



COSIC

Post-Quantum Cryptography

Performance, Side Channels and Implementation Pitfalls



Quinten Norga

COSIC, KU Leuven

October 6, 2026

Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC
- ④ MPC-in-the-Head-based PQC
- ⑤ Outlook: AI and (PQ) Cryptography



Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC
- ④ MPC-in-the-Head-based PQC
- ⑤ Outlook: AI and (PQ) Cryptography



PQC Standardization Competitions

Standardization of post-quantum key-encapsulation mechanisms (KEMs) and digital signature algorithms (DSAs) is well underway, all *around the world*.

	NIST ¹ 	NIS/NSR ² 	ICCS ³ 
Status	Ongoing	Completed	Ongoing
# Published	3	4	0
# Upcoming	≥ 3	0	?

¹<https://csrc.nist.gov/projects/post-quantum-cryptography>

²https://www.kpqc.or.kr/contents/03_exhibit/sub_03.html

³<https://www.niccs.org.cn/en/>



PQC Standardization Competitions

Standardization of post-quantum key-encapsulation mechanisms (KEMs) and digital signature algorithms (DSAs) is well underway, all *around the world*.

	NIST ¹ 	NIS/NSR ² 	ICCS ³ 
Status	Ongoing	Completed	Ongoing
# Published	3	4	0
# Upcoming	≥ 3	0	?

► Focus of community/industry: **adoption and deployment**.

¹<https://csrc.nist.gov/projects/post-quantum-cryptography>

²https://www.kpqc.or.kr/contents/03_exhibit/sub_03.html

³<https://www.niccs.org.cn/en/>



PQC Standards: Algorithm Diversity

Code	Lattice	Hash	MPC-in-the-Head
Classic McEliece**	ML-KEM	SLH-DSA	AIMer
HQC-KEM	ML-DSA		
	FrodoKEM		
	SMAUG-T		
	NTRU+		
	HAETAE		
	FN-DSA		



PQC Standards: Algorithm Diversity

Code	Lattice	Hash	MPC-in-the-Head
Classic McEliece**	ML-KEM	SLH-DSA	AIMer
HQC-KEM	ML-DSA		
	FrodoKEM		
	SMAUG-T		
	NTRU+		
	HAETAE		
	FN-DSA		

Focus of this talk: many schemes and families, each with distinct implementation pitfalls and challenges when implementing *securely and efficiently*.



NIST PQC Standardization

NIST's PQC standardization effort has been ongoing since 2016.

<i>April 28, 2016</i>	•	NIST IR 8105, Report on Post-Quantum Cryptography [CJL⁺16]
<i>January 31, 2019</i>	•	NIST IR 8240, Round 1 Status Report [AASA⁺19]
<i>July 22, 2020</i>	•	NIST IR 8309, Round 2 Status Report [AASA⁺20]
<i>July 5, 2022</i>	•	NIST IR 8413, Round 3 Status Report [AAC⁺22]
<i>August 13, 2024</i>	•	FIPS 203 (ML-KEM) [Nat24b], FIPS 204 (ML-DSA) [Nat24a], FIPS205 (SLH-DSA) [Nat24c]
<i>March 11, 2025</i>	•	NIST IR 8545, Round 4 Status Report [ABC⁺25]
<hr/>		
<i>2027</i>	•	FIPS 206 (FN-DSA)
<i>2028</i>	•	FIPS 207 (HQC-KEM)



NIST PQC Standardization

NIST's PQC standardization effort has been ongoing since 2016.

<i>April 28, 2016</i>	•	NIST IR 8105, Report on Post-Quantum Cryptography [CJL⁺16]
<i>January 31, 2019</i>	•	NIST IR 8240, Round 1 Status Report [AASA⁺19]
<i>July 22, 2020</i>	•	NIST IR 8309, Round 2 Status Report [AASA⁺20]
<i>July 5, 2022</i>	•	NIST IR 8413, Round 3 Status Report [AAC⁺22]
<i>August 13, 2024</i>	•	FIPS 203 (ML-KEM) [Nat24b], FIPS 204 (ML-DSA) [Nat24a], FIPS205 (SLH-DSA) [Nat24c]
<i>March 11, 2025</i>	•	NIST IR 8545, Round 4 Status Report [ABC⁺25]
<hr/>		
<i>2027</i>	•	FIPS 206 (FN-DSA)
<i>2028</i>	•	FIPS 207 (HQC-KEM)

- First PQC standards released in 2024, majority **lattice-based schemes**.



FN-DSA: Latest Updates

Ray Perlner (NIST) on pqc-forum (September 28th 2026)

In light of recent work on implementing Falcon [...] we think it's a good idea to update our plan for specifying FN-DSA in draft FIPS 206. In particular [...] specifies the use of **fixed-point arithmetic in both key generation and signing**.

⁴<https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/7z-y5qp-W2Y/m/a11CmUsuBQAJ>



FN-DSA: Latest Updates

Ray Perlner (NIST) on pqc-forum (September 28th 2026)

In light of recent work on implementing Falcon [...] we think it's a good idea to update our plan for specifying FN-DSA in draft FIPS 206. In particular [...] specifies the use of **fixed-point arithmetic in both key generation and signing**.

Observations⁴:

1. NIST are being more *methodical* (see ML-KEM, Classic McEliece)
2. Reduces specification and implementation complexity.
3. Cost of side-channel countermeasures is unclear.

⁴<https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/7z-y5qp-W2Y/m/a11CmUsuBQAJ>



HQC-KEM: Latest Updates

Angela Robinson (NIST) on pqc-forum (August 3rd 2026)

NIST is leaning toward specifying that the **only [key] format to be used is the seed**. Caching expanded keys would not be allowed as there is minimal performance benefit to doing so [...] This is a different approach from FIPS 203 [...]

⁵https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/YI_whYYZ9RQ/m/752xQUmPBQAJ



HQC-KEM: Latest Updates

Angela Robinson (NIST) on pqc-forum (August 3rd 2026)

NIST is leaning toward specifying that the **only [key] format to be used is the seed**. Caching expanded keys would not be allowed as there is minimal performance benefit to doing so [...] This is a different approach from FIPS 203 [...]

Observations⁵:

1. Simplify implemetation (and adoption).
2. Caching of specific expanded values allowed.

⁵https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/YI_whYYZ9RQ/m/752xQUmPBQAJ



NIST Additional PQC DSA Standardization

NIST announced *Additional PQC DSA* competition in 2022, seeking to diversify its signature portfolio.

July 5, 2022	•	NIST IR 8413, Announcement of CfP Additional DSAs [AAC+22]
October 24, 2024	•	NIST IR 8528, DSA Round 1 Status Report [ABC+24]
May 14, 2026	•	NIST IR 8610, DSA Round 2 Status Report [ABC+26]
July 26, 2026	•	HAWK is withdrawn.
<hr/>		
2028	•	Start Round 4
2030-2035	•	Additional DSA Standards



NIST Additional PQC DSA Standardization

NIST announced *Additional PQC DSA* competition in 2022, seeking to diversify its signature portfolio.

July 5, 2022	•	NIST IR 8413, Announcement of CfP Additional DSAs [AAC ⁺ 22]
October 24, 2024	•	NIST IR 8528, DSA Round 1 Status Report [ABC ⁺ 24]
May 14, 2026	•	NIST IR 8610, DSA Round 2 Status Report [ABC ⁺ 26]
July 26, 2026	•	HAWK is withdrawn.
<hr/>		
2028	•	Start Round 4
2030-2035	•	Additional DSA Standards

- Eight Round 3 candidates remain, four are **multivariate-based schemes** and three are **MPCitH-based schemes**.



Overview of Round 3 Candidates

Code	Lattice	MPC-in-the-Head	Multivariate	Isogeny	Symmetric
CROSS LESS	HAWK	Mirath MQOM PERK RYDE SDitH	UOV MAYO QR-UOV SNOVA	SQIsign	FAEST
0	0	2	4	1	1



Round 3 Evaluation

NIST IR 8610

*“NIST is also interested in additional **performance data** on each of the candidates, including optimized implementations written in **assembly code** or using **instruction set extensions**, analyses of the implementation suitability of candidate algorithms in **constrained platforms**, and performance data for **hardware** implementations.”*



Round 3 Evaluation

NIST IR 8610

*“NIST is also interested in additional **performance data** on each of the candidates, including optimized implementations written in **assembly code** or using **instruction set extensions**, analyses of the implementation suitability of candidate algorithms in **constrained platforms**, and performance data for **hardware** implementations.”*

- Focus will increasingly shift towards **implementation** aspects



Round 3 Evaluation

NIST IR 8610

*“NIST is also interested in additional **performance data** on each of the candidates, including optimized implementations written in **assembly code** or using **instruction set extensions**, analyses of the implementation suitability of candidate algorithms in **constrained platforms**, and performance data for **hardware** implementations.”*

- ▶ Focus will increasingly shift towards **implementation** aspects
- ▶ Efficiency on various platforms



Round 3 Evaluation

NIST IR 8610

*“NIST is also interested in additional **performance data** on each of the candidates, including optimized implementations written in **assembly code** or using **instruction set extensions**, analyses of the implementation suitability of candidate algorithms in **constrained platforms**, and performance data for **hardware** implementations.”*

- ▶ Focus will increasingly shift towards **implementation** aspects
- ▶ Efficiency on various platforms
- ▶ Attention for side-channel attacks and defenses



ICCS PQC Standardization

Round 1 candidates were recently announced.

DSAs	KEMs	KEXs
34	41	9



ICCS PQC Standardization

Round 1 candidates were recently announced.

DSAs	KEMs	KEXs
34	41	9

- Similar pattern as NIST competition, a wide variety of *crypto families*.

Code	Lattice	Isogeny	Multivariate	MPCitH	Hash	Symmetric	Other
19	43	5	4	3	4	5	1



ICCS PQC Standardization

Currently known information:



ICCS PQC Standardization

Currently known information:

- ▶ 84/150+ accepted submissions



ICCS PQC Standardization

Currently known information:

- ▶ 84/150+ accepted submissions
- ▶ Round 1 until June 2027



ICCS PQC Standardization

Currently known information:

- ▶ 84/150+ accepted submissions
- ▶ Round 1 until June 2027
- ▶ Focus: security/cryptanalysis (?)



ICCS PQC Standardization

Currently known information:

- ▶ 84/150+ accepted submissions
- ▶ Round 1 until June 2027
- ▶ Focus: security/cryptanalysis (?)

This is also the first PQC standardization competition with heavy AI impact (more on this later).



Challenges

Many standardized PQC algorithms, many algorithm families.



Challenges

Many standardized PQC algorithms, many algorithm families.

1. Efficient implementations optimized for a wide range of target platforms.



Challenges

Many standardized PQC algorithms, many algorithm families.

1. Efficient implementations optimized for a wide range of target platforms.
 - FPGA, ASIC, MCU, GPU, CPU...



Challenges

Many standardized PQC algorithms, many algorithm families.

1. Efficient implementations optimized for a wide range of target platforms.
 - FPGA, ASIC, MCU, GPU, CPU...
2. Countermeasures against physical attacks



Challenges

Many standardized PQC algorithms, many algorithm families.

1. Efficient implementations optimized for a wide range of target platforms.
 - FPGA, ASIC, MCU, GPU, CPU...
2. Countermeasures against physical attacks
 - Passive, active and combined attacks



Challenges

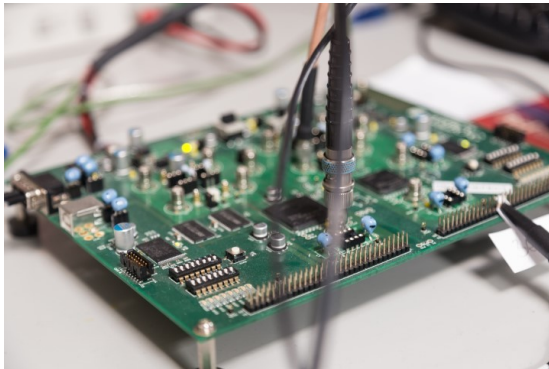
Many standardized PQC algorithms, many algorithm families.

1. Efficient implementations optimized for a wide range of target platforms.
 - FPGA, ASIC, MCU, GPU, CPU...
2. Countermeasures against physical attacks
 - Passive, active and combined attacks

→ **Efficient and secure** implementations.



Physical Attacks



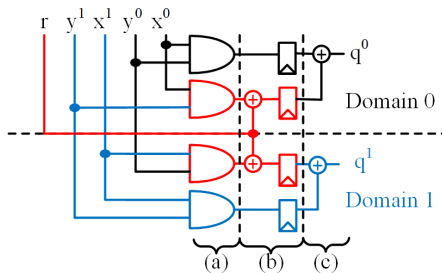
Examples: differential or correlation power analysis, fault injection attack, combined attacks...



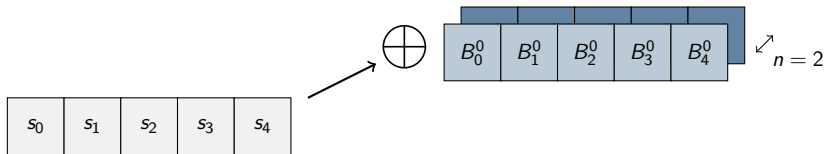
Masking

Countermeasure against differential power analysis (DPA)

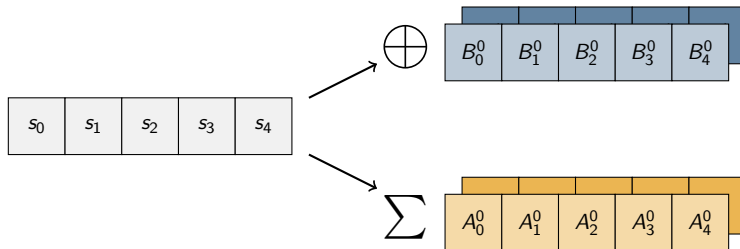
- ▶ Based on **secret sharing**
- ▶ Blind computations through **randomness** and **registers**
- ▶ High overhead
(randomness/latency/area/memory footprint)



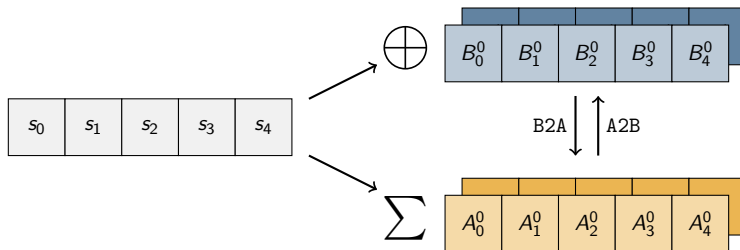
Boolean and Arithmetic Masking



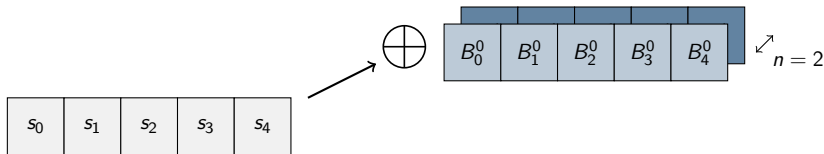
Boolean and Arithmetic Masking



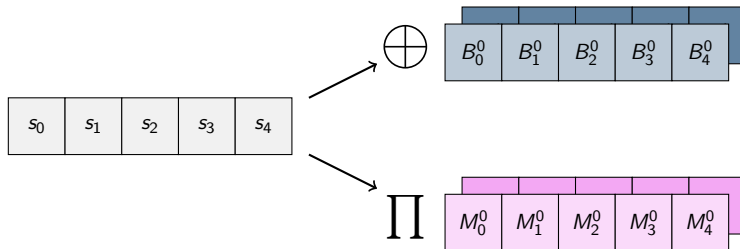
Boolean and Arithmetic Masking



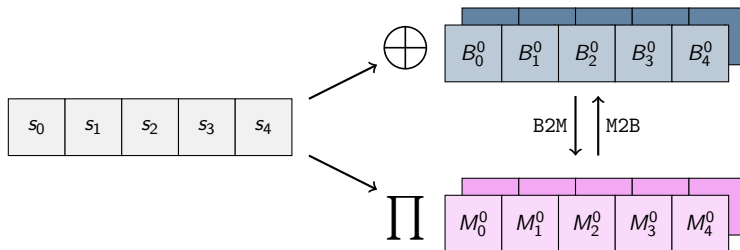
Boolean and Multiplicative Masking



Boolean and Multiplicative Masking



Boolean and Multiplicative Masking



Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC
- ④ MPC-in-the-Head-based PQC
- ⑤ Outlook: AI and (PQ) Cryptography



Lattice-based PQC

ML-KEM & ML-DSA

- ▶ Good performance: polynomial arithmetic and SHA-3

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

Category: Computer Security

Subcategory: Cryptography

FIPS 204

Federal Information Processing Standards Publication

Module-Lattice-Based Digital Signature Standard

Category: Computer Security

Subcategory: Cryptography



Lattice-based PQC

ML-KEM & ML-DSA

- ▶ Good performance: polynomial arithmetic and SHA-3

Real-world deployment:

- ▶ (Protection against) **Physical attacks**

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

Category: Computer Security

Subcategory: Cryptography

FIPS 204

Federal Information Processing Standards Publication

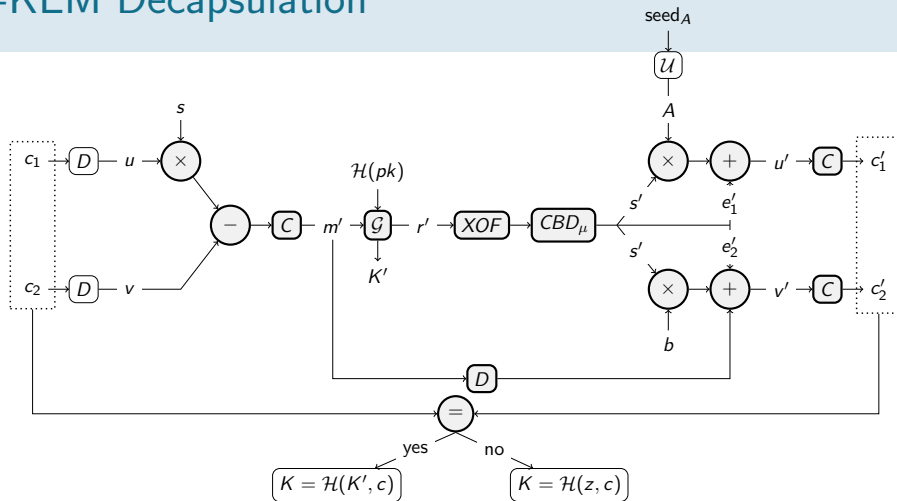
Module-Lattice-Based Digital Signature Standard

Category: Computer Security

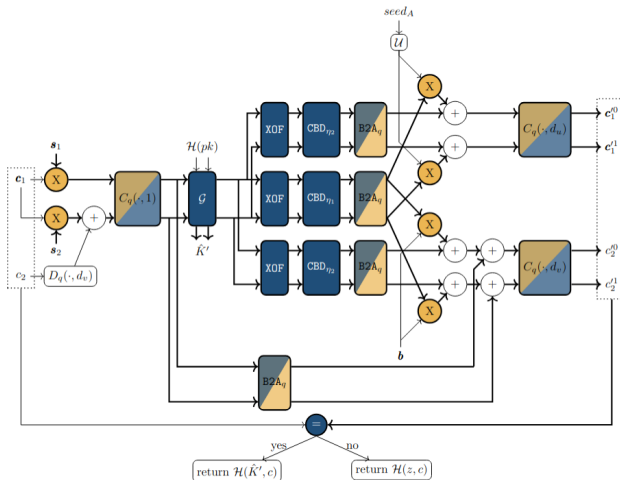
Subcategory: Cryptography



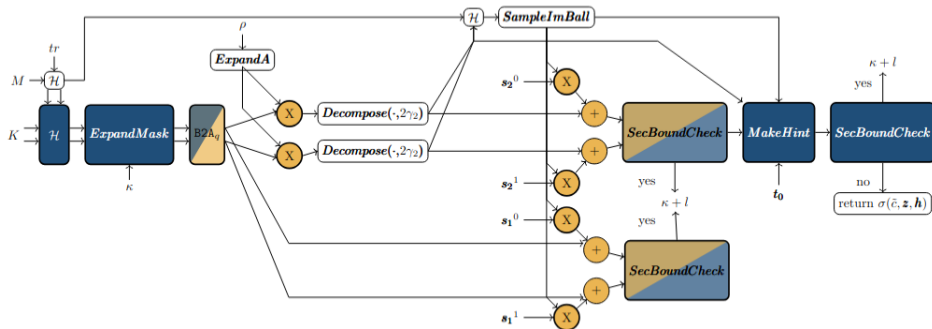
ML-KEM Decapsulation



Masked ML-KEM Decapsulation



Masked ML-DSA Signing



Masking Lattice-based PQC

Masking Lattice-based PQC (ML-KEM.Decaps and ML-DSA.Sign) requires a mix of **arithmetic** and **Boolean** sharing.

- ▶ **Polynomial** arithmetic (e.g., PolMult): $x = \sum_{i=0}^d x^{\{i\}}$
- ▶ **Bitwise** arithmetic (e.g., Hashing): $x = \bigoplus_{i=0}^d x^{\{i\}}$



Masking Lattice-based PQC

Masking Lattice-based PQC (ML-KEM.Decaps and ML-DSA.Sign) requires a mix of **arithmetic** and **Boolean** sharing.

- ▶ **Polynomial** arithmetic (e.g., PolMult): $x = \sum_{i=0}^d x^{\{i\}}$
- ▶ **Bitwise** arithmetic (e.g., Hashing): $x = \bigoplus_{i=0}^d x^{\{i\}}$

Need **A2B** and **B2A**!

- ▶ Performance bottleneck $\rightarrow$ HW acceleration



X2X: A2B and B2A in HW

The X2X accelerator⁶ supports

- ▶ **ANY** protection order d
- ▶ **ANY** modulus p or q
- ▶ **ANY** operation (A2B or B2A)

⁶<https://github.com/KULeuven-COSIC/X2X>



X2X: A2B and B2A in HW

The X2X accelerator⁶ supports

- ▶ **ANY** protection order d
- ▶ **ANY** modulus p or q
- ▶ **ANY** operation (A2B or B2A)
- ▶ Reduce randomness & area cost (up to 62%, 45-60%), improved throughput (29-92%)
- ▶ Mix of arithmetic and circuit-level optimizations.
- ▶ Formal and practical (TVLA) security validation.

⁶<https://github.com/KULeuven-COSIC/X2X>



Secure Addition: SecADD

$$s^{\{0:d\}} = x^{\{0:d\}} + y^{\{0:d\}} \bmod q = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y^{\{i\}} \bmod q$$

- "Arithmetic addition on Boolean shares"

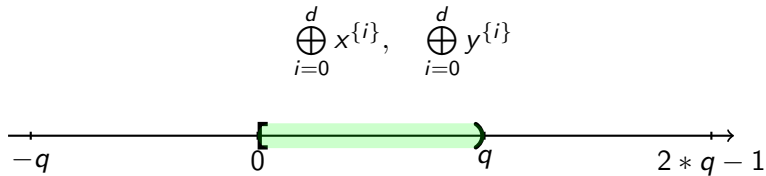


Secure Addition: SecADD

$$s^{\{0:d\}} = x^{\{0:d\}} + y^{\{0:d\}} \bmod q = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y^{\{i\}} \bmod q$$

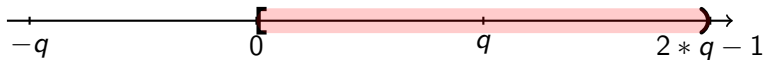
- "Arithmetic addition on Boolean shares"



SecADD_q: Typical Approach

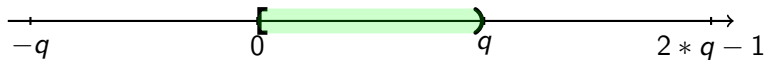
SecADD_q: Typical Approach

$$\text{Step 1: } s^{\{0:d\}} = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y^{\{i\}}$$



SecADD_q: Typical Approach

$$\text{Step 2: } s^{\{0:d\}} = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y^{\{i\}} \bmod q$$



► SecMUX [BBE⁺18] or $2 \times$ SecADD [Cas22]



A2B

► $A2B \approx \text{SecADD}(\text{SecADD}(\dots))$

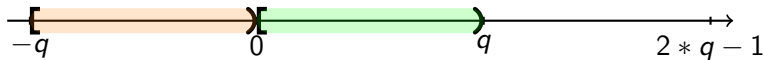
$$B^{\{0:d\}} = z^{\{0\}} + z^{\{1\}} + \dots + z^{\{d\}}$$

► $\uparrow d \rightarrow \uparrow \# \text{SecADD}$



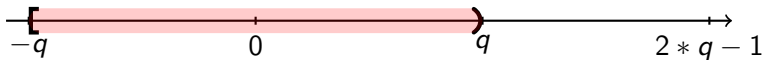
SecADDChain_q

$$\text{Step } 0: \quad \bigoplus_{i=0}^d x^{\{i\}}, \quad \bigoplus_{i=0}^d y^{\{i\}} - q$$



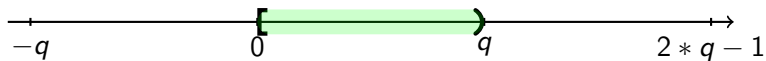
SecADDChain_q

$$\text{Step 1: } s^{\{0:d\}} = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y'^{\{i\}}$$



SecADDChain_q

$$\text{Step 2: } s^{\{0:d\}} = \bigoplus_{i=0}^d x^{\{i\}} + \bigoplus_{i=0}^d y^{\{i\}} \bmod q$$

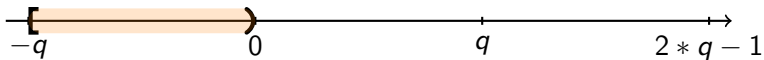


► **1** × SecADD



SecADDChain_q

$$\text{Step 2: } s'^{\{0:d\}} = s^{\{0:d\}} - q$$



- ▶ $1 \times \text{SecADD}$
- ▶ Interleave 2 options



B2A

► $B2A \approx A2B \ \& \ SecADD^d$

Sample : $R^0, \ R^1 \ \dots \ R^{d-1}$



B2A

► $B2A \approx A2B$ & SecADD^d

Sample : $R^0, R^1 \dots R^{d-1}$

$$B^{\{0:d\}} = R^{\{0\}} + R^{\{1\}} + \dots + 0$$



B2A

► $B2A \approx A2B$ & SecADD^d

Sample : $R^0, R^1 \dots R^{d-1}$

$$B^{\{0:d\}} = R^{\{0\}} + R^{\{1\}} + \dots + 0$$

$$z^{\{0:d\}} = B^{\{0:d\}} + x^{\{0:d\}}$$



B2X2A & X2B

► $B2X2A \approx X2B$

Sample : $R^0, R^1 \dots R^{d-1}$



B2X2A & X2B

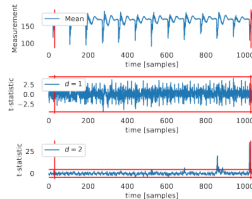
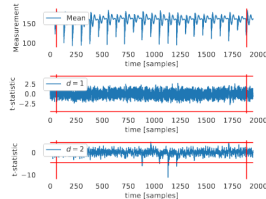
► B2X2A $\approx$ X2B

Sample : $R^0, R^1 \dots R^{d-1}$

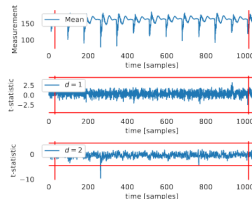
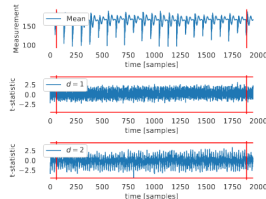
$$z^{\{0:d\}} = R^{\{0\}} + R^{\{1\}} + \dots + x^{\{0:d\}}$$



Security Evaluation: TVLA in Lab

(a) A2B mod 2^{13} .

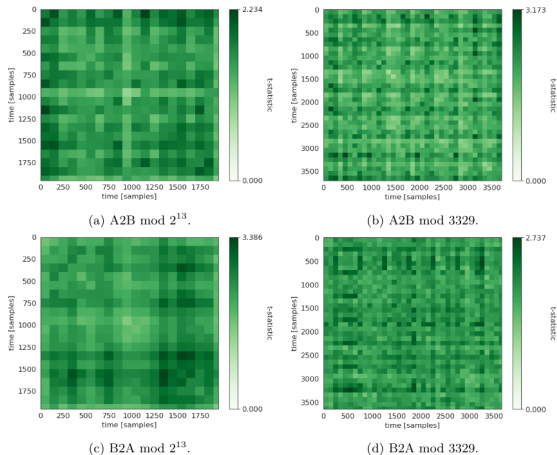
(b) A2B mod 3329.

(c) B2A mod 2^{13} .

(d) B2A mod 3329.



Security Evaluation: TVLA in Lab



Agile HW-SW Co-Design for PQC

Challenge: Unified HW architecture for ML-KEM and ML-DSA (and more).

⁷<https://github.com/cisec-su/RISQrypt>



Agile HW-SW Co-Design for PQC

Challenge: Unified HW architecture for ML-KEM and ML-DSA (and more).

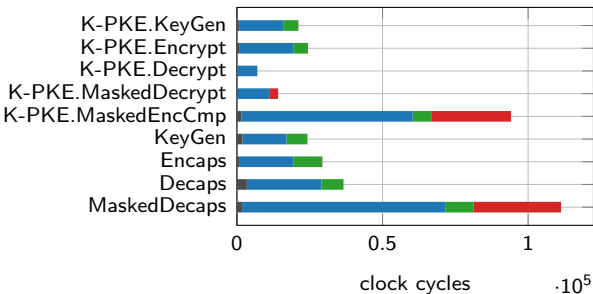
RISQrypt⁷

- ▶ Unified and reprogrammable HW-SW co-design
- ▶ Poly arithmetic & SHA-3 HW acceleration
- ▶ Integrated SCA countermeasures (X2X engine)
- ▶ Extended to SLH-DSA and Falcon (agility)

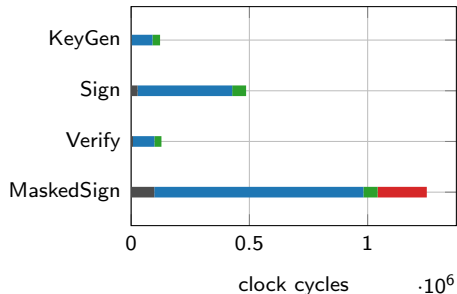
⁷<https://github.com/cisec-su/RISQrypt>



Performance Analysis



ML-KEM-768



ML-DSA-65



ML-DSA

ML-DSA signing $\approx$ rejection sampling

- ▶ Implementation challenge (e.g., batched signing, early stop)
- ▶ Non-fixed nb. iterations
- ▶ Polynomial arithmetic & SHA-3



ML-DSA

ML-DSA signing $\approx$ rejection sampling

- ▶ Implementation challenge (e.g., batched signing, early stop)
- ▶ Non-fixed nb. iterations
- ▶ Polynomial arithmetic & SHA-3

Secure implementation $\rightarrow$ additional challenges

- ▶ Improve performance through HW acceleration



ML-DSA-OSH

An open-source, unified (RTL) implementation of ML-DSA⁸, enables future work.

⁸<https://github.com/KULeuven-COSIC/ML-DSA-OSH>, based on [BNG21]



ML-DSA-OSH

An open-source, unified (RTL) implementation of ML-DSA⁸, enables future work.

- ▶ KeyGen, SigGen, and Verify (NIST II, III, and V)

⁸<https://github.com/KULeuven-COSIC/ML-DSA-OSH>, based on [BNG21]



ML-DSA-OSH

An open-source, unified (RTL) implementation of ML-DSA⁸, enables future work.

- ▶ KeyGen, SigGen, and Verify (NIST II, III, and V)
- ▶ Prototype optimizations, architectures, ...

⁸<https://github.com/KULeuven-COSIC/ML-DSA-OSH>, based on [BNG21]



ML-DSA-OSH

An open-source, unified (RTL) implementation of ML-DSA⁸, enables future work.

- ▶ KeyGen, SigGen, and Verify (NIST II, III, and V)
- ▶ Prototype optimizations, architectures, ...
- ▶ Integration & validation of side-channel attacks and countermeasures.

⁸<https://github.com/KULeuven-COSIC/ML-DSA-OSH>, based on [BNG21]



ML-DSA-OSH

An open-source, unified (RTL) implementation of ML-DSA⁸, enables future work.

- ▶ KeyGen, SigGen, and Verify (NIST II, III, and V)
- ▶ Prototype optimizations, architectures, ...
- ▶ Integration & validation of side-channel attacks and countermeasures.
- ▶ Demonstrate signing latency can be reduced (16-36 %) through optimized operation scheduling & pre-computation.

⁸<https://github.com/KULeuven-COSIC/ML-DSA-OSH>, based on [BNG21]



Summary

Implementation and countermeasures for ML-KEM and ML-DSA have been widely studied



Summary

Implementation and countermeasures for ML-KEM and ML-DSA have been widely studied

- ▶ Mostly component-wise solutions



Summary

Implementation and countermeasures for ML-KEM and ML-DSA have been widely studied

- ▶ Mostly component-wise solutions
- ▶ Integration is challenging, yet critical (“*crypto agility*”)



Summary

Implementation and countermeasures for ML-KEM and ML-DSA have been widely studied

- ▶ Mostly component-wise solutions
- ▶ Integration is challenging, yet critical (*“crypto agility”*)
- ▶ Cost from SCA countermeasures can be offset through HW acceleration



Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC**
- ④ MPC-in-the-Head-based PQC
- ⑤ Outlook: AI and (PQ) Cryptography



NIST Candidates

UOV, MAYO, QR-UOV, and SNOVA.

- ▶ Significant advances in cryptanalysis⁹, affecting **all** multivariate-based candidates.
 - “*NIST anticipates a longer timeline for the potential standardization [...] and is unlikely to standardize them without a further round of evaluation.*”
- ▶ Overall, most **mature**. Each scheme has **distinct** benefits (key size, hard problem).

⁹Wedges, Oil, and Vinegar (Lars Ran), EUROCRYPT 2026 [[Ran26](#)]



NIST Candidates

UOV, MAYO, QR-UOV, and SNOVA.

- ▶ Significant advances in cryptanalysis⁹, affecting **all** multivariate-based candidates.
 - *“NIST anticipates a longer timeline for the potential standardization [...] and is unlikely to standardize them without a further round of evaluation.”*
- ▶ Overall, most **mature**. Each scheme has **distinct** benefits (key size, hard problem).
- ▶ Open questions: protection against physical attacks and performance on various platforms.

⁹Wedges, Oil, and Vinegar (Lars Ran), EUROCRYPT 2026 [[Ran26](#)]



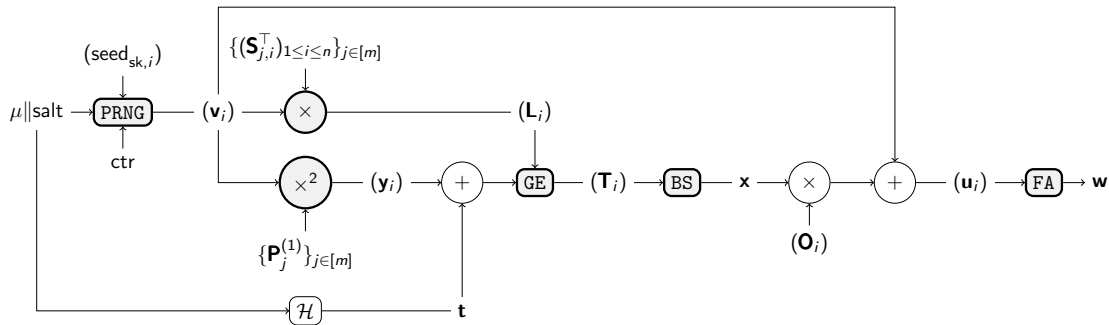
UOV

► Fast SigGen and Verify

Algorithm	EC-DSA	ML-DSA <i>FIPS 204</i> [Nat24a]	SLH-DSA <i>FIPS 205</i> [Nat24c]	FN-DSA <i>FIPS 206</i> [PFH ⁺ 22]	UOV [BCD ⁺ 24]
Signature [Bytes]	128	4 627 (36.1×)	29 792 (232.8×)	1 280 (10.0×)	260 (2.0×)
Public Key [Bytes]	66	2 592 (39.3×)	64 (0.97×)	1 793 (27.2×)	446 992 (461.0×)
Implementation Complexity					

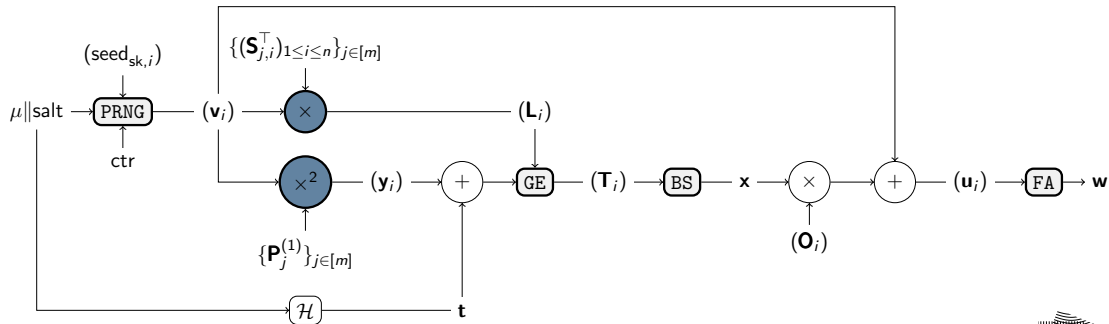


(Masked) UOV Signing



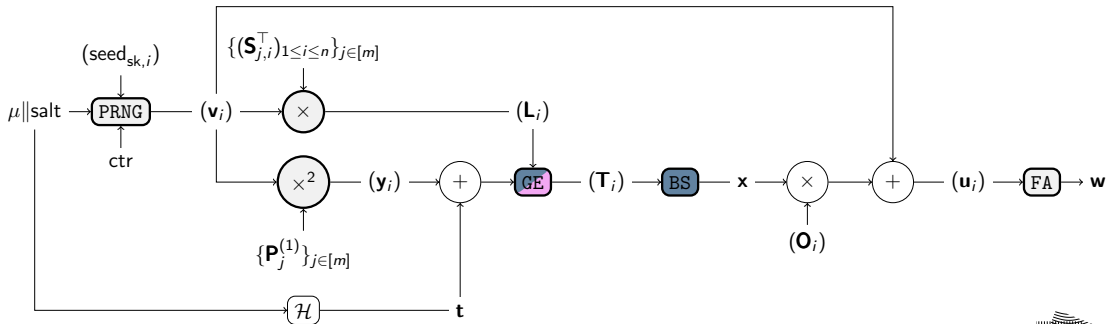
(Masked) UOV Signing

► Vector-Matrix Arithmetic (Boolean)



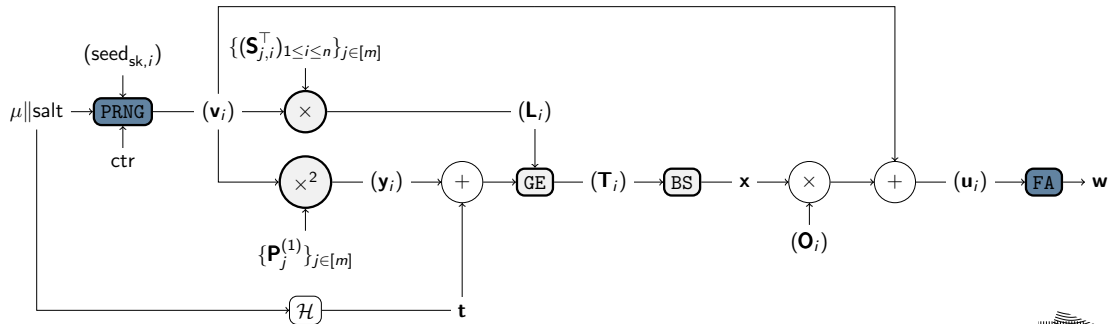
(Masked) UOV Signing

► Solving System of Linear Equations (Boolean & Multiplicative)



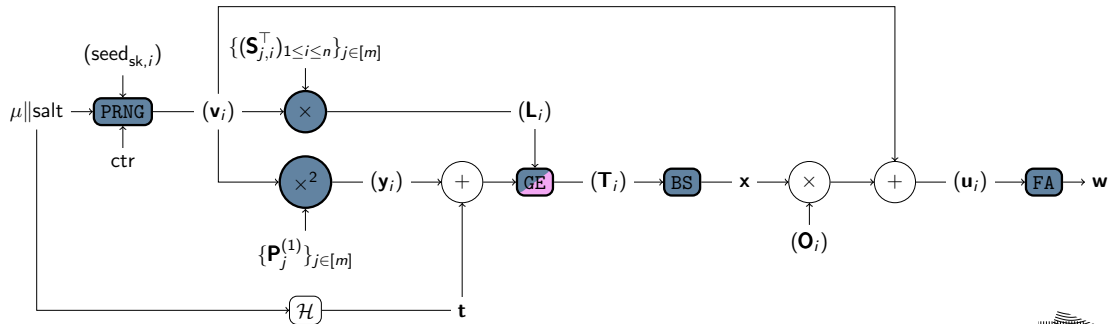
(Masked) UOV Signing

- Other: PRNG, linear operations (Boolean)



(Masked) UOV Signing

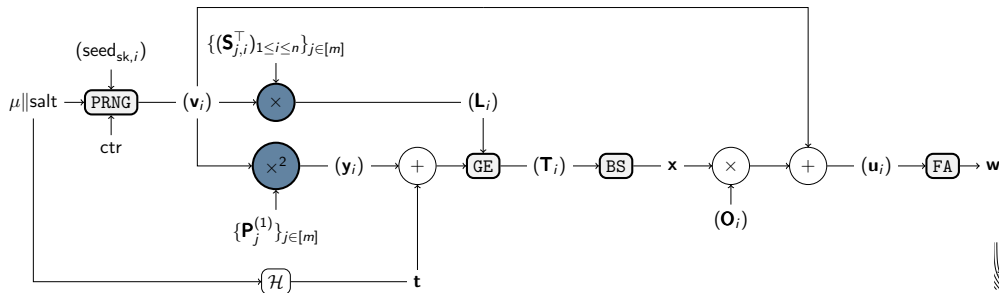
► Mainly Boolean Masking



Masked UOV Signing

(Masked) Matrix-Vector Multiplication & Quadratic Form Evaluation

- **Repeated** masked GF(256) multiplication
- *Lazy Compression* → reduce fresh randomness



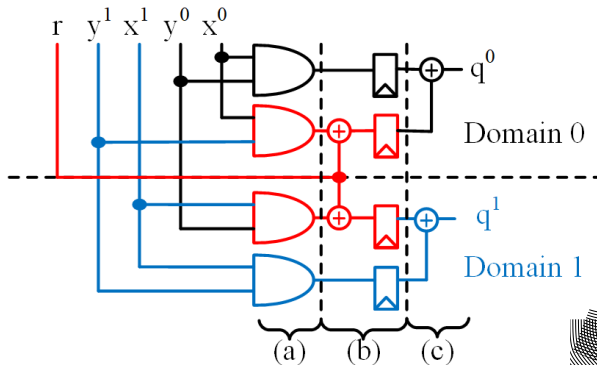
(Repeated) Masked Multiplication

Masked Multiplication = Calculation (a) + Compression (b, c) [GMK16]

NAÏVE

k multiplications:

- (a) k cross-products
- (b) k fresh randoms & registers
- (c) k ADD's



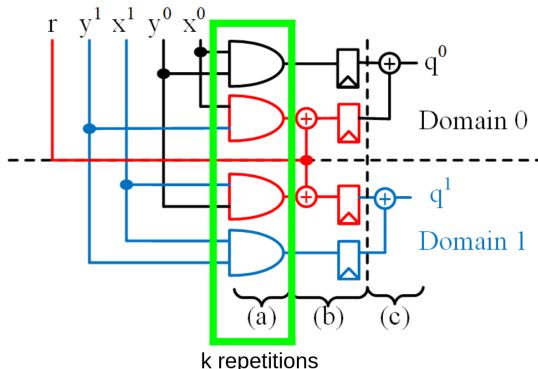
Lazy Compression

$$k \text{ Masked Multiplications} = k \times (a) + 1 \times (b, c)$$

DELAYED COMPRESSION

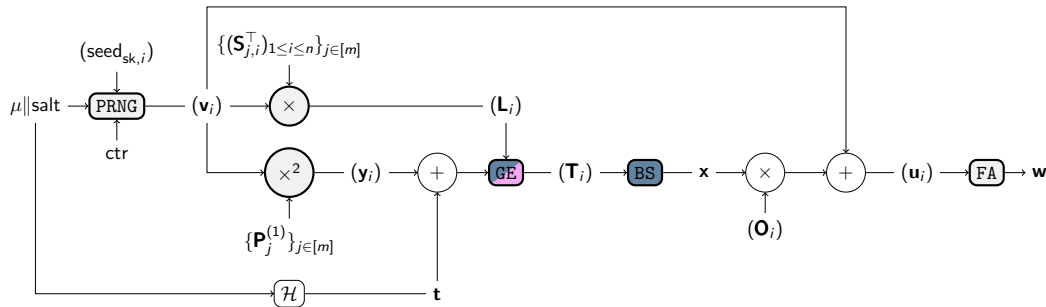
k multiplications:

- (a) k cross-products
- (b) 1 fresh randoms & register
- (c) $2k$ ADD's



Masked Gaussian Elimination

Row-Echelon Conversion w/ Back Substitution



Masked Gaussian Elimination

Row-Echelon Conversion w/ Back Substitution

Steps:

- (1) Make pivot $\neq 0 \rightarrow$ timing side-channel!
- (2) Make pivot = 1 $\rightarrow$ multiply with inverse (B2M)
- (3) Make elements below pivot = 0 $\rightarrow$ multiply & subtract

Step 3 is bottleneck ($\pm 80\%$)



Security Evaluation

Formal

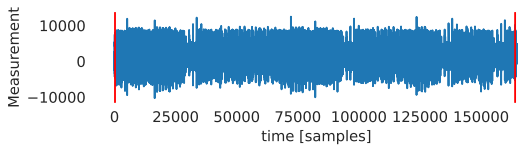
- ▶ Proofs in t -probing model, mechanically verified (`maskVerif`)

Practical/Experimental

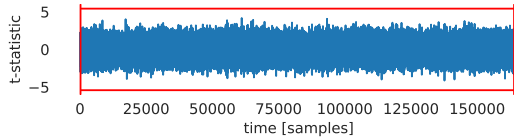
- ▶ SCA Evaluations Lab: NewAE CW308 UFO w/ STM32F415RG Arm Cortex-M4 + PicoScope (@ 125 MS/s)
- ▶ TVLA fix vs. random t -test (up to **1M** traces)



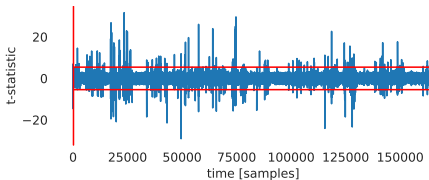
Security Evaluation



(a) Mean Trace



(b) RNG ON



(c) RNG OFF

Micro-Architectural Effects & Compiler Optimization

Transitional (Hamming Weight) Leakages in CPUs

- ▶ Register overwrites
- ▶ Sequential datapath/ALU access
- ▶ ...

Need to take care off these to ensure **countermeasure is effective!**



Micro-Architectural Effects & Compiler Optimization

Transitional (Hamming Weight) Leakages in CPUs

- ▶ Register overwrites
- ▶ Sequential datapath/ALU access
- ▶ ...

Need to take care off these to ensure **countermeasure is effective!**

To make matters worse:

- ▶ manually mitigate leakages (ASM, RND ...)
- ▶ compiler might optimize away [BGG⁺14, ZM24]

Possible solution: **disable compiler optimization** (-o0)



Proposed Solution

Enable **aggressive** compiler optimization (`-O3`)

A combination of:

- ▶ Masking countermeasure (written in C)
- ▶ interleaved ASM routines (clear ALU, pre-load RND ...)
- ▶ `-fno-schedule-insns` and `-fno-schedule-insns2` : **no re-ordering**

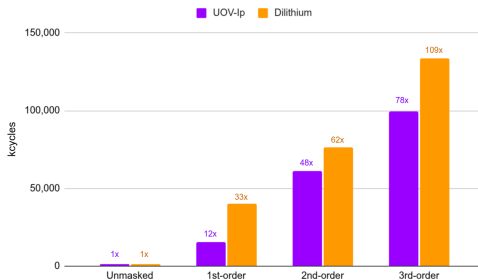
From our experiments, this is sufficient for practical security w low impact on performance.



Summary

Masking UOV¹⁰ is relatively *easy* and *efficient*:

- ▶ GF(2^8) & matrix-vector arithmetic
- ▶ **NO** A2Bs or B2As



¹⁰<https://github.com/KULeuven-COSIC/mUOV>



Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC
- ④ MPC-in-the-Head-based PQC**
- ⑤ Outlook: AI and (PQ) Cryptography



NIST Candidates

FAEST, MQOM, and SDitH.

- ▶ Newer constructions (TCitH and VOLEitH), security proofs are not verified (yet).
- ▶ Improved signing and verification speeds, smaller signatures.
- ▶ Most confident in security of FAEST (symmetric primitives), performance of MQOM, and hard-problem of SDitH.



NIST Candidates

FAEST, MQOM, and SDitH.

- ▶ Newer constructions (TCitH and VOLEitH), security proofs are not verified (yet).
 - ▶ Improved signing and verification speeds, smaller signatures.
 - ▶ Most confident in security of FAEST (symmetric primitives), performance of MQOM, and hard-problem of SDitH.
-
- ▶ Open questions: protection against physical attacks and performance on hardware platforms.



MPC-in-the-Head

Use *secret sharing* to prove knowledge of a secret key, without revealing any additional information.

- ▶ Simulate MPC evaluation *locally*
- ▶ Repeat τ times (soundness)
- ▶ Derive party shares using GGM trees



MPC-in-the-Head

Use *secret sharing* to prove knowledge of a secret key, without revealing any additional information.

- ▶ Simulate MPC evaluation *locally*
- ▶ Repeat τ times (soundness)
- ▶ Derive party shares using GGM trees

Mirath: One-GGM-tree optimization

MQOM: Correlated GGM trees



Symmetric Primitives in MPCitH

All PQC, including MPCitH, heavily relies on symmetric primitives

- ▶ Memory footprint: GGM trees/party shares
- ▶ Latency: tree-walking

Similar pattern for all MPCitH candidates!



HW Acceleration

Leverage **HW to accelerate MPCitH**¹¹, with minimal memory footprint

- ▶ On-the-fly GGM tree addressing and leaf derivation ...
- ▶ ...and MPC computations

¹¹<https://github.com/KULeuven-COSIC/MIRATH-in-HW>



HW Acceleration

Leverage **HW to accelerate MPCitH**¹¹, with minimal memory footprint

- ▶ On-the-fly GGM tree addressing and leaf derivation ...
- ▶ ... and MPC computations

→ Level of success depends on parameter sets!

- ▶ MQOM: 15+ parameter sets

¹¹<https://github.com/KULeuven-COSIC/MIRATH-in-HW>



SCA and DPA Resistance

Does the MPC/secret sharing parts offer *implicit* SCA resistance?



SCA and DPA Resistance

Does the MPC/secret sharing parts offer *implicit* SCA resistance?

NO.

- ▶ Modeling attacks



SCA and DPA Resistance

Does the MPC/secret sharing parts offer *implicit* SCA resistance?

NO.

- ▶ Modeling attacks
- ▶ But schemes can be *tweaked* for DPA resistance at lower cost than masking.
- ▶ Active area, impact on standards unclear.



Summary

The overheads typically associated with MPCitH-based PQC schemes can be largely avoided in HW:

- ▶ High-performance SHA-3/AES engines + MPC arithmetic
- ▶ Flexible GGM tree walking w/ small buffers



Outline

- ① Post-Quantum Cryptography Standardization
- ② Lattice-based PQC
- ③ Multivariate-based PQC
- ④ MPC-in-the-Head-based PQC
- ⑤ Outlook: AI and (PQ) Cryptography



The Summer of AI and Cryptography

In the Summer of 2026, the capabilities of AI for cryptanalysis became clear, with usage now exploding.

Examples:

- ▶ HAWK attack
- ▶ Classic McEliece attacks
- ▶ ICCS PQC candidate evaluation



Example: HAWK

HAWK- n Key Recovery Reduces to SVP in Dimension $n/2+1$ (Z. Straznickas and S. A. Weis, Anthropic)

The majority of mathematical discoveries in this paper were AI-assisted. Human author contribution mainly consisted of directing, organizing and verifying AI work.



Example: HAWK

HAWK- n Key Recovery Reduces to SVP in Dimension $n/2+1$ (Z. Straznickas and S. A. Weis, Anthropic)

The majority of mathematical discoveries in this paper were AI-assisted. Human author contribution mainly consisted of directing, organizing and verifying AI work.

- Impact: need to double parameters, high performance cost.



Example: HAWK

HAWK- n Key Recovery Reduces to SVP in Dimension $n/2+1$ (Z. Straznickas and S. A. Weis, Anthropic)

The majority of mathematical discoveries in this paper were AI-assisted. Human author contribution mainly consisted of directing, organizing and verifying AI work.

- ▶ Impact: need to double parameters, high performance cost.
- ▶ First high-profile *break* of PQC scheme by AI.



Example: HAWK

HAWK- n Key Recovery Reduces to SVP in Dimension $n/2+1$ (Z. Straznickas and S. A. Weis, Anthropic)

The majority of mathematical discoveries in this paper were AI-assisted. Human author contribution mainly consisted of directing, organizing and verifying AI work.

- ▶ Impact: need to double parameters, high performance cost.
- ▶ First high-profile *break* of PQC scheme by AI.
- ▶ Demonstrates that latest AI models are good at extending/closing gaps in human-developed attack techniques.



Example: Classic McEliece

Several recent works/improvements acknowledge impact of AI:

- ▶ *HOVER: Higher-Order Vanishing Endomorphism Recovery* (M.-J. O. Saarinen)
- ▶ *Bit Operation Cost of “Holdout” Key-Recovery Attacks Against Classic McEliece* (M.-J. O. Saarinen)
- ▶ *Improving GJS Key Recovery for Classic McEliece* (S. A. Weis)



Example: Classic McEliece

Several recent works/improvements acknowledge impact of AI:

- ▶ *HOVER: Higher-Order Vanishing Endomorphism Recovery* (M.-J. O. Saarinen)
- ▶ *Bit Operation Cost of “Holdout” Key-Recovery Attacks Against Classic McEliece* (M.-J. O. Saarinen)
- ▶ *Improving GIJS Key Recovery for Classic McEliece* (S. A. Weis)

- ▶ All use SotA LLMs for: theory formulation, proofs, analysis, preparing and executing experiments, and more.



Example: Classic McEliece

Several recent works/improvements acknowledge impact of AI:

- ▶ *HOVER: Higher-Order Vanishing Endomorphism Recovery* (M.-J. O. Saarinen)
 - ▶ *Bit Operation Cost of “Holdout” Key-Recovery Attacks Against Classic McEliece* (M.-J. O. Saarinen)
 - ▶ *Improving GJS Key Recovery for Classic McEliece* (S. A. Weis)
-
- ▶ All use SotA LLMs for: theory formulation, proofs, analysis, preparing and executing experiments, and more.
 - ▶ **Impact:** BSI: *Classic McEliece should currently no longer be used in new developments or when planning new cryptographic applications.*



Example: ICCS PQC Standardization

Chinese PQC standardization competition just launched: 84 fresh targets. Use AI for analysis and evaluation?



Example: ICCS PQC Standardization

Chinese PQC standardization competition just launched: 84 fresh targets. Use AI for analysis and evaluation?

- ▶ More eyes → more feedback → better judgement



Example: ICCS PQC Standardization

Chinese PQC standardization competition just launched: 84 fresh targets. Use AI for analysis and evaluation?

- ▶ More eyes → more feedback → better judgement
- ▶ !! 191 findings in 1 week (incl. hash competition) !!



Example: ICCS PQC Standardization

Chinese PQC standardization competition just launched: 84 fresh targets. Use AI for analysis and evaluation?

- ▶ More eyes → more feedback → better judgement
- ▶ !! 191 findings in 1 week (incl. hash competition) !!
- ▶ Reports are generated faster than they can be digested.



Example: ICCS PQC Standardization

Chinese PQC standardization competition just launched: 84 fresh targets. Use AI for analysis and evaluation?

- ▶ More eyes → more feedback → better judgement
- ▶ !! 191 findings in 1 week (incl. hash competition) !!
- ▶ Reports are generated faster than they can be digested.

Quality of (AI-generated) reports has been mixed...

- ▶ LLM dumps do not provide insight
- ▶ `ngcc.dev` is great for keeping track



Impact of AI on PQC

It is clear SotA LLMs are very useful for rapid iteration and extension of known techniques.



Impact of AI on PQC

It is clear SotA LLMs are very useful for rapid iteration and extension of known techniques.

- ▶ Most headlines related to *destructive* examples.



Impact of AI on PQC

It is clear SotA LLMs are very useful for rapid iteration and extension of known techniques.

- ▶ Most headlines related to *destructive* examples.
- ▶ Way fewer stories about using it *constructively*.



Impact of AI on PQC

It is clear SotA LLMs are very useful for rapid iteration and extension of known techniques.

- ▶ Most headlines related to *destructive* examples.
- ▶ Way fewer stories about using it *constructively*.

I am interested to read about those! What works? What doesn't?

- ▶ Spec-to-code translation
- ▶ Optimized implementations
- ▶ Catching (trivial) bugs
- ▶ Countermeasures (?)



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:

- ▶ Mostly scattered solutions.



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:

- ▶ Mostly scattered solutions.
- ▶ Analysis of cost scaling for different families.



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:

- ▶ Mostly scattered solutions.
- ▶ Analysis of cost scaling for different families.
- ▶ Formal, machine-checked verification.



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:

- ▶ Mostly scattered solutions.
- ▶ Analysis of cost scaling for different families.
- ▶ Formal, machine-checked verification.
- ▶ Optimizations for different target platforms, costs, hybrid countermeasures, . . .



Summary

PQC deployment and adoption should be done with **care and urgency**.

Several open challenges related to **secure implementations**, for example:

- ▶ Mostly scattered solutions.
- ▶ Analysis of cost scaling for different families.
- ▶ Formal, machine-checked verification.
- ▶ Optimizations for different target platforms, costs, hybrid countermeasures, . . .
- ▶ Using AI in constructive fashion (?)



Q&A

Thanks! Questions?





Gorjan Alagic, Daniel Apon, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone.
Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process.
NIST Internal Report (IR) 8413-upd1, National Institute of Standards and Technology, Gaithersburg, MD, 2022.



Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone.
Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process.
NIST Internal Report (IR) 8240, National Institute of Standards and Technology, Gaithersburg, MD, 2019.



Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone.
Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process.
NIST Internal Report (IR) 8309, National Institute of Standards and Technology, Gaithersburg, MD, 2020.



Gorjan Alagic, Maxime Bros, Pierre Ciadoux, David Cooper, Quynh Dang, Thinh Hung Dang, John Kelsy, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller.
Status Report on the First Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process.
NIST Internal Report (IR) 8528, National Institute of Standards and Technology, Gaithersburg, MD, 2021.





Gorjan Alagic, Maxime Bros, Pierre Ciadoux, David Cooper, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller.

Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process.
NIST Internal Report (IR) 8545, National Institute of Standards and Technology, Gaithersburg, MD, 2025.



Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Hung Dang, John Kelsy, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller.

Status Report on the Second Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process.

NIST Internal Report (IR) 8610, National Institute of Standards and Technology, Gaithersburg, MD, 2026.



Gilles Barthe, Sonia Belaïd, Thomas Espitau, Pierre-Alain Fouque, Benjamin Grégoire, Mélissa Rossi, and Mehdi Tibouchi.

Masking the GLP lattice-based signature scheme at any order.

In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 354–384. Springer, Cham, April / May 2018.





Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang.

UOV — Unbalanced Oil and Vinegar.

Technical report, National Institute of Standards and Technology, 2024.

available at <https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures>.



Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert.

On the cost of lazy engineering for masked software implementations.

In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 64–81. Springer, 2014.



Luke Beckwith, Duc Tri Nguyen, and Kris Gaj.

High-performance hardware implementation of crystals-dilithium.

In *2021 International Conference on Field-Programmable Technology (ICFPT)*, pages 1–10, 2021.



Gaëtan Cassiers.

Composable and efficient masking schemes for side-channel secure implementations.

PhD thesis, École polytechnique de Louvain and Université catholique de Louvain, 2022.





Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone.
Report on Post-Quantum Cryptography.
NIST Internal Report (IR) 8105, National Institute of Standards and Technology, Gaithersburg, MD, 2016.



Hannes Gross, Stefan Mangard, and Thomas Korak.
Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order.
In *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security*, TIS '16, page 3, New York, NY, USA, 2016. Association for Computing Machinery.



National Institute of Standards and Technology.
Module-Lattice-Based Digital Signature Standard.
Federal Information Processing Standards Publication (FIPS) 204, Department of Commerce, Washington, DC, 2024.



National Institute of Standards and Technology.
Module-Lattice-Based Key-Encapsulation Mechanism.
Federal Information Processing Standards Publication (FIPS) 203, Department of Commerce, Washington, DC, 2024.



National Institute of Standards and Technology.
Stateless Hash-Based Digital Signature Standard.
Federal Information Processing Standards Publication (FIPS) 205, Department of Commerce, Washington, DC, 2024.





Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang.
FALCON.

Technical report, National Institute of Standards and Technology, 2022.
available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.



Lars Ran.

Wedges, Oil, and Vinegar.

In Joan Daemen and Emmanuel Thomé, editors, *Advances in Cryptology – EUROCRYPT 2026*, pages 363–388, Cham, 2026. Springer Nature Switzerland.



Jannik Zeitschner and Amir Moradi.

Pommes: Prevention of micro-architectural leakages in masked embedded software.

IACR Transactions on Cryptographic Hardware and Embedded Systems, 2024(3):342–376, Jul. 2024.

