MQDSS

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

2019-08-22
Second 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 $\mathcal{MQ}$ cryptography
- Very small keys, big signatures

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 $\mathcal{MQ}$ cryptography
- Very small keys, big signatures

- First proposed at ASIACRYPT 2016 [CHR+16]

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 $\mathcal{MQ}$ cryptography
- Very small keys, big signatures
- First proposed at ASIACRYPT 2016 [CHR+16]
- **New in Second Round submission**
  - Reduction of number of rounds
  - Added randomness in commitments
  - more precise analysis of best attacks against $\mathcal{MQ}$

Fiat-Shamir transform

IDS

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

Fiat-Shamir transform

IDS

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

Fiat-Shamir transform

IDS

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

$\downarrow \quad \downarrow \quad \downarrow$

FS signature

Signer

$\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$

$\mathsf{ch} \leftarrow H(m, \mathsf{com})$

$\mathsf{resp} \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch})$

output : $\sigma = (\mathsf{com}, \mathsf{resp})$

Verifier

$\mathsf{ch} \leftarrow H(m, \mathsf{com})$

$b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}, \mathsf{resp})$

output : b

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

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

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

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

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

$$c_1 \leftarrow Com(\mathbf{r}_1, \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0) \quad \xrightarrow{(c_0, c_1)}$$

$$\xleftarrow{\alpha} \quad \alpha \leftarrow_R \mathbb{F}_q$$

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

$$\mathbf{e}_1 \leftarrow \alpha \mathbf{F}(\mathbf{r}_0) - \mathbf{e}_0 \quad \xrightarrow{\mathrm{resp}_1 = (\mathbf{t}_1, \mathbf{e}_1)}$$

$$\xleftarrow{\mathrm{ch}_2} \quad \mathrm{ch}_2 \leftarrow_R \{0, 1\}$$

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

$$\text{Else } \mathrm{resp}_2 \leftarrow \mathbf{r}_1 \quad \xrightarrow{\mathrm{resp}_2}$$

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

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

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

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

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 of message 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 of message 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 of message 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 of message 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 of message 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 of message 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 of message 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: n, m, $\mathbb{F}_q$, Com, hash functions, PRGs

New in Round 2: Parameter Sets

	Sec. cat.	q	n $(= m)$	r	pk (bytes)	sk (bytes)	Signature (bytes)
MQDSS-31-48 (Round 1)	1-2	31	48	135 269	46 62	16 32	20854 32882
MQDSS-31-64 (Round 1)	3-4	31	64	202 403	64 88	24 48	43728 67800

Table: Round 1 parameters in black, Round 2 parameters in red.

- q, $n = m$ chosen using best attacks on $\mathcal{MQ}$
  - q additionally chosen for fast arithmetic

New in Round 2: Parameter Sets

	Sec. cat.	q	n $(= m)$	r	pk (bytes)	sk (bytes)	Signature (bytes)
MQDSS-31-48 (Round 1)	1-2	31	48	135 269	46 62	16 32	20854 32882
MQDSS-31-64 (Round 1)	3-4	31	64	202 403	64 88	24 48	43728 67800

Table: Round 1 parameters in black, Round 2 parameters in red.

- q, $n = m$ chosen using best attacks on $\mathcal{MQ}$
  - q additionally chosen for fast arithmetic
- r chosen such that $2^{-(r \log \frac{2q}{q+1})} < 2^{-k}$
  - **mistake in calculation in Round 1, chose k too large**

New in Round 2: Commitments

- MQDSS uses hash for commitments - instantiated with SHAKE-256

New in Round 2: Commitments

- MQDSS uses hash for commitments - instantiated with SHAKE-256
- In Round 1 proof assumes *statistically hiding commitments*

New in Round 2: Commitments

- MQDSS uses hash for commitments - instantiated with SHAKE-256
- In Round 1 proof assumes *statistically hiding commitments*
  - Requires a lot of randomness: $5\times$ commitment length [Lei18]
  - Round 1 MQDSS does not provide any (dedicated) randomness

New in Round 2: Commitments

- MQDSS uses hash for commitments - instantiated with SHAKE-256
- In Round 1 proof assumes *statistically hiding commitments*
  - Requires a lot of randomness: $5\times$ commitment length [Lei18]
  - Round 1 MQDSS does not provide any (dedicated) randomness
- **Round 2:**
  - Computationally hiding commitments suffices!
  - Proof updated accordingly

New in Round 2: Commitments

- MQDSS uses hash for commitments - instantiated with SHAKE-256
- In Round 1 proof assumes *statistically hiding commitments*
  - Requires a lot of randomness: $5\times$ commitment length [Lei18]
  - Round 1 MQDSS does not provide any (dedicated) randomness
- **Round 2:**
  - Computationally hiding commitments suffices!
  - Proof updated accordingly
  - Still needs randomness - $2\times$ commitment length [Lei18]
  - $\Rightarrow$ adds approx 4KB (10KB) to signature for MQDSS-31-48 (MQDSS-31-64)

Round 2 performance

- Reference implementation

	keygen	signing	verification
MQDSS-31-48	1 192 984	26 630 590	19 840 136
Round 1	1 206 730	52 466 398	38 686 506
MQDSS-31-64	2 767 384	85 268 712	62 306 098
Round 1	2 806 750	169 298 364	123 239 874

Table: Round 1 performance in black, Round 2 performance in red.

Round 2 performance

- Reference implementation

	keygen	signing	verification
MQDSS-31-48	1 192 984	26 630 590	19 840 136
Round 1	1 206 730	52 466 398	38 686 506
MQDSS-31-64	2 767 384	85 268 712	62 306 098
Round 1	2 806 750	169 298 364	123 239 874

Table: Round 1 performance in black, Round 2 performance in red.

- AVX2 implementation

	keygen	signing	verification
MQDSS-31-48	1 074 644	3 816 106	2 551 270
MQDSS-31-64	2 491 050	9 047 148	6 132 948

Round 2: More precise analysis of hardness of $\mathcal{MQ}$

- Best strategy: Algebraic techniques with exhaustive search
  - HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]

Round 2: More precise analysis of hardness of $\mathcal{MQ}$

- Best strategy: Algebraic techniques with exhaustive search
  - HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- Analyze both classically and using Grover
  - Classical gates, quantum gates, circuit depth

Round 2: More precise analysis of hardness of $\mathcal{MQ}$

- Best strategy: Algebraic techniques with exhaustive search
  - HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- Analyze both classically and using Grover
  - Classical gates, quantum gates, circuit depth
  - minor changes in **Round 2** - more precise analysis
  - no influence to security of parameter sets

Recent attack

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48

Recent attack

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48
- Can be mitigated by $\approx 1.4\times$(number of rounds)

Recent attack

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48
- Can be mitigated by $\approx 1.4\times$(number of rounds)
- **Proof still valid!**
  - Attack is result of not taking into account non-tightness of proof for choosing parameters

Recent attack

- August 2019, Daniel Kales and Greg Zaverucha - forgery in approx. 2^{95} hash calls for MQDSS-31-48
- Can be mitigated by $\approx 1.4\times$(number of rounds)
- **Proof still valid!**
  - Attack is result of not taking into account non-tightness of proof for choosing parameters
- **New parameters after attack (estimate):**

	Sec. cat.	q	n	r	pk	sk	Signature
MQDSS-31-48 (new)	1-2	31	48	184	46B	16B	28400B
Round 1				269	62B	32B	32882B
MQDSS-31-64 (new)	3-4	31	64	277	64B	24B	59928B
Round 1				403	88B	48B	67800B

Table: Round 1 parameters in black, New parameters (attack fixed) in red.

In a nutshell..

- Fiat-Shamir transform from $\mathcal{MQ}$-based 5-pass identification scheme
- Security proof in ROM, instead of typical ‘break and tweak’ in $\mathcal{MQ}$ cryptography
- Very small keys, big signatures

In a nutshell..

- Fiat-Shamir transform from $\mathcal{MQ}$-based 5-pass identification scheme
- Security proof in ROM, instead of typical 'break and tweak' in $\mathcal{MQ}$ cryptography
- Very small keys, big signatures
- **Main improvement in Round 2: Smaller signatures**
  - Even after: Recent attack + Added randomness in commitments

	Sec. cat.	q	n	r	pk	sk	Signature
MQDSS-31-48 (new)	1-2	31	48	184	46B	16B	28400B
Round 1				269	62B	32B	32882B
MQDSS-31-64 (new)	3-4	31	64	277	64B	24B	59928B
Round 1				403	88B	48B	67800B

Table: Round 1 parameters in black, New parameters (attack fixed) in red.

In a nutshell..

- Fiat-Shamir transform from $\mathcal{MQ}$-based 5-pass identification scheme
- Security proof in ROM, instead of typical 'break and tweak' in $\mathcal{MQ}$ cryptography
- Very small keys, big signatures
- **Main improvement in Round 2: Smaller signatures**
  - Even after: Recent attack + Added randomness in commitments

	Sec. cat.	q	n	r	pk	sk	Signature
MQDSS-31-48 (new)	1-2	31	48	184	46B	16B	28400B
Round 1				269	62B	32B	32882B
MQDSS-31-64 (new)	3-4	31	64	277	64B	24B	59928B
Round 1				403	88B	48B	67800B

Table: Round 1 parameters in black, New parameters (attack fixed) in red.

Thank you for your attention!

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.

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.

Dominik Leichtle.
Post-quantum signatures from identification schemes.
Master Thesis, Technicshe Universiteit Eindhoven, 2018.

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.

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