MQDSS

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

2018-04-12
NIST PQC Standardization Conference

In a nutshell..

- $\mathcal{MQ}$-based 5-pass identification scheme
  - Fiat-Shamir transform
- Loose reduction from (only!) $\mathcal{MQ}$ problem
  - Security proof, instead of typical 'break and tweak'

In a nutshell..

- $\mathcal{MQ}$-based 5-pass identification scheme
  - Fiat-Shamir transform
- Loose reduction from (only!) $\mathcal{MQ}$ problem
  - Security proof, instead of typical ‘break and tweak’

- `MQDSS-31-48`: level 1, 32.1 KiB sigs.
- `MQDSS-31-64`: level 3, 66.2 KiB sigs.
- 62 resp. 88 byte public keys

In a nutshell..

- $\mathcal{MQ}$-based 5-pass identification scheme
  - Fiat-Shamir transform
- Loose reduction from (only!) $\mathcal{MQ}$ problem
  - Security proof, instead of typical 'break and tweak'

- `MQDSS-31-48`: level 1, 32.1 KiB sigs.
- `MQDSS-31-64`: level 3, 66.2 KiB sigs.
- 62 resp. 88 byte public keys

- Not blazingly fast, not prohibitively slow:
  0.3 - 0.7 ms keygen, 2 - 4 ms sign, 1 - 3 ms verify

  (3.5GHz Haswell, AVX2)

In a nutshell..

- $\mathcal{MQ}$-based 5-pass identification scheme
  - Fiat-Shamir transform
- Loose reduction from (only!) $\mathcal{MQ}$ problem
  - Security proof, instead of typical 'break and tweak'

- `MQDSS-31-48`: level 1, 32.1 KiB sigs.
- `MQDSS-31-64`: level 3, 66.2 KiB sigs.
- 62 resp. 88 byte public keys

- Not blazingly fast, not prohibitively slow:
  0.3 - 0.7 ms keygen, 2 - 4 ms sign, 1 - 3 ms verify

  (3.5GHz Haswell, AVX2)

- Only small tweaks since ASIACRYPT 2016 [CHR+16]

Canonical Identification Schemes

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

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

Fiat-Shamir transform

$$
\begin{array}{lcr}
\underline{\mathcal{P}, \mathsf{sk}, m} & & \underline{\mathcal{V}, \mathsf{pk}} \\
\mathsf{com} \leftarrow_R \mathcal{P}_0(\mathsf{sk}) & & \\
\mathsf{ch} \leftarrow \mathcal{H}(\mathsf{com}, m) & & \\
\mathsf{resp} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}) & \xrightarrow{m, \mathsf{com}, \mathsf{resp}} & \\
& & \mathsf{ch} \leftarrow \mathcal{H}(\mathsf{com}, m) \\
& & b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})
\end{array}
$$

- Unpredictably derive ch from m and com
- Repeat to compensate for adversary ‘guessing right’

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

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

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

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

(evaluating $\mathbf{G} \approx$ evaluating $\mathbf{F}$)

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 $\mathbf{pk} = \mathbf{F}(\mathbf{sk})$ $\Rightarrow (\mathcal{S}_F, \mathbf{pk})$

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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations

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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
- 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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges

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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges
  - Reconstruct commitments
    - r commitments
    - $\sim 1\frac{1}{2}r$ $\mathcal{MQ}$ evaluations

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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges
  - Reconstruct commitments
    - r commitments
    - $\sim 1\frac{1}{2}r$ $\mathcal{MQ}$ evaluations
  - 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
    - Sample r vectors $\mathbf{r}$, $\mathbf{t}$ and $\mathbf{e}$
    - $2r$ commitments, some multiplications in $\mathbb{F}_q$
    - $2r$ $\mathcal{MQ}$ evaluations
- Verifying
  - Reconstruct D, $\mathbf{F}$
  - Reconstruct challenges
  - Reconstruct commitments
    - r commitments
    - $\sim 1\frac{1}{2}r$ $\mathcal{MQ}$ evaluations
  - Check combined commitments hash
- Parameters: k, n, m, $\mathbb{F}_q$, Com, hash functions, PRGs

Hardness of $\mathcal{MQ}$

- Assume $m \geqslant n$, $m \in \mathcal{O}(n)$
- HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- Algebraic techniques with exhaustive search

Hardness of $\mathcal{MQ}$

- Assume $m \geqslant n$, $m \in \mathcal{O}(n)$
- HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- Algebraic techniques with exhaustive search
  - Instantiate with Grover?

Hardness of $\mathcal{MQ}$

- Assume $m \geqslant n$, $m \in \mathcal{O}(n)$
- HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- Algebraic techniques with exhaustive search
  - Instantiate with Grover?
- Analyze both classically and using Grover
  - Classical gates, quantum gates, circuit depth

MQDSS-31-48, MQDSS-31-64

- $k = 256$ (level 1) $k = 384$ (level 3)

MQDSS-31-48, MQDSS-31-64

- $k = 256$ (level 1) $k = 384$ (level 3)
- $n = m = 48$ $n = m = 64$
- $\mathbb{F}_q = \mathbb{F}_{31}$
  - Fast arithmetic, parallelizes nicely
  - Loose reduction $\Rightarrow$ consider best known attacks

MQDSS-31-48, MQDSS-31-64

- $k = 256$ (level 1) $k = 384$ (level 3)
- $n = m = 48$ $n = m = 64$
- $\mathbb{F}_q = \mathbb{F}_{31}$
  - Fast arithmetic, parallelizes nicely
  - Loose reduction $\Rightarrow$ consider best known attacks
- $r = 269$ $r = 403$
  - Follows from k: $2^{-(r \log \frac{2q}{q+1})} < 2^{-k}$

MQDSS-31-48, MQDSS-31-64

- $k = 256$ (level 1) $k = 384$ (level 3)
- $n = m = 48$ $n = m = 64$
- $\mathbb{F}_q = \mathbb{F}_{31}$
  - Fast arithmetic, parallelizes nicely
  - Loose reduction $\Rightarrow$ consider best known attacks
- $r = 269$ $r = 403$
  - Follows from k: $2^{-(r \log \frac{2q}{q+1})} < 2^{-k}$
- SHAKE-256 for commitments / hashes
  - Match output length to k

Implementation considerations

- Very natural internal parallelism

Implementation considerations

- Very natural internal parallelism
- Naively constant-time

Implementation considerations

- Very natural internal parallelism
- Naively constant-time
- Mathematically straight-forward
  - Multiplications and additions in $\mathbb{F}_{31}$

Implementation considerations

- Very natural internal parallelism
- Naively constant-time
- Mathematically straight-forward
  - Multiplications and additions in $\mathbb{F}_{31}$

- Naively slow
  - But still constant-time when optimized

Implementation considerations

- Very natural internal parallelism
- Naively constant-time
- Mathematically straight-forward
  - Multiplications and additions in $\mathbb{F}_{31}$

- Naively slow
  - But still constant-time when optimized
- Expanding **F** is memory-intensive (134 KiB)
  - Problematic on small devices

Implementation considerations

- Very natural internal parallelism
- Naively constant-time
- Mathematically straight-forward
  - Multiplications and additions in $\mathbb{F}_{31}$

- Naively slow
  - But still constant-time when optimized
- Expanding **F** is memory-intensive (134 KiB)
  - Problematic on small devices

In a nutshell..

- $\mathcal{MQ}$-based 5-pass identification scheme
  - Fiat-Shamir transform
- Loose reduction from (only!) $\mathcal{MQ}$ problem
  - Security proof, instead of typical 'break and tweak'

- `MQDSS-31-48`: level 1, 32.1 KiB sigs.
- `MQDSS-31-64`: level 3, 66.2 KiB sigs.
- 62 resp. 88 byte public keys

- Not blazingly fast, not prohibitively slow:
  0.3 - 0.7 ms keygen, 2 - 4 ms sign, 1 - 3 ms verify

  (3.5GHz Haswell, AVX2)

- Only small tweaks since ASIACRYPT 2016 [CHR+16]

References I

Magali Bardet, Jean-Charles Faugère, and Bruno Salvy.
On the complexity of the F5 Gröbner basis algorithm.
Journal of Symbolic Computation, 70(Supplement C):49 – 70, 2015.
https://arxiv.org/pdf/1312.1655.pdf.

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.

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.

References II

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

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.