r)} \leftarrow \mathcal{H}(\sigma_0, M) \\
& & \mathsf{resp}^{(1)}, \mathsf{resp}^{(2)}, \ldots, \mathsf{resp}^{(r)} \leftarrow \sigma_1 \\
& & b \leftarrow \forall_i \, \mathsf{Vf}(\mathsf{pk}, \mathsf{com}^{(i)}, \mathsf{ch}^{(i)}, \mathsf{resp}^{(i)})
\end{array}
$$

Fiat-Shamir transform

- First transform IDS with soundness error κ to negl(k)
  - Using parallel composition
- Transform IDS into signature
- Non-interactive:
  - Signer is ‘prover’
  - Function $\mathcal{H}$ provides challenges
  - Transcript is signature
- Generalize to 5-pass
  - Benefit from lower soundness error
  - Two challenges and responses

Fiat-Shamir transform

- First transform IDS with soundness error κ to negl(k)
  - Using parallel composition
- Transform IDS into signature
- Non-interactive:
  - Signer is 'prover'
  - Function $\mathcal{H}$ provides challenges
  - Transcript is signature
- Generalize to 5-pass
  - Benefit from lower soundness error
  - Two challenges and responses
  - See Simona's talk!

Canonical 5-pass IDS

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk}) & & \\
& \xrightarrow{\quad \mathsf{com} \quad} & \\
& & \mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k) \\
& \xleftarrow{\quad \mathsf{ch}_1 \quad} & \\
\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1) & & \\
& \xrightarrow{\quad \mathsf{resp}_1 \quad} & \\
& & \mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k) \\
& \xleftarrow{\quad \mathsf{ch}_2 \quad} & \\
\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2) & & \\
& \xrightarrow{\quad \mathsf{resp}_2 \quad} & \\
& & b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \\
& & \qquad\qquad \mathsf{ch}_2, \mathsf{resp}_2)
\end{array}
$$

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

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

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

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

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

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 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 $(\mathbf{t}_1, \mathbf{e}_1)$

Sakumoto-Shirai-Hiwatari 3-pass IDS [SSH11]

$\mathcal{P} : (\mathbf{F}, \mathbf{v}, \mathbf{s})$		$\mathcal{V} : (\mathbf{F}, \mathbf{v})$
$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$		
$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0, \quad \mathbf{t}_1 \leftarrow \mathbf{r}_0 - \mathbf{t}_0$		
$\mathbf{e}_1 \leftarrow \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$		
$c_0 \leftarrow Com(\mathbf{r}_1, G(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$		
$c_1 \leftarrow Com(\mathbf{t}_0, \mathbf{e}_0)$		
$c_2 \leftarrow Com(\mathbf{t}_1, \mathbf{e}_1)$	$\xrightarrow{(c_0, c_1, c_2)}$	
	$\xleftarrow{\text{ch}}$	$\text{ch} \leftarrow_R \{0, 1, 2\}$
If $\text{ch} = 0$, $\text{resp} \leftarrow (\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$		
If $\text{ch} = 1$, $\text{resp} \leftarrow (\mathbf{r}_1, \mathbf{t}_1, \mathbf{e}_1)$		
If $\text{ch} = 2$, $\text{resp} \leftarrow (\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$	$\xrightarrow{\text{resp}}$	
		If $\text{ch} = 0$, Parse $\text{resp} = (\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$, check
		$c_1 \stackrel{?}{=} Com(\mathbf{r}_0 - \mathbf{t}_1, F(\mathbf{r}_0) - \mathbf{e}_1)$
		$c_2 \stackrel{?}{=} Com(\mathbf{t}_1, \mathbf{e}_1)$
		If $\text{ch} = 1$, Parse $\text{resp} = (\mathbf{r}_1, \mathbf{t}_1, \mathbf{e}_1)$, check
		$c_0 \stackrel{?}{=} Com(\mathbf{r}_1, \mathbf{v} - F(\mathbf{r}_1) - G(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$
		$c_2 \stackrel{?}{=} Com(\mathbf{t}_1, \mathbf{e}_1)$
		If $\text{ch} = 2$, Parse $\text{resp} = (\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, check
		$c_0 \stackrel{?}{=} Com(\mathbf{r}_1, G(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$
		$c_1 \stackrel{?}{=} Com(\mathbf{t}_0, \mathbf{e}_0)$

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

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

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations

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

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

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
  - Tricks to reduce size
    - Only include necessary commits (hash others) [SSH11]
    - Commit to seeds

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

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

$\mathcal{P} : (\mathbf{F}, \mathbf{v}, \mathbf{s})$		$\mathcal{V} : (\mathbf{F}, \mathbf{v})$
$\mathbf{r}_0, \mathbf{t}_0 \leftarrow_R \mathbb{F}_q^n, \mathbf{e}_0 \leftarrow_R \mathbb{F}_q^m$		
$\mathbf{r}_1 \leftarrow \mathbf{s} - \mathbf{r}_0$		
$c_0 \leftarrow Com(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$		
$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0)$	$\xrightarrow{\mathcal{H}(c_0, c_1)}$	
	$\xleftarrow{\alpha}$	$\alpha \leftarrow_R \mathbb{F}_q$
$\mathbf{t}_1 \leftarrow \alpha \mathbf{r}_0 - \mathbf{t}_0$		
$\mathbf{e}_1 \leftarrow \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0$	$\xrightarrow{\mathrm{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$	
	$\xleftarrow{\mathrm{ch}_2}$	$\mathrm{ch}_2 \leftarrow_R \{0, 1\}$
If $\mathrm{ch}_2 = 0$, $\mathrm{resp}_2 \leftarrow \mathbf{r}_0$		
Else $\mathrm{resp}_2 \leftarrow \mathbf{r}_1$	$\xrightarrow{\mathrm{resp}_2, c_{(1-\mathrm{ch}_2)}}$	
		If $\mathrm{ch}_2 = 0$, Parse $\mathrm{resp}_2 = \mathbf{r}_0$,
		$c_0' \leftarrow Com(\mathbf{r}_0, \alpha \mathbf{r}_0 - \mathbf{t}_1, \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_1)$
		$c_1' \leftarrow c_1$
		Else Parse $\mathrm{resp}_2 = \mathbf{r}_1$,
		$c_0' \leftarrow c_0$
		$c_1' \leftarrow Com(\mathbf{r}_1, \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}_1)) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{e}_1)$
		check $\mathcal{H}(c_0, c_1) \stackrel{?}{=} \mathcal{H}(c_0', c_1')$

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
  - Tricks to reduce size
    - Only include necessary commits (hash others) [SSH11]
    - Commit to seeds
- Verifying
  - Reconstruct D, $\mathbf{F}$

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
  - Tricks to reduce size
    - Only include necessary commits (hash others) [SSH11]
    - Commit to seeds
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges from σ_0, σ_1
  - Verify responses in σ_2

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
  - Tricks to reduce size
    - Only include necessary commits (hash others) [SSH11]
    - Commit to seeds
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges from σ_0, σ_1
  - Verify responses in σ_2
  - Reconstruct missing commitments
  - Check combined commitments hash

MQDSS

- Generate keys
  - Sample seed $\mathcal{S}_F \in \{0,1\}^k$, $\mathbf{sk} \in \mathbb{F}_q^n$ $\Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - Expand $\mathcal{S}_F$ to $\mathbf{F}$, compute $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- Signing
  - Sign randomized digest D over M
  - Perform r parallel rounds of transformed IDS
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
  - Tricks to reduce size
    - Only include necessary commits (hash others) [SSH11]
    - Commit to seeds
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges from σ_0, σ_1
  - Verify responses in σ_2
  - Reconstruct missing commitments
  - Check combined commitments hash
- Parameters: k, n, m, $\mathbb{F}_q$, Com, hash functions, PRGs

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- Soundness error $\kappa = \frac{q+1}{2q}$
  - Determines number of rounds; larger $q \Rightarrow$ less rounds
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- Soundness error $\kappa = \frac{q+1}{2q}$
  - Determines number of rounds; larger $q \Rightarrow$ less rounds
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
  - Restricted by security
    - If $m > n$, F5 and Hybrid approach perform better
    - If $m < n$, simply fix $n - m$ variables
    - Classically, analysis from [BFP12]: $n \geq 51$
    - Post-quantum: Grover search

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- Soundness error $\kappa = \frac{q+1}{2q}$
  - Determines number of rounds; larger $q \Rightarrow$ less rounds
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
  - Restricted by security
    - If $m > n$, F5 and Hybrid approach perform better
    - If $m < n$, simply fix $n - m$ variables
    - Classically, analysis from [BFP12]: $n \geq 51$
    - Post-quantum: Grover search
  - Chosen for ease of implementation
    - Fast reduction
    - 5-bit elements; 16x 16-bit words in AVX2

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- Soundness error $\kappa = \frac{q+1}{2q}$
  - Determines number of rounds; larger $q \Rightarrow$ less rounds
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
  - Restricted by security
    - If $m > n$, F5 and Hybrid approach perform better
    - If $m < n$, simply fix $n - m$ variables
    - Classically, analysis from [BFP12]: $n \geq 51$
    - Post-quantum: Grover search
  - Chosen for ease of implementation
    - Fast reduction
    - 5-bit elements; 16x 16-bit words in AVX2
    - In hindsight: reduce size by going lower?

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- Soundness error $\kappa = \frac{q+1}{2q}$
  - Determines number of rounds; larger $q \Rightarrow$ less rounds
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
  - Restricted by security
    - If $m > n$, F5 and Hybrid approach perform better
    - If $m < n$, simply fix $n - m$ variables
    - Classically, analysis from [BFP12]: $n \geq 51$
    - Post-quantum: Grover search
  - Chosen for ease of implementation
    - Fast reduction
    - 5-bit elements; 16x 16-bit words in AVX2
    - In hindsight: reduce size by going lower?
- From $q = 31$, we get $r = 269$: $\kappa^{269} < (\frac{1}{2})^{256}$

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
- $r = 269$

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
- $r = 269$
- Commitments, hashes, PRGs: SHA3-256, SHAKE-128

MQDSS-31-64

- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
- $r = 269$
- Commitments, hashes, PRGs: SHA3-256, SHAKE-128
- Signature σ contains:
  - R, for random digest $\Rightarrow$ 32B
  - Hash $\mathcal{H}(commits)$ $\Rightarrow$ 32B
  - For every round: $\Rightarrow 269 \times$
    - Response vectors $\mathbf{r}$, $\mathbf{t}$, $\mathbf{e}$ $\Rightarrow 3 \times 40$B
    - 'Missing commit' $\Rightarrow$ 32B

MQDSS-31-64

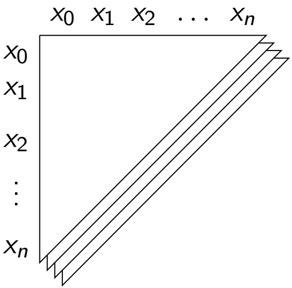
- Security parameter $k = 256$ ($\Rightarrow$ 128-bit PQ security)
- $\mathbb{F}_q = \mathbb{F}_{31}$, $n = m = 64$
- $r = 269$
- Commitments, hashes, PRGs: SHA3-256, SHAKE-128
- Signature σ contains:
  - R, for random digest $\Rightarrow$ 32B
  - Hash $\mathcal{H}(commits)$ $\Rightarrow$ 32B
  - For every round: $\Rightarrow 269 \times$
    - Response vectors $\mathbf{r}$, $\mathbf{t}$, $\mathbf{e}$ $\Rightarrow 3 \times$ 40B
    - 'Missing commit' $\Rightarrow$ 32B
- Adds up to 40 952 bytes

Evaluating $\mathcal{MQ}$

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

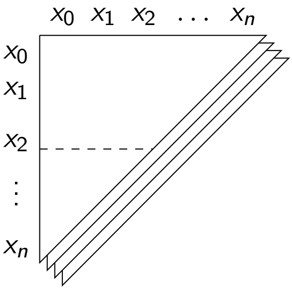
Evaluating $\mathcal{MQ}$

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



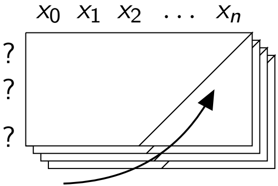
Evaluating $\mathcal{MQ}$

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



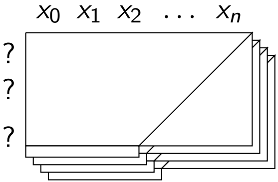
Evaluating $\mathcal{MQ}$

- 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 $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

$<<<$

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

<<<

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

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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	-	-

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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	-	-

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

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	-	-

Evaluating $\mathcal{MQ}$

- First compute monomials, then evaluate polynomials
- 64 elements in $\mathbb{F}_{31}$; 16 (or 32) per 256 bit AVX2 register
- Monomials: intuition of arrangement using 4×4:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	$\boxed{\text{E} \quad \text{F}}$	
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	-	-

Benchmarks & conclusion

- Signatures: $\sim$40 KB ($\approx$ SPHINCS)
- Public and private keys: 72 resp. 64 bytes
- Signing time: $\sim$8.5M cycles (2.43ms @ 3.5GHz)
  - Verification 5.2M, key generation 1.8M
- $\sim$6x faster than SPHINCS, $>$10x slower than lattices

Benchmarks & conclusion

- Signatures: $\sim$40 KB ($\approx$ SPHINCS)
- Public and private keys: 72 resp. 64 bytes
- Signing time: $\sim$8.5M cycles (2.43ms @ 3.5GHz)
  - Verification 5.2M, key generation 1.8M
- $\sim$6x faster than SPHINCS, $>$10x slower than lattices

- Competitive signatures with (non-tight) reduction to $\mathcal{MQ}$
  - Rewinding extractor for Fiat-Shamir, see e.g. [DOR+16]
- Relies only on $\mathcal{MQ}$ (and not IP)

Benchmarks & conclusion

- Signatures: ~40 KB ($\approx$ SPHINCS)
- Public and private keys: 72 resp. 64 bytes
- Signing time: ~8.5M cycles (2.43ms @ 3.5GHz)
  - Verification 5.2M, key generation 1.8M
- ~6x faster than SPHINCS, $>$10x slower than lattices
- Competitive signatures with (non-tight) reduction to $\mathcal{MQ}$
  - Rewinding extractor for Fiat-Shamir, see e.g. [DOR+16]
- Relies only on $\mathcal{MQ}$ (and not IP)
- Code is available (public domain):
  `https://joostrijneveld.nl/papers/mqdss/`

References I

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.

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.

Özgür Dagdelen, David Galindo, Pascal Véron, Sidi Mohamed El Yousfi Alaoui, and Pierre-Louis Cayrel.
Extended security arguments for signature schemes.
In Designs, Codes and Cryptography, 78(2), pages 441–461. Springer, 2016.

References II

Daniel J. Bernstein, Diana Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Peter Schwabe and Zooko Wilcox O'Hearn.
SPHINCS: Stateless, practical, hash-based, incredibly nice cryptographic signatures.
In Marc Fischlin and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 368-397. Springer, 2015.

Johannes Buchmann, Erik Dahmen and Andreas Hülsing.
XMSS – a practical forward secure signature scheme based on minimal security assumptions.
In Bo-Yin Yang, editor, *PQCrypto 2011*, volume 7071 of *LNCS*, pages 117-129. Springer, 2011.

Andreas Hülsing, Joost Rijneveld and Fang Song.
Mitigating multi-target attacks in hash-based signatures.
In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, volume 9614 of *LNCS*, pages 387-416. Springer, 2016.

References III

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, Tajjeeddine Rachidi, editors, *Progress in Cryptology – AFRICACRYPT 2016*, volume 9646 of *LNCS*, pages 44-60. Springer, 2016.

Erdem Alkim, Nina Bindel, Johannes Buchmann, Özgür Dagdelen and Peter Schwabe.
TESLA: Tightly-Secure Efficient Signatures from Standard Lattices.
In Cryptology ePrint Archive, Report 2015/755, 2015.

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.

References IV

Tim Güneysu, Vadim Lyubashevsky and Thomas Pöppelmann.
Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems.
In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *LNCS*, pages 530-547. Springer, 2012.

David Pointcheval and Jacques Stern.
Security proofs for signature schemes.
In Ueli Maurer, editor, *Advances in Cryptology – EUROCRYPT 1996*, volume 1070 of *LNCS*, pages 387-398. Springer, 1996.

David Derler, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger and Daniel Slamanig.
Digital Signatures from Symmetric-Key Primitives.
In Cryptology ePrint Archive, Report 2016/1085, 2016.

References V

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