

# MQDSS

Ming-Shing Chen<sup>1</sup>, Andreas Hülsing<sup>2</sup>, **Joost Rijneveld**<sup>3</sup>,  
Simona Samardjiska<sup>3,4</sup>, and Peter Schwabe<sup>3</sup>

<sup>1</sup> National Taiwan University / Academia Sinica, Taipei, Taiwan

<sup>2</sup> Technische Universiteit Eindhoven, Eindhoven, The Netherlands

<sup>3</sup> Radboud University, Nijmegen, The Netherlands

<sup>4</sup> “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..

- ▶  $MQ$ -based 5-pass identification scheme
  - ▶ Fiat-Shamir transform
- ▶ Loose reduction from (only!)  $MQ$  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

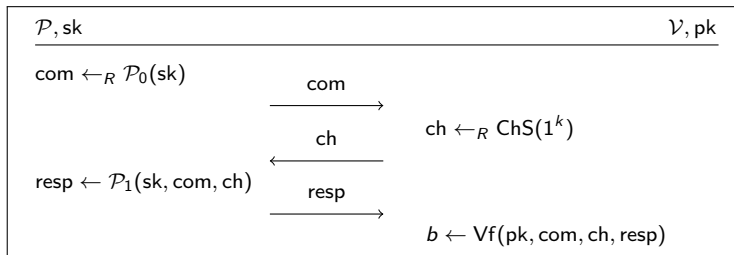
## In a nutshell..

- ▶  $MQ$ -based 5-pass identification scheme
  - ▶ Fiat-Shamir transform
- ▶ Loose reduction from (only!)  $MQ$  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..

- ▶  $MQ$ -based 5-pass identification scheme
  - ▶ Fiat-Shamir transform
- ▶ Loose reduction from (only!)  $MQ$  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<sup>+</sup>16]

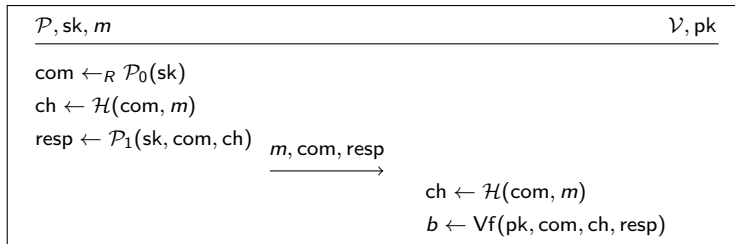
# Canonical Identification Schemes



Informally:

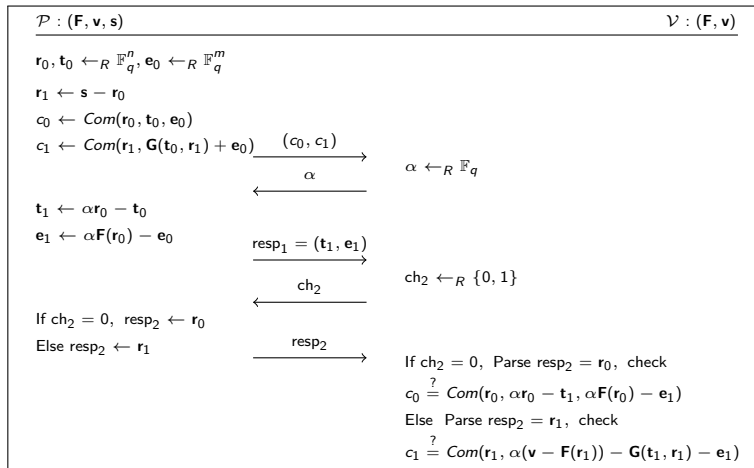
1. Prover commits to some (randomized) value derived from  $sk$
2. Verifier picks a challenge 'ch'
3. Prover computes response 'resp'
4. Verifier checks if response matches challenge

# Fiat-Shamir transform



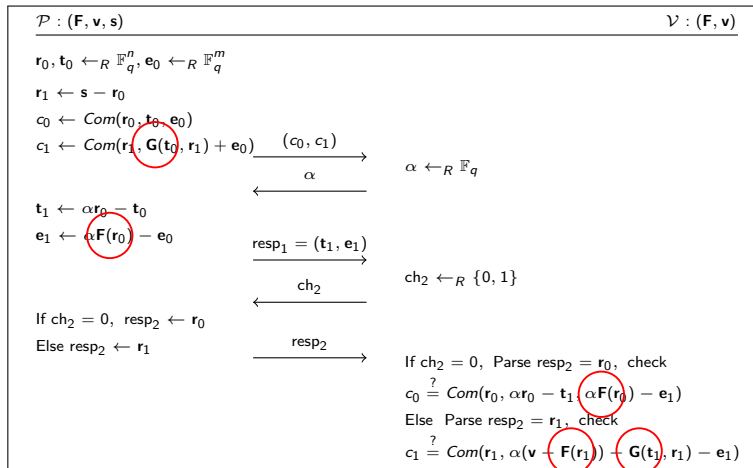
- ▶ Unpredictably derive  $\text{ch}$  from  $m$  and  $\text{com}$
- ▶ Repeat to compensate for adversary 'guessing right'

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





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



(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  $\mathbb{F}_q$
    - ▶  $2r$   $MQ$  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  $\mathbb{F}_q$
    - ▶  $2r$   $MQ$  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  $\mathcal{S}_F \in \{0, 1\}^k$ ,  $\mathbf{sk} \in \mathbb{F}_q^n \quad \Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - ▶ Expand  $\mathcal{S}_F$  to  $\mathbf{F}$ , compute  $\mathbf{pk} = \mathbf{F}(\mathbf{sk}) \quad \Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- ▶ 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  $\mathcal{S}_F \in \{0, 1\}^k$ ,  $\mathbf{sk} \in \mathbb{F}_q^n \quad \Rightarrow (\mathcal{S}_F, \mathbf{sk})$
  - ▶ Expand  $\mathcal{S}_F$  to  $\mathbf{F}$ , compute  $\mathbf{pk} = \mathbf{F}(\mathbf{sk}) \quad \Rightarrow (\mathcal{S}_F, \mathbf{pk})$
- ▶ 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  $m \geq n$ ,  $m \in \mathcal{O}(n)$
- ▶ HybridF5 [BFS15], BooleanSolve [BFSS13], Crossbred [JV17]
- ▶ Algebraic techniques with exhaustive search

## Hardness of $\mathcal{MQ}$

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

## Hardness of $\mathcal{MQ}$

- ▶ Assume  $m \geq n$ ,  $m \in \mathcal{O}(n)$
- ▶ 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  $\mathbf{F}$  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  $\mathbf{F}$  is memory-intensive (134 KiB)
  - ▶ Problematic on small devices

## In a nutshell..

- ▶  $MQ$ -based 5-pass identification scheme
  - ▶ Fiat-Shamir transform
- ▶ Loose reduction from (only!)  $MQ$  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<sup>+</sup>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](http://www-polsys.lip6.fr/~jcf/Papers/BFSS12.pdf).



Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe.

From 5-pass  $MQ$ -based identification to  $MQ$ -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:](https://www.iacr.org/archive/crypto2011/68410703/68410703.pdf)

[//www.iacr.org/archive/crypto2011/68410703/68410703.pdf](https://www.iacr.org/archive/crypto2011/68410703/68410703.pdf).