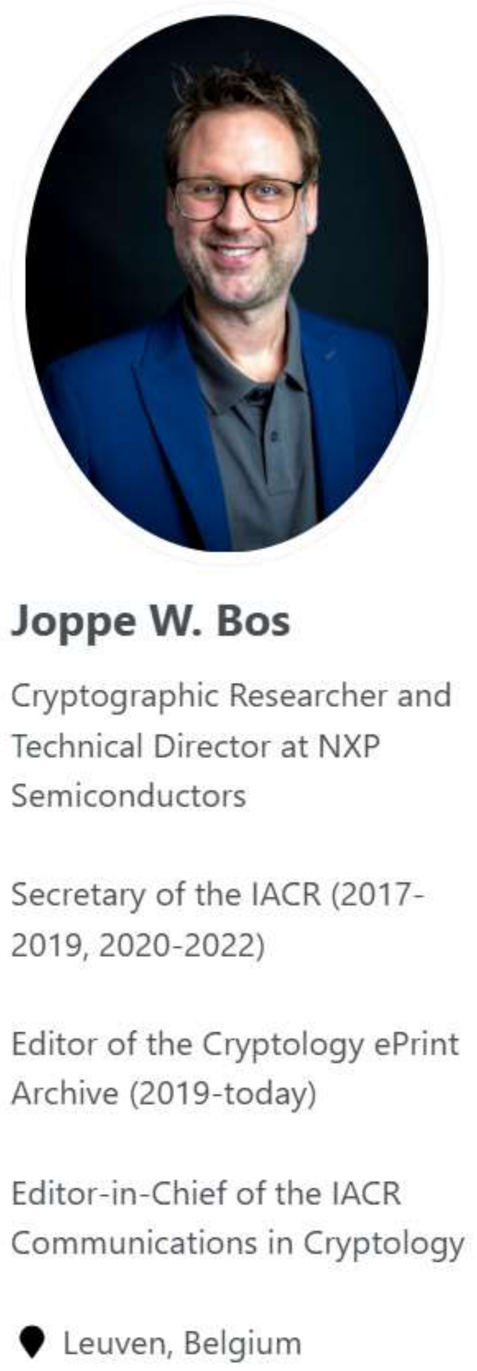
Thanks to co-pilot for the pictures in this readout

Embedded Security: Challenges and Opportunities when Migrating to Post-Quantum Cryptography

Joppe Bos

June 2025

Chania, Crete, Greece



Joppe W. Bos

Cryptographic Researcher and Technical Director at NXP Semiconductors

Secretary of the IACR (2017-2019, 2020-2022)

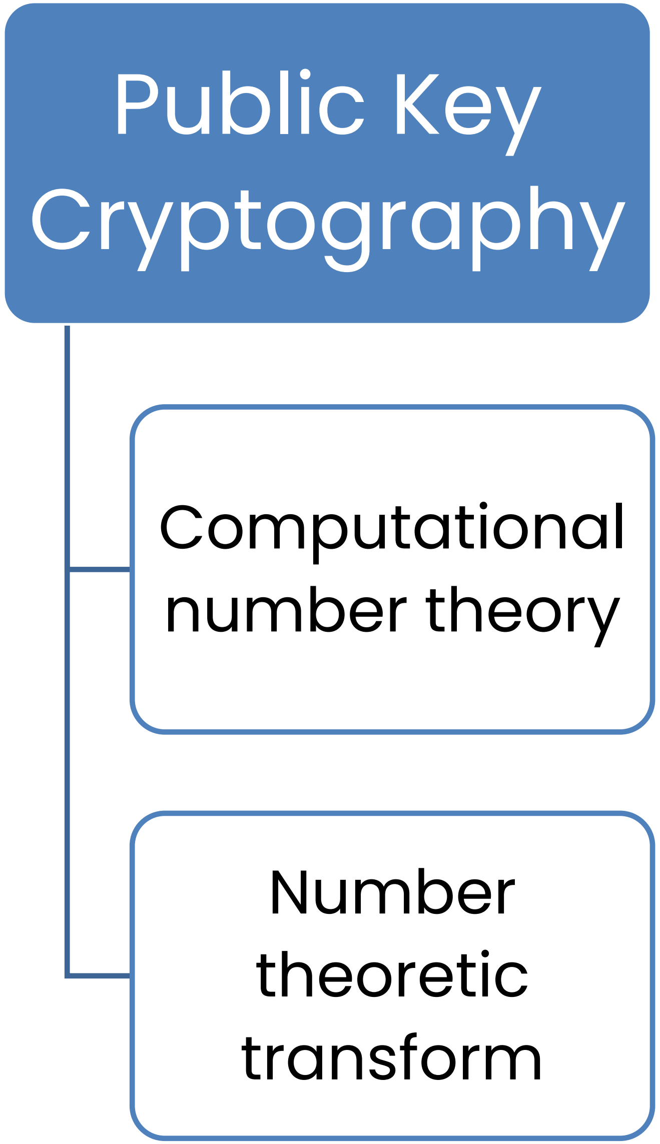
Editor of the Cryptology ePrint Archive (2019-today)

Editor-in-Chief of the IACR Communications in Cryptology

Leuven, Belgium

WHOAMI

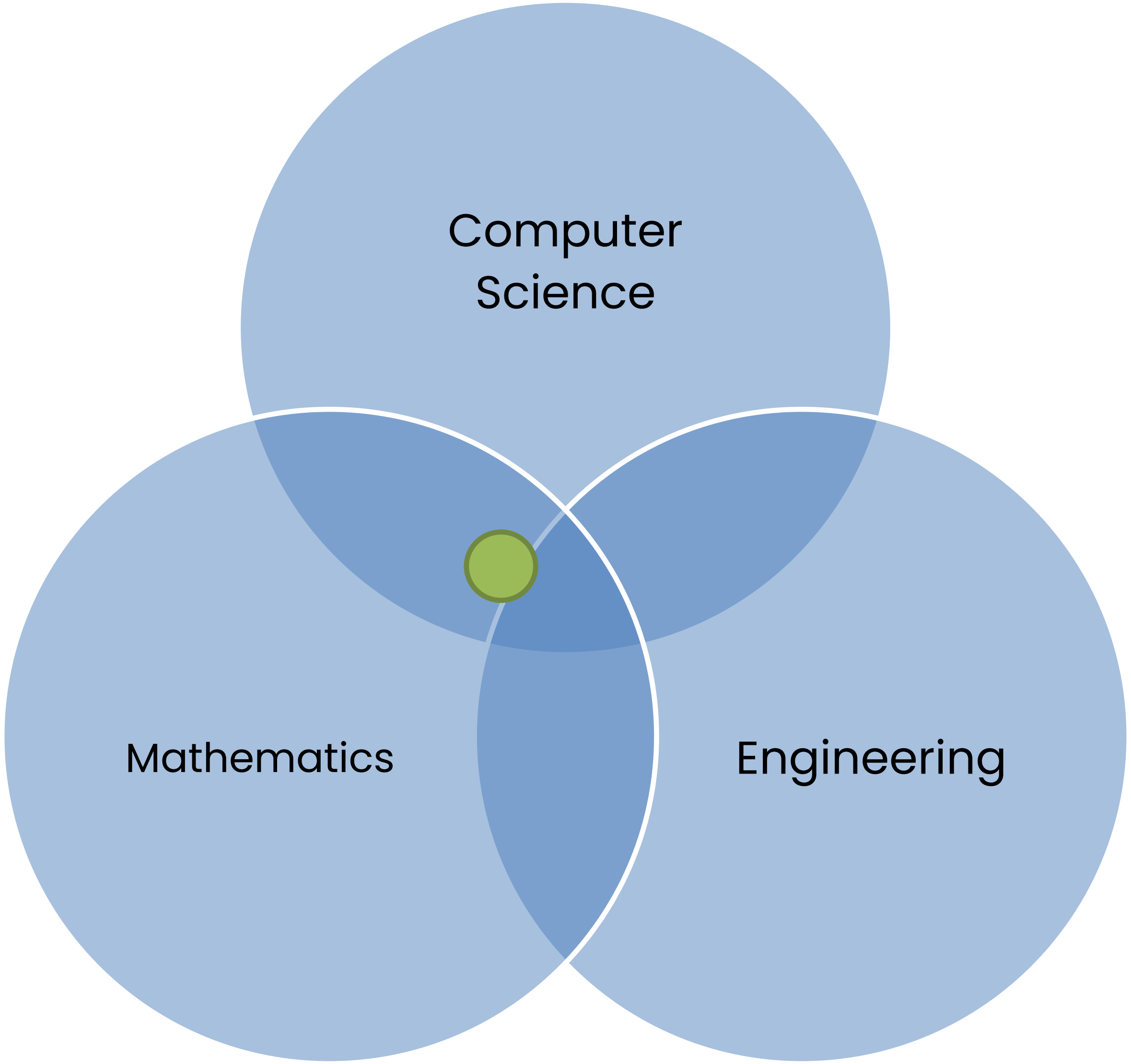
- Cryptographic researcher + Technical Director
  - Competence center crypto & security at NXP Semiconductors, Leuven
  - Lead the PQC team
  - Lead security + crypto funded projects & university relations
- Post-doc
  - Cryptography Research Group at Microsoft Research, Redmond, USA.
- PhD in Cryptology
  - EPFL, Lausanne, Switzerland
- Bachelor / Master in Computer Science
  - University of Amsterdam



Public Key Cryptography

Computational number theory

Number theoretic transform



Computer Science

Mathematics

Engineering

Breaking ECC

112-bit ECDLP solved using 224 PlayStation 3 game consoles.

NXP Corporate Overview

Together we accelerate the breakthroughs that advance our world

We design purpose-built, rigorously tested technologies that enable devices to sense, think, connect and act intelligently to improve people's daily lives.

Automotive

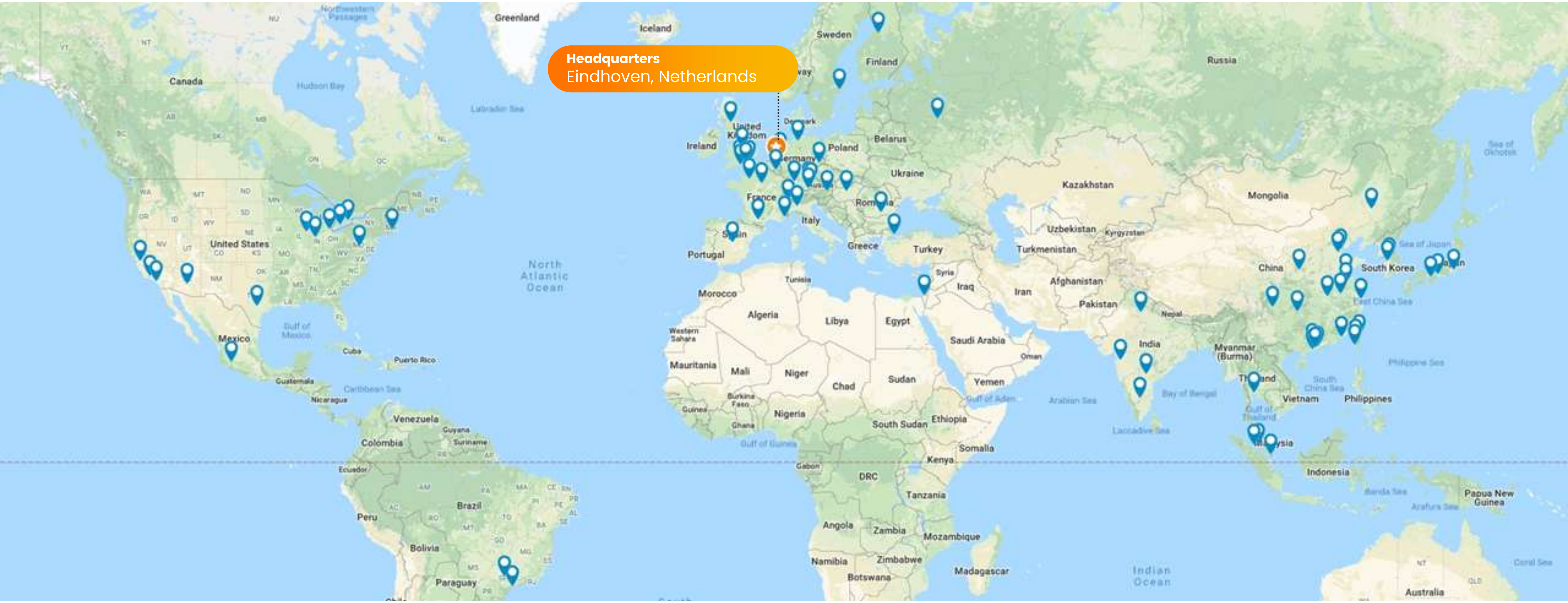
Industrial & IoT

Mobile

Communication Infrastructure

NXP locations

~34,200 team members with operations in more than **30 countries**



Automotive

market positions

Automotive

Technology Leadership

+

Applications Leadership

#1 Auto processors

#1 Auto applications processors

#1 Auto RF

#1 Auto DSPs

#1 Cross-domain processors

#1 Infotainment

#1 Car radio

#1 Secure car access

#1 In-vehicle networking

Sources: Strategy Analytics: Automotive Semiconductors Vendor Market Shares, April 2024, Strategy Analytics: Infotainment and Telematics Semiconductors Vendor Market Shares, April 2024, Gartner: Semiconductors Market Shares, April 2024, S&P: competitive landscaping tool, April 2024, IHS: automotive semiconductors market tracker, April 2024

Edge processing –
a distributed intelligence pyramid

Millions

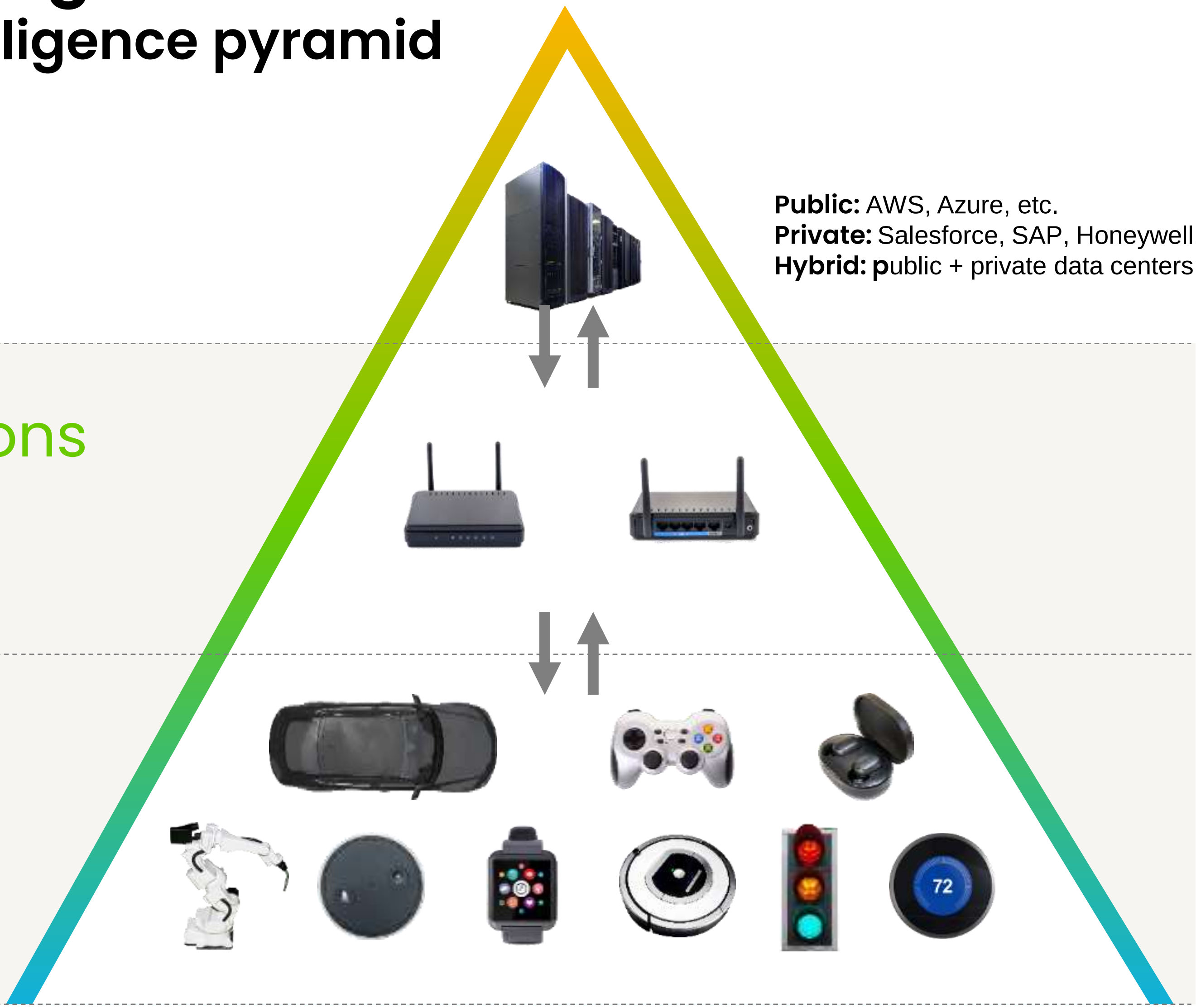
Cloud
Data centers

10's to 100's Millions

Network Edge
Network computing

Billions

Application Edge
IoT end points



Public: AWS, Azure, etc.
Private: Salesforce, SAP, Honeywell, etc.
Hybrid: public + private data centers

Edge processing
served market

End-to-end solutions for Matter

A unified IP-based protocol to securely and robustly connect smart devices with each other, regardless of brand, and across smart home platforms

Bring interoperability in the Smart Home industry

Simplify development for “things”

Increase reliability for consumers

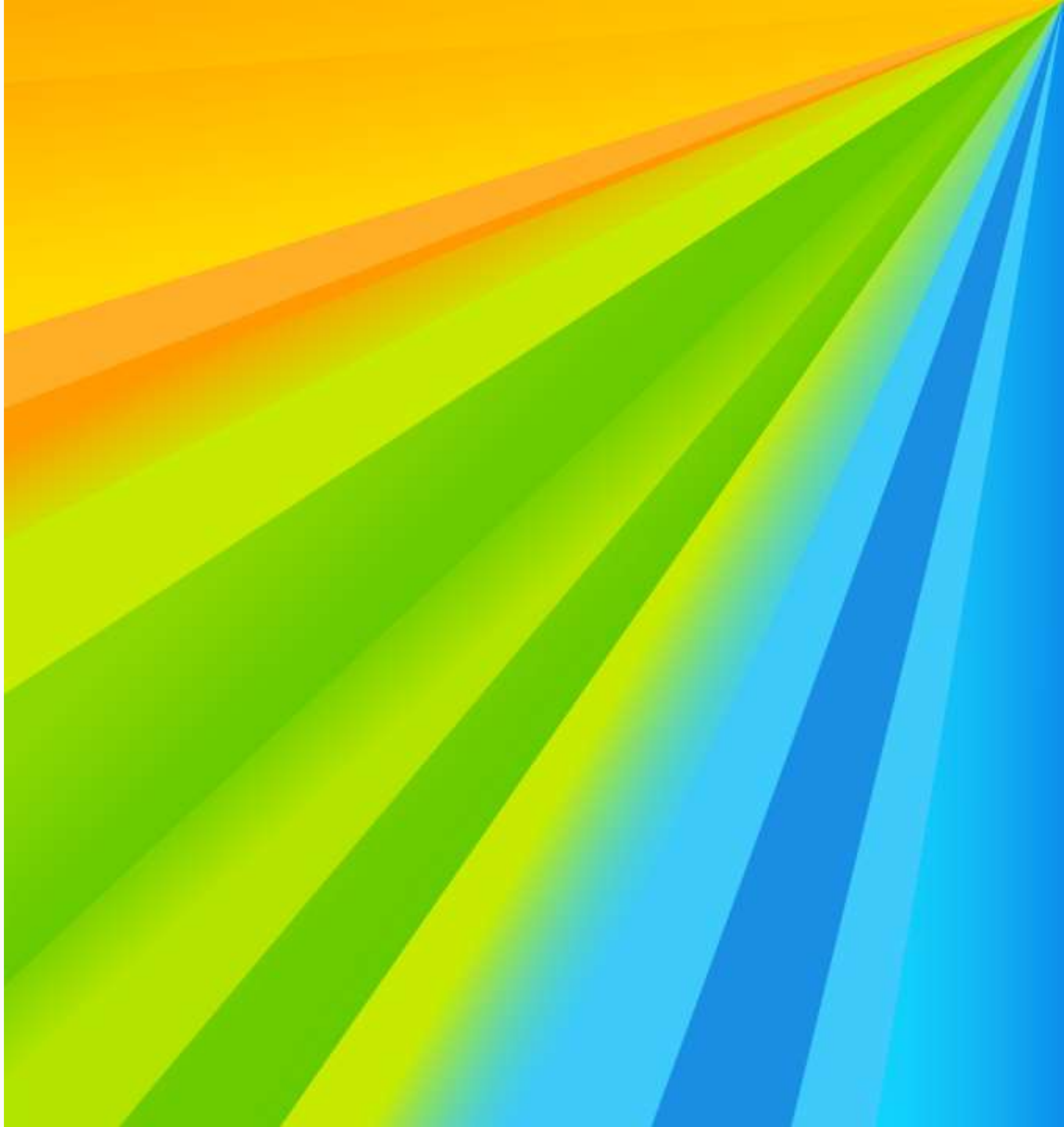
Ensure security and privacy

Led by global brands and 200+ companies

amazon ASSA ABLOY COMCAST Google HUAWEI IKEA infineon Kroger LEEDARSON legrand NXP

oppo resideo Schneider Electric Signify SILICON LABS SmartThings somfy ST Texas Instruments tuya Wulian csa

Classical Cryptography



Public-Key Cryptography

In **public-key** cryptography the theoretical foundation of the schemes used are problems which are **believed** to be hard

- Integer factorization problem (RSA)
- Discrete logarithm problem (DSA, ElGamal)

One of the main ingredients to these problems is a group

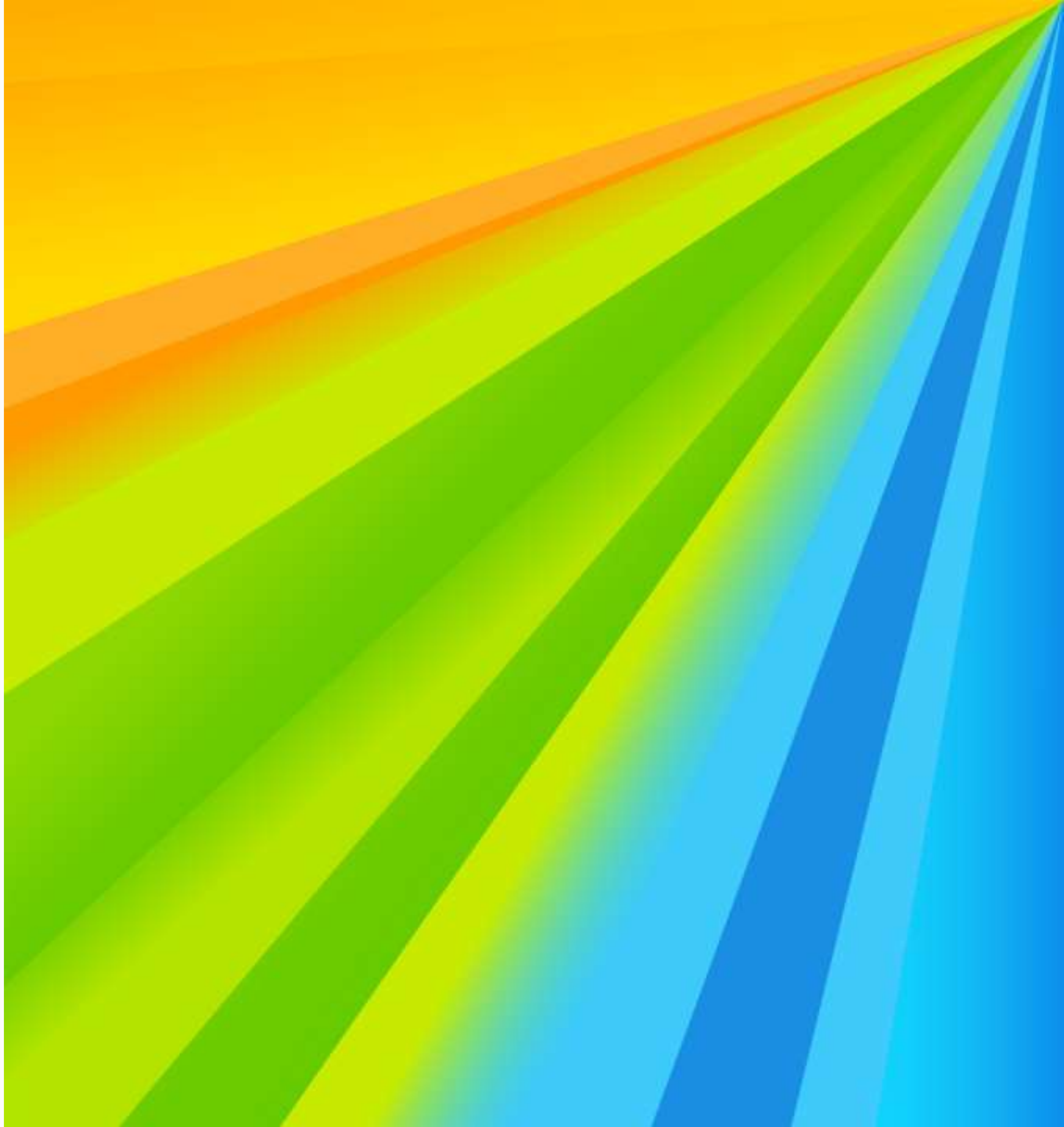
RSA → $(\mathbb{Z}/N\mathbb{Z})^{\times}$→ integers $[\,1, 2, \ldots, N-1]$ which are co-prime to N

DSA/ElGamal → $\mathbb{F}_p^{\times} = (\mathbb{Z}/p\mathbb{Z})^{\times}$ → integers $[\,1, 2, \ldots, p-1]$ where p is prime

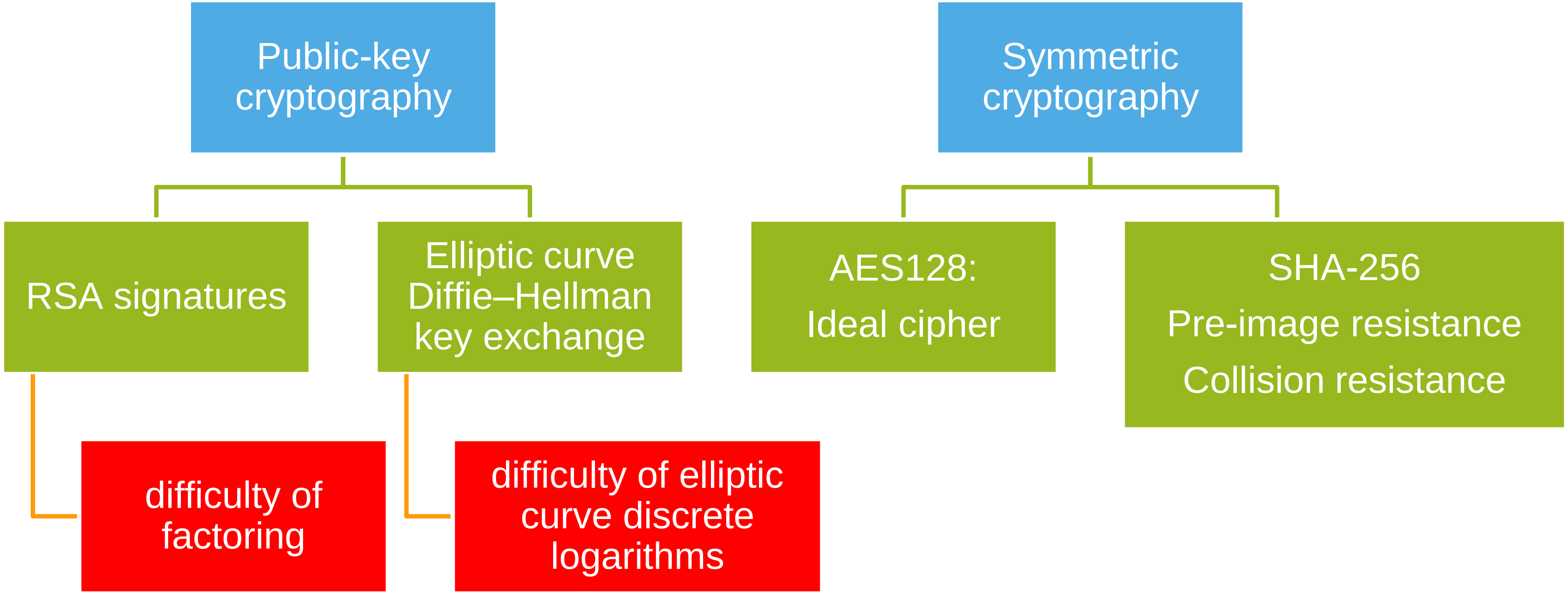
Elliptic Curve Cryptography → $E/\mathbb{F}_p$ → point on $E(\mathbb{F}_p)$ where p is prime

Application	Encryption Scheme, Signature Scheme, Identification Scheme, etc.		
Cryptosystem	DSA, ElGamal, Schnorr, etc.		RSA, Rabin, etc.
Computational Problem	The Discrete Logarithm Problem in a Group of prime Order		The Factoring Problem
Algebraic Structure	The multiplicative group of integers modulo a prime	Elliptic Curve Group over a Finite Field	The set of integers modulo the product of two primes

Post-Quantum Cryptography



Contemporary Cryptography
TLS-ECDHE-RSA-AES128-GCM-SHA256



Public-key cryptography

RSA signatures

difficulty of factoring

Elliptic curve Diffie–Hellman key exchange

difficulty of elliptic curve discrete logarithms

Symmetric cryptography

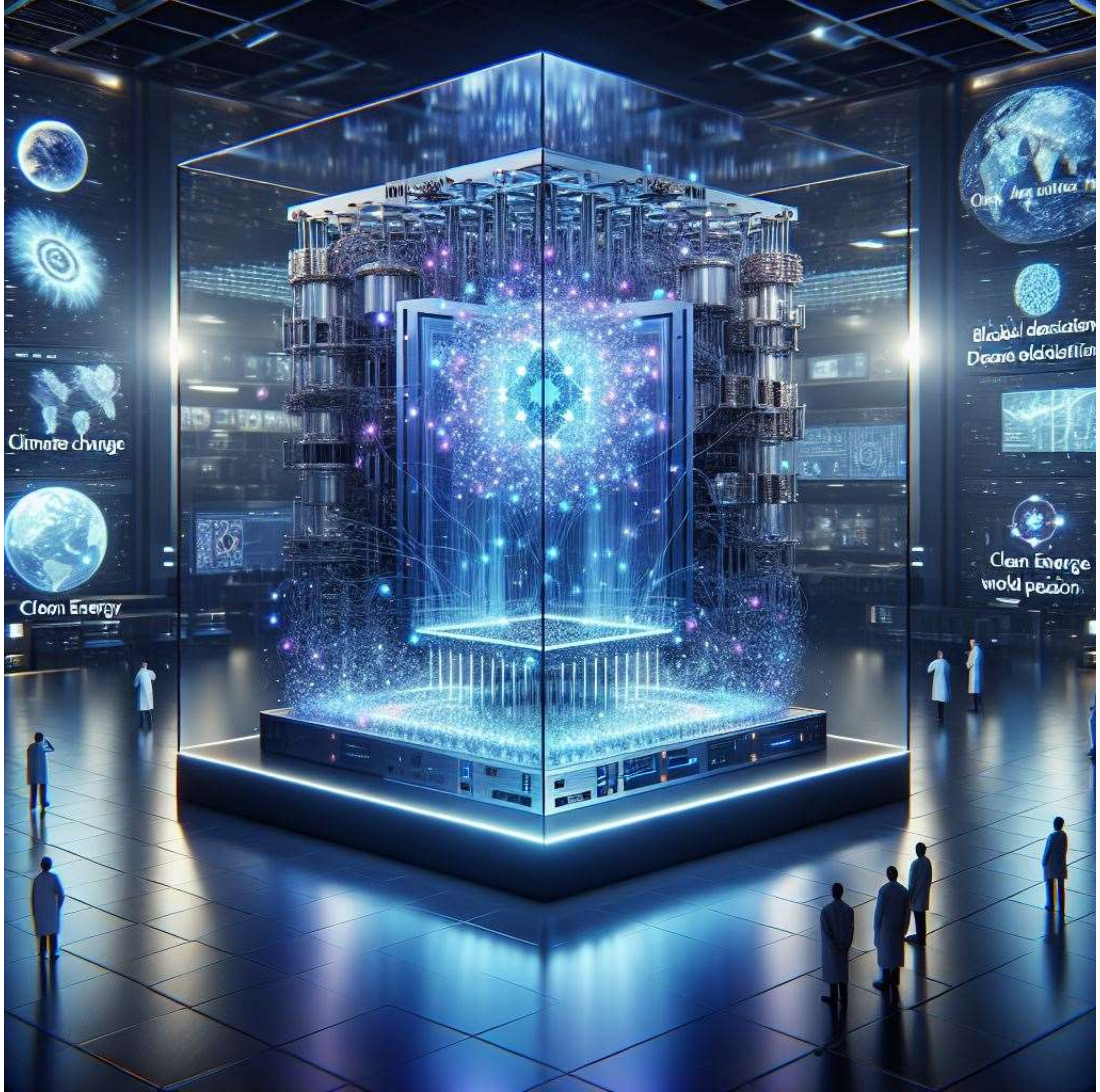
AES128:

Ideal cipher

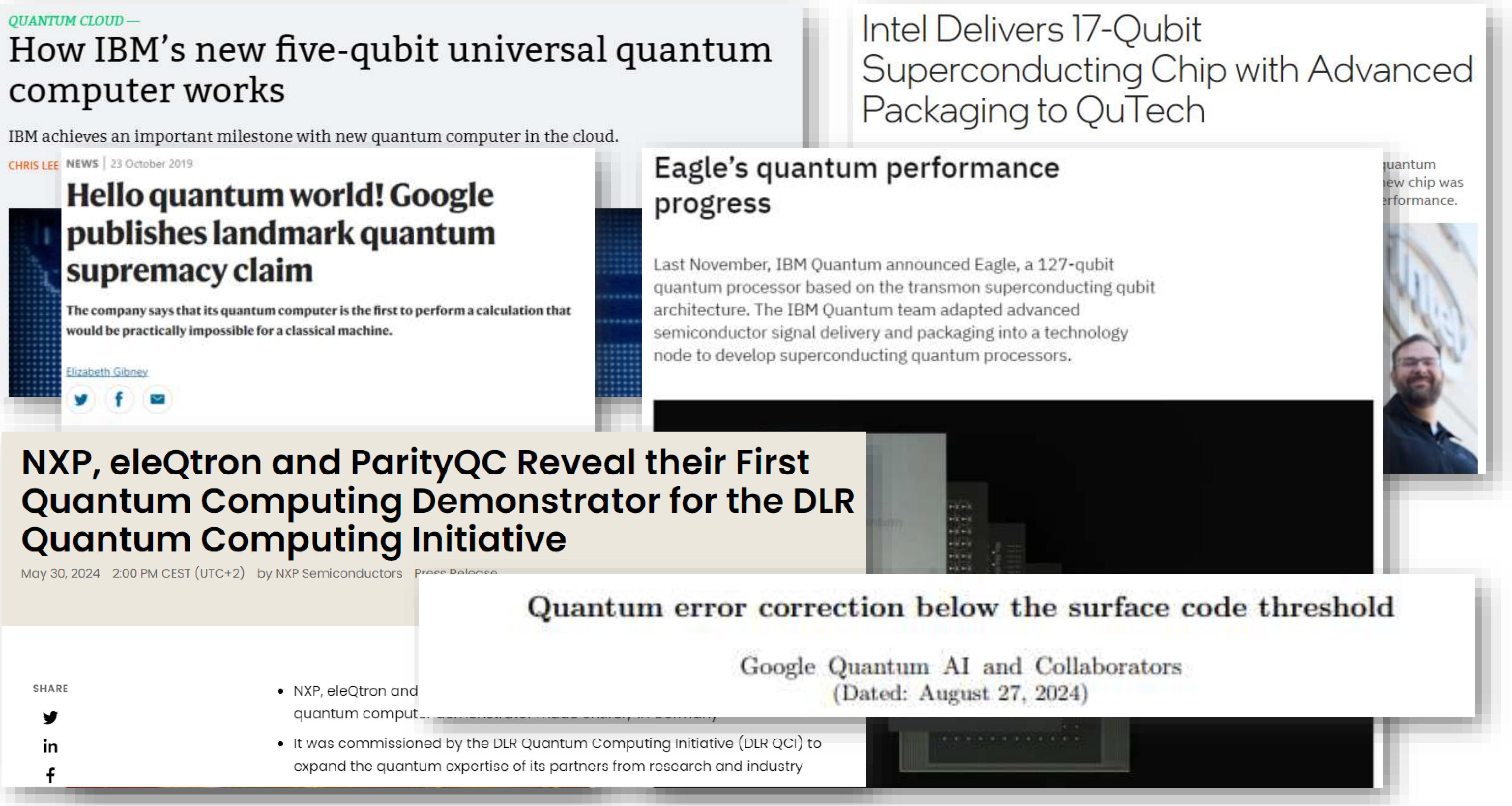
SHA-256

Pre-image resistance

Collision resistance



Climate change
Cloan Energy
Global dasdaisn
Deare oldaisiton
Clan Energe
wold peaon



QUANTUM CLOUD —

How IBM's new five-qubit universal quantum computer works

IBM achieves an important milestone with new quantum computer in the cloud.

CHRIS LEE

NEWS | 23 October 2019

Hello quantum world! Google publishes landmark quantum supremacy claim

The company says that its quantum computer is the first to perform a calculation that would be practically impossible for a classical machine.

Elizabeth Gibney

Intel Delivers 17-Qubit Superconducting Chip with Advanced Packaging to QuTech

uantum
ew chip was
rformance.

Eagle's quantum performance progress

Last November, IBM Quantum announced Eagle, a 127-qubit quantum processor based on the transmon superconducting qubit architecture. The IBM Quantum team adapted advanced semiconductor signal delivery and packaging into a technology node to develop superconducting quantum processors.

NXP, eleQtron and ParityQC Reveal their First Quantum Computing Demonstrator for the DLR Quantum Computing Initiative

May 30, 2024 2:00 PM CEST (UTC+2) by NXP Semiconductors Press Release

SHARE

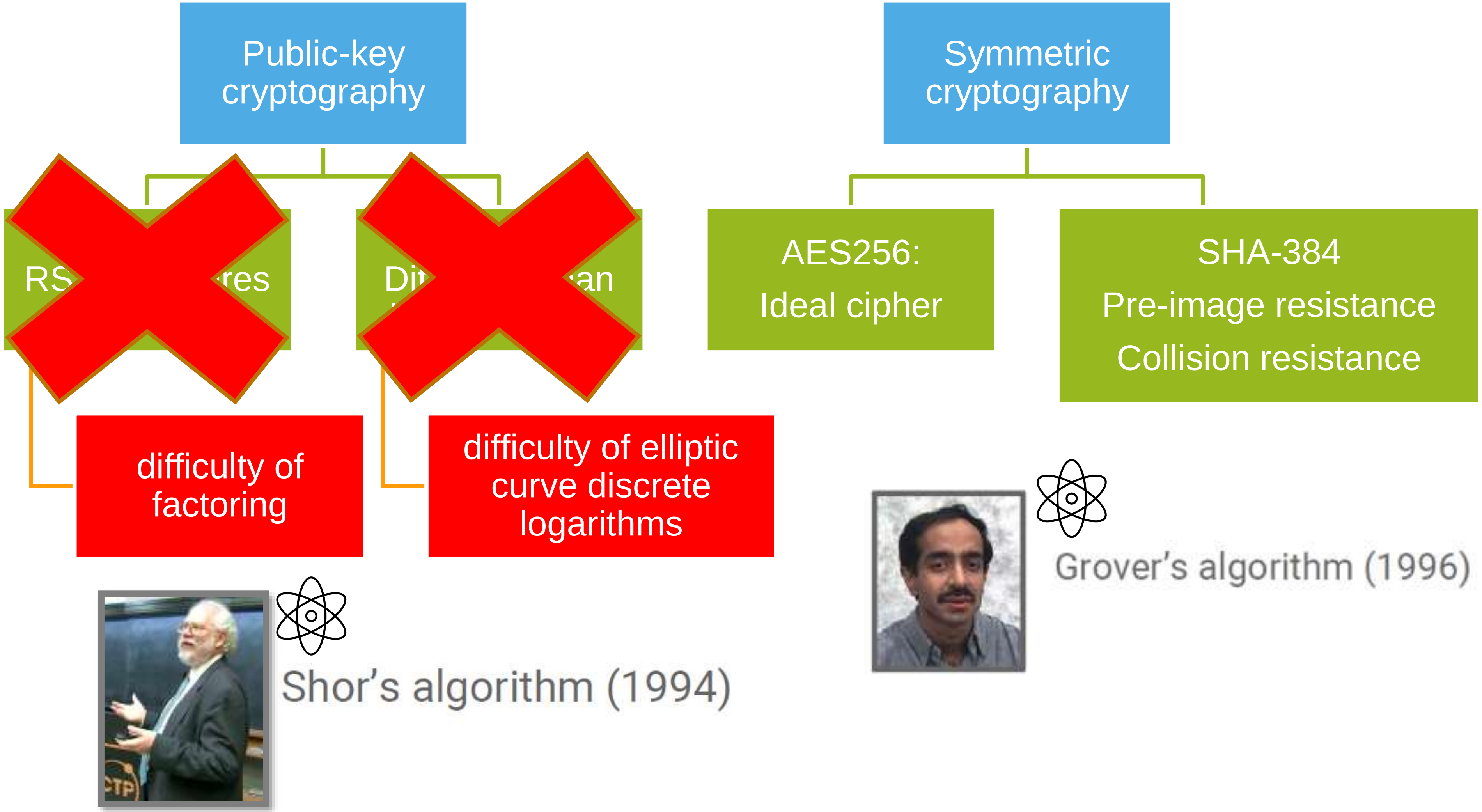
- NXP, eleQtron and
quantum comput
- It was commissioned by the DLR Quantum Computing Initiative (DLR QCI) to expand the quantum expertise of its partners from research and industry

Quantum error correction below the surface code threshold

Google Quantum AI and Collaborators
(Dated: August 27, 2024)

Contemporary cryptography
TLS-~~ECDHE-RSA~~-AES256-GCM-SHA384

“Double” the key sizes



Public-key cryptography

RS res

Dit an

difficulty of factoring

difficulty of elliptic curve discrete logarithms

Symmetric cryptography

AES256:
Ideal cipher

SHA-384
Pre-image resistance
Collision resistance

Grover’s algorithm (1996)

Shor’s algorithm (1994)

Quantum potential to destroy security as we know it

Confidential email messages, private documents, and financial transactions
Secure today but could be compromised in the future, even if encrypted

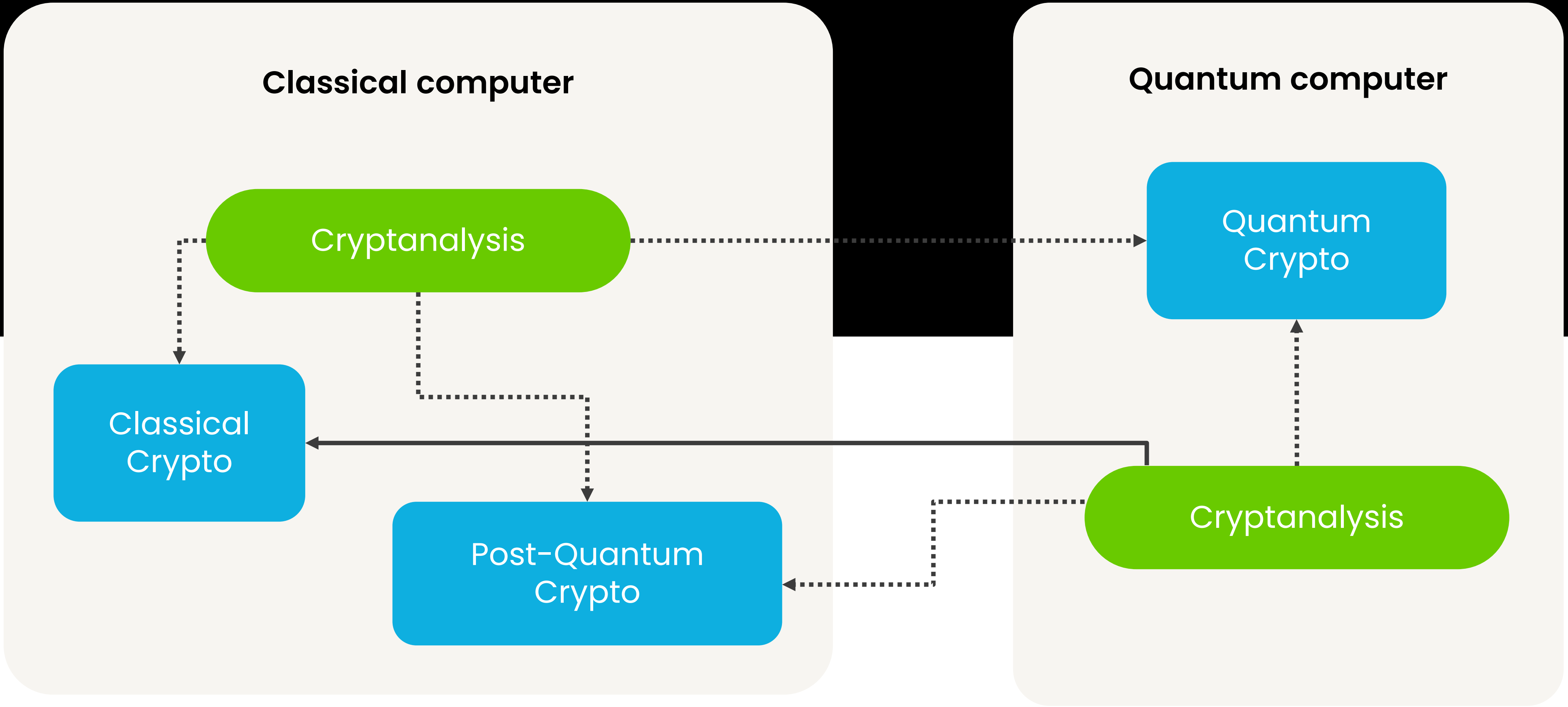
Firmware update mechanisms in vehicles
Could be circumvented and allow dangerous modifications

Critical industrial and public service infrastructure (for healthcare, utilities, and transportation using internet and virtual private networks)
Could become exposed – potentially destabilize cities

Audit trails and digitally signed documents associated with safety (auto certification and pharmaceutical authorizations)
Could be retrospectively modified

The integrity of blockchains
Could be retrospectively compromised -
could include fraudulent manipulation of ledger and cryptocurrency transactions

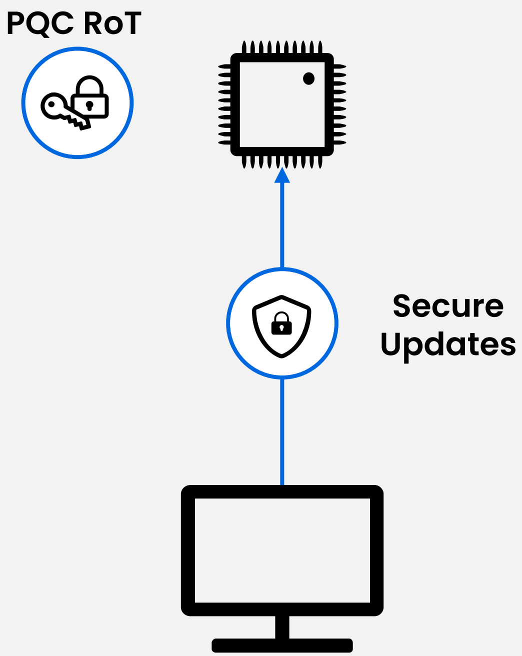
Post-quantum versus quantum crypto



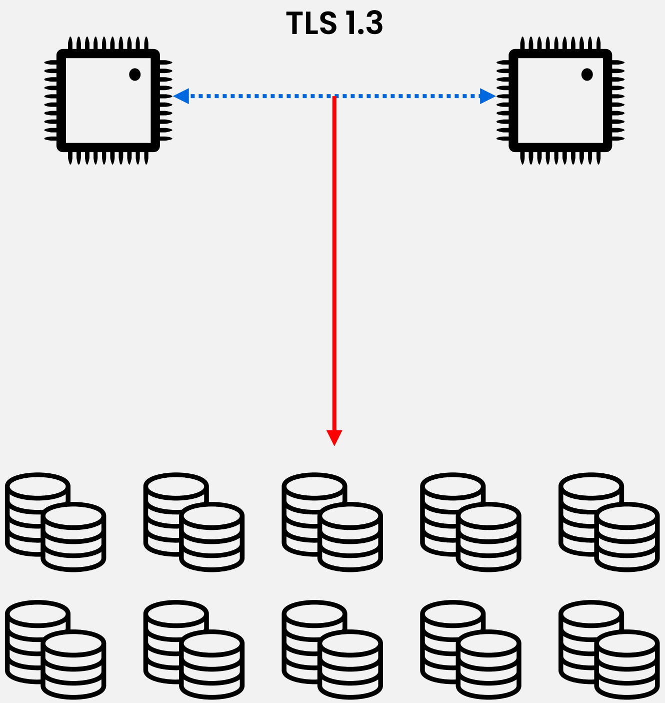
Is Post-Quantum Cryptography relevant for you?

Standards & Compliance

Crypto Agility

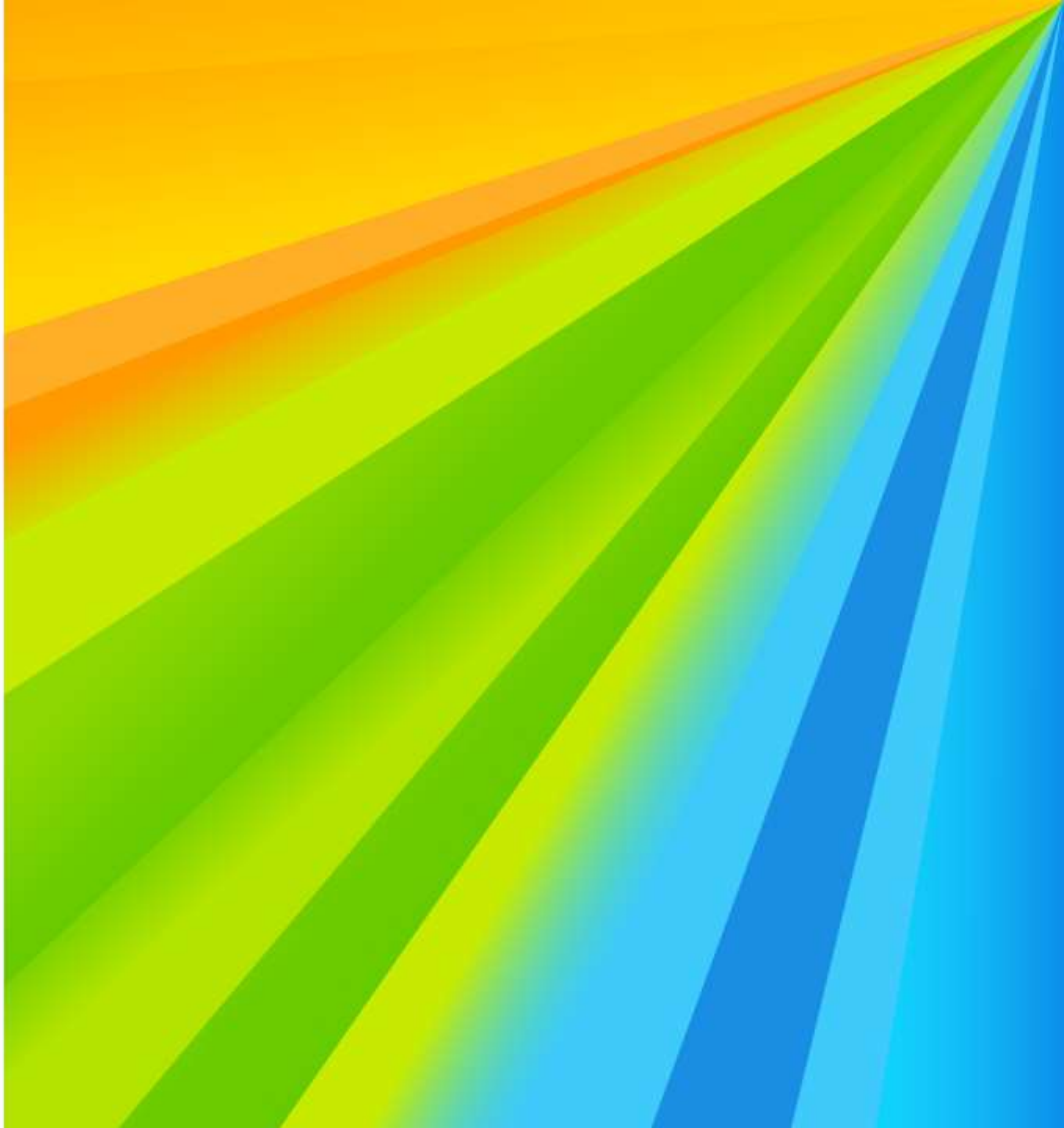


Store Now Decrypt Later



Post-quantum crypto standards are coming
It doesn’t matter if you believe in quantum computers or not

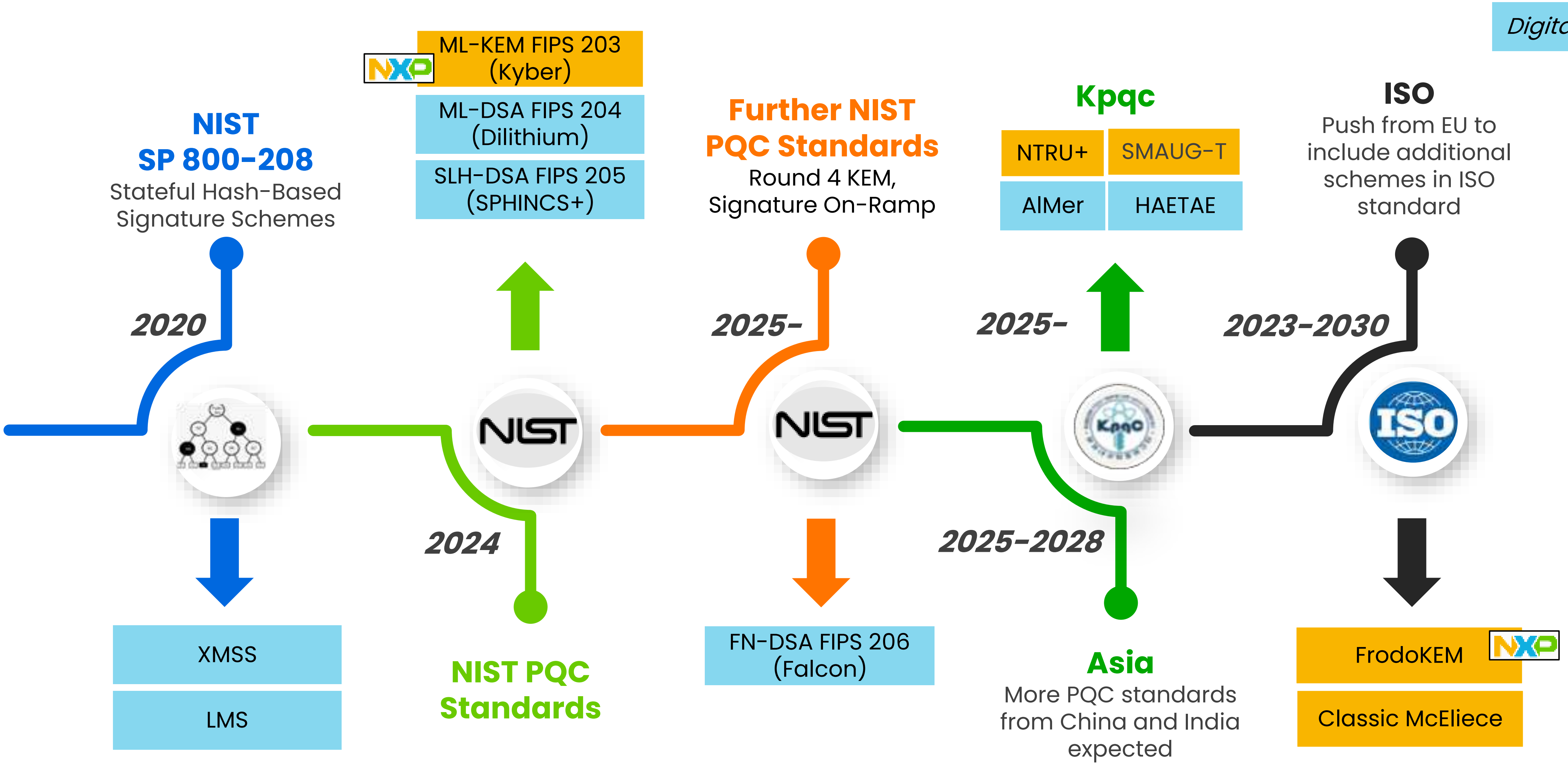
PQC Standards



PQC standards

Key Exchange

Digital Signature



NIST SP 800-208

Stateful Hash-Based Signature Schemes

2020

XMSS

LMS

ML-KEM FIPS 203 (Kyber)

ML-DSA FIPS 204 (Dilithium)

SLH-DSA FIPS 205 (SPHINCS+)

2024

NIST PQC Standards

Further NIST PQC Standards

Round 4 KEM, Signature On-Ramp

2025-

FN-DSA FIPS 206 (Falcon)

Kpqc

NTRU+

SMAUG-T

AIMer

HAETAE

2025-

2025-2028

Asia

More PQC standards from China and India expected

ISO

Push from EU to include additional schemes in ISO standard

2023-2030

FrodoKEM

Classic McEliece

New algorithms and standards

FIPS 203

Federal Information Processing Standards Publication

Module-Lattice-Based
Key-Encapsulation Mechanism Standard

Category: Computer Security   Subcategory: Cryptography

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
https://doi.org/10.6028/NIST.FIPS.203

Published August 13, 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Key Exchange / Encapsulation

FIPS 204

Federal Information Processing Standards Publication

Module-Lattice-Based Digital
Signature Standard

Category: Computer Security   Subcategory: Cryptography

Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900

This publication is available free of charge from:
https://doi.org/10.6028/NIST.FIPS.204

Published August 13, 2024

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

**Digital Signatures
(generic)**

NIST Special Publication 800-208

Recommendation for Stateful
Hash-Based Signature Schemes

David A. Cooper
Daniel C. Apon
Quynh H. Dang
Michael S. Davidson
Morris J. Dworkin
Carl A. Miller

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-208

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

lards Publication

ised Digital Signature

Subcategory: Cryptography

ology

from:

retary of Commerce for Standards and Technology

**Digital Signatures
(software / firmware signing)**

More ongoing and upcoming! FIPS 206, Round 4, On-Ramp, ISO, etc..

[1] ML-KEM, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.203.pdf
[2] ML-DSA, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.204.pdf
[3] SLH-DSA, https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.205.pdf
[4] LMS / XMSS, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf

PQC migration guidance

USA (NSA)

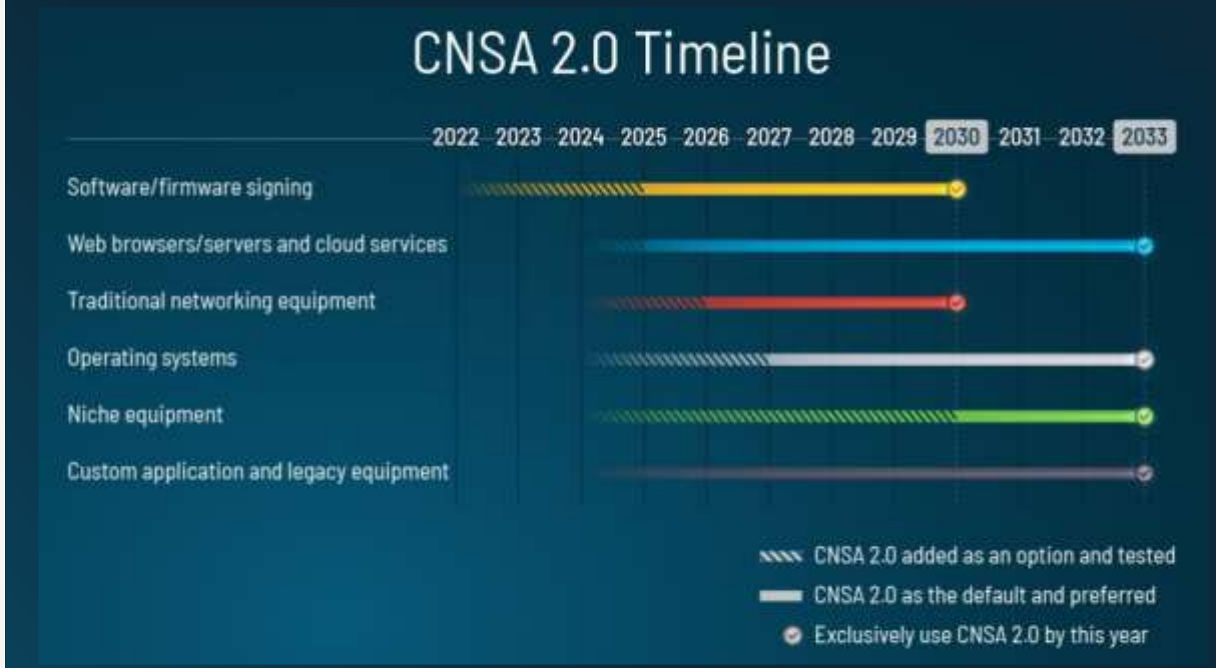
- NSA recommendation available
- Commercial National Security Algorithm Suite 2.0
- **Begin transitioning immediately**
- PQC FW signature supported **by 2025**
- PQC **transition complete by 2030** using SW update

Germany (BSI)

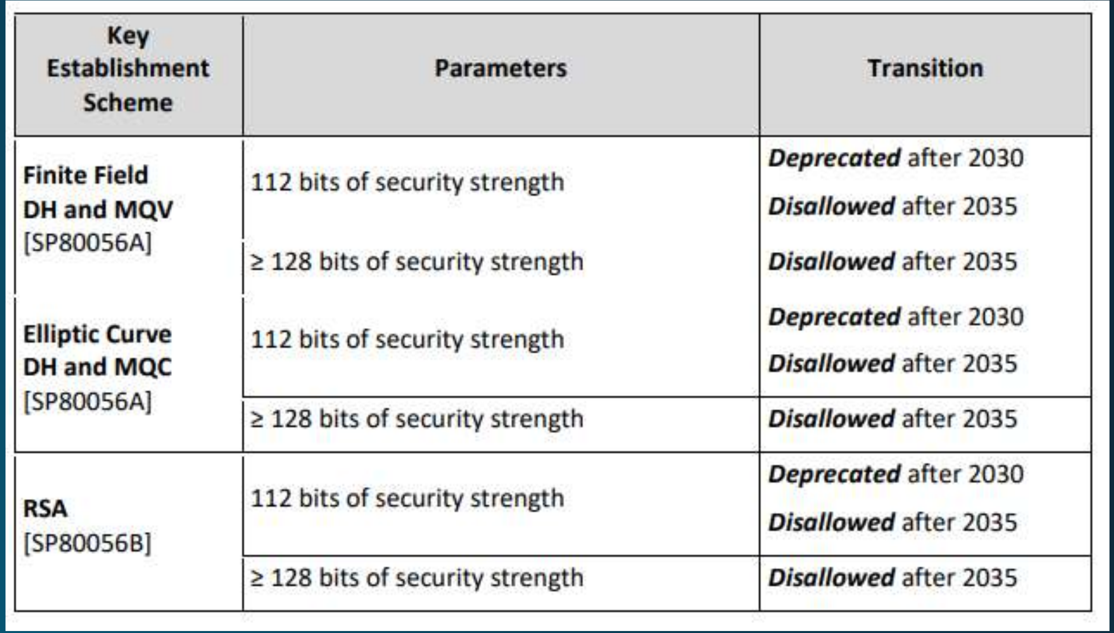
- BSI first recommendation (English)
- BSI considerations (German)
- Expectation is that beginning of 2030s, a relevant quantum computer is available to be a threat for high-secure applications
- “QKD is only suitable for specific use cases”

France (ANSSI)

- PQC recommendations for security products
- **“As soon as possible”** when long-lasting protection is required
- Others to **migrate to classic-PQC hybrid in 2025 – 2030**
- Switch to PQC-only expected by 2030

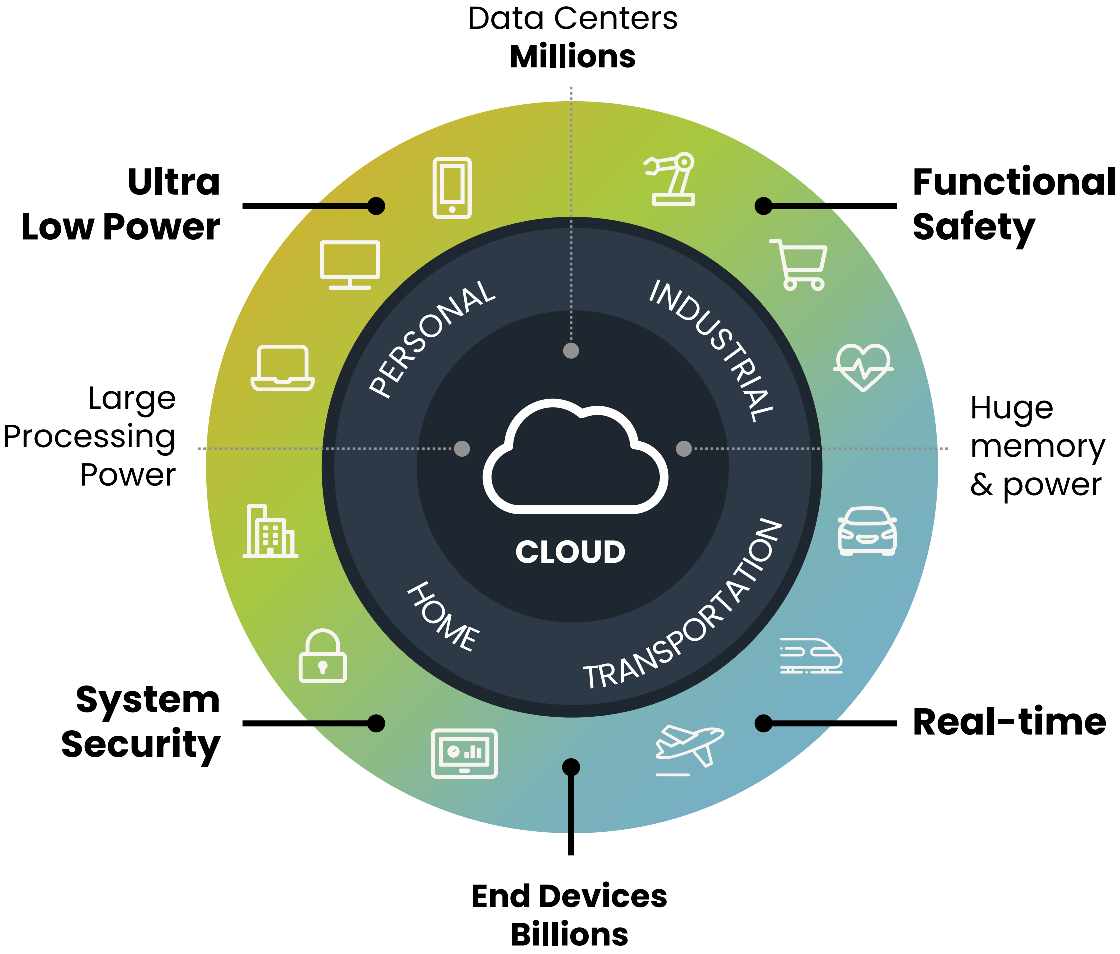


NIST IR 8547 (Initial Public Draft)
Transition to Post-Quantum Cryptography Standards



Key Establishment Scheme	Parameters	Transition
Finite Field DH and MQV [SP80056A]	112 bits of security strength	***Deprecated*** after 2030 ***Disallowed*** after 2035
	≥ 128 bits of security strength	***Disallowed*** after 2035
Elliptic Curve DH and MQC [SP80056A]	112 bits of security strength	***Deprecated*** after 2030 ***Disallowed*** after 2035
	≥ 128 bits of security strength	***Disallowed*** after 2035
RSA [SP80056B]	112 bits of security strength	***Deprecated*** after 2030 ***Disallowed*** after 2035
	≥ 128 bits of security strength	***Disallowed*** after 2035

Impact PQC on our eco-system



Data collection, processing and decisions at the edge
Devices securely connected to the cloud

No Silver Bullet

If a crypto scheme was better, we would have standardized this already

Cryptographic Keys

Orders of magnitude larger.
In the final: up to 1.3MB
Winners: up to 4.8KB
(ECC: 32 bytes, RSA: 384 bytes)

Performance

Varies: some faster some significantly slower.
SHA-3 is a dominating component (~80%)

Memory

Orders of magnitude more:
up 100KB memory of RAM when executing
NXP has dedicated implementations reaching ~16KB of RAM

Bandwidth & Power

Larger signatures (up to 4.6KB)
→ more bandwidth required
→ increase in power usage

Typical embedded use cases for new algorithms

Many more ongoing and upcoming!

		FIPS 203 ML-KEM	FIPS 204 ML-DSA	FIPS 205 (Verify) SLH-DSA	SP 800-208 (Verify) XMSS / LMS
Security Goals	Secure Boot	✓	✓	✓	✓
	Secure Update	✓	✓	✓	✓
	Secure Attestation	✗	✓	✗	✗
	Secure Debug / Test	✓	✓	✗	✗
	Certificates (PKI)	✗	✓	✓	✓**
	Runtime Crypto API	✓	✓	✓	✓
Protocols	TLS 1.3 (Hybrid)	✓	✓*	✗	✗
	IKEv2 (Hybrid)	✓	✓*	✗	✗
	GSMA eSIM	✓	✓	✗	✗
	GlobalPlatform: TEE/MCU	✓	✓	✓	✓

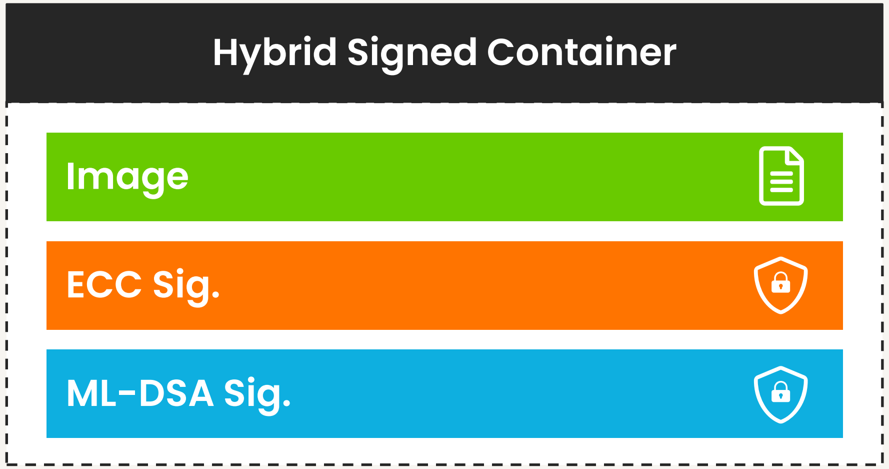
** Signatures for client authentication excluded from initial proposals, discussions ongoing*
*** Possible but the number of issued certificates should be carefully managed (e.g., Root CA)*

Hybrid migration

Transition Period

ECC / RSA benefit from decades of cryptanalysis including logical / physical attacks

Can combine security of both in a hybrid mode



Hybrid Signed Container

Image

ECC Sig.

ML-DSA Sig.

" NIST will **accommodate** the use of a hybrid key-establishment mode and dual signatures in FIPS 140 validation when suitably combined with a NIST-approved scheme "

" the BSI does not recommend using post-quantum cryptography alone, but **only "hybrid"** "

" the role of hybridation in the cryptographic security is crucial and will be **mandatory** for phases 1 and 2.

public key cryptography [...] would strongly benefit from the introduction of new alternative algorithms. "

Technical aspects of new algorithms

See pqm4 open source project for benchmarks! [A]
Assuming Cortex-M4 @ 200 MHz software-only.
For LMS numbers taken from Campos et al. [B]

Algorithm	PQC	Encaps	Decaps	SK	PK	CT	Algorithm
EC-P384	No	“Fast”	“Fast”	48 B	48 B	96 B	EC-P384
FIPS 203 (ML-KEM)	Yes	4 ms	4 ms	2 400 B	1 184 B	1 088 B	FIPS 203 (ML-KEM)

Algorithm	PQC	Encaps	Decaps	SK	PK	CT	Algorithm
ECDSA-P384	No	“Fast”	“Fast”	48 B	48 B	96 B	ECDSA-P384
FIPS 204 (ML-DSA)	Yes	31 ms	12 ms	4 032 B	1 952 B	3 309 B	FIPS 204 (ML-DSA)
FIPS 205 (SLH-DSA)***	Yes	77 s	68 ms	96 B	48 B	16 224 B	FIPS 205 (SLH-DSA)***
SP 800-20 (LMS/XMSS)	Yes	**(Stateful) 19 s	13 ms	48 B	48 B	1 860 B	SP 800-208 (LMS/XMSS)

**   NIST Level 3 parameter sets*
*** Significant reduction possible by increasing memory consumption for state*
**** New parameter sets coming that will improve performance & signature size!*

[A] pqm4; pqm4/benchmarks.md at master · mupq/pqm4 · GitHub
[B] Campos, Kohlstadt, Reith, Stöttinger; https://eprint.iacr.org/2020/470.pdf

What is the impact on the billions of embedded devices?

Automotive

70%

70% connected cars by 2025

Industrial & IoT

12B

IoT Edge & end nodes from **6B units** in 2021 to **12B units** in 2025

Mobile

60B

Tagging **60B products** per year by 2025

Communication Infrastructure

40B

Secure anchors & services for **40B processors**

Automotive

eGovernment

Bank cards

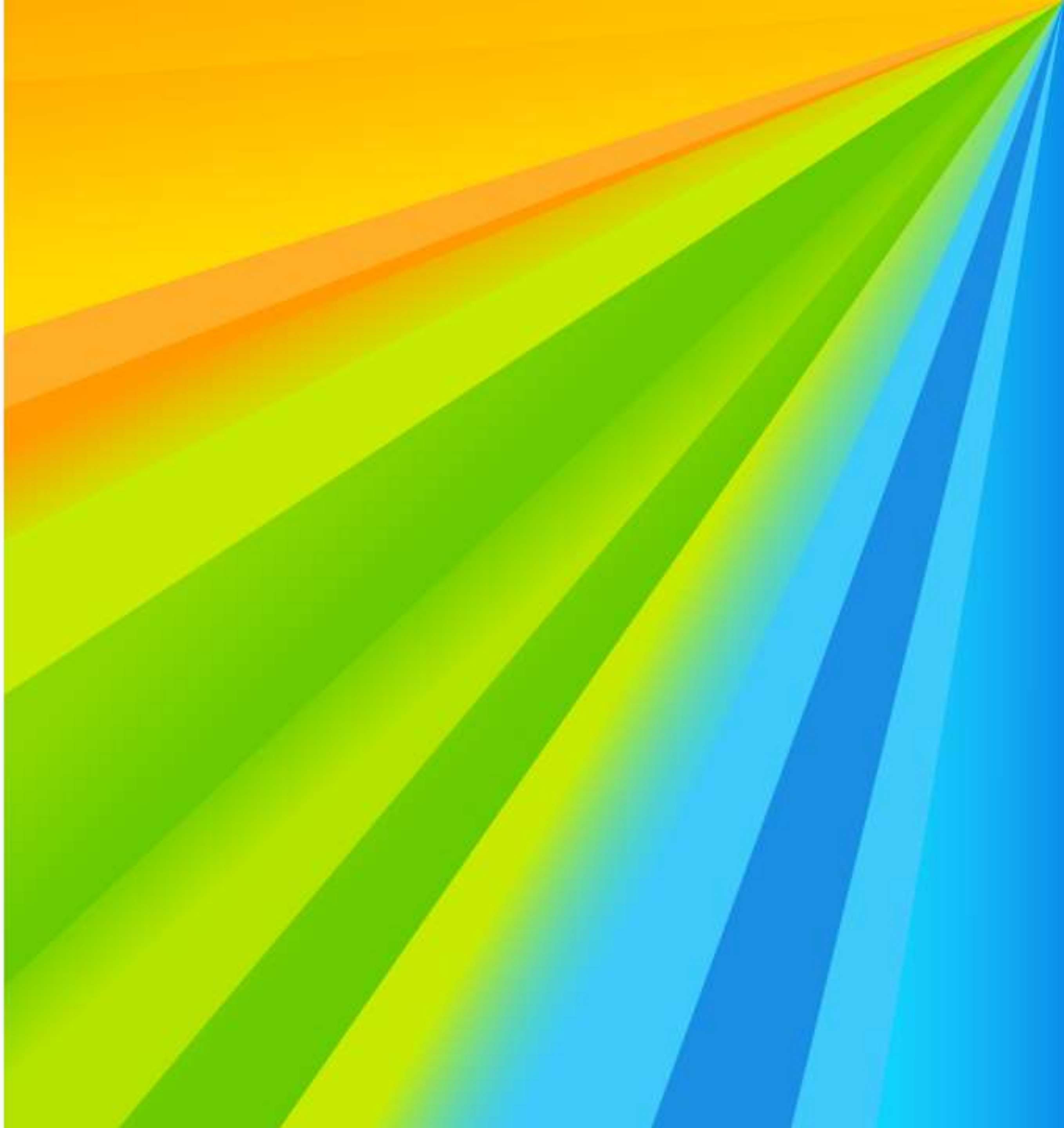
Smart mobility (MIFARE) cards

Tags & Authentication

Readers

Mobile

Learning with Errors



Cryptographic Suite for Algebraic Lattices (CRYSTALS)

The Cryptographic Suite for Algebraic Lattices (CRYSTALS) encompasses

- **Kyber**, a Key Encapsulation Mechanism (KEM) -> referred to in FIPS 203 as **ML-KEM**
- **Dilithium**, for Digital Signatures -> referred to in FIPS 204 as **ML-DSA**

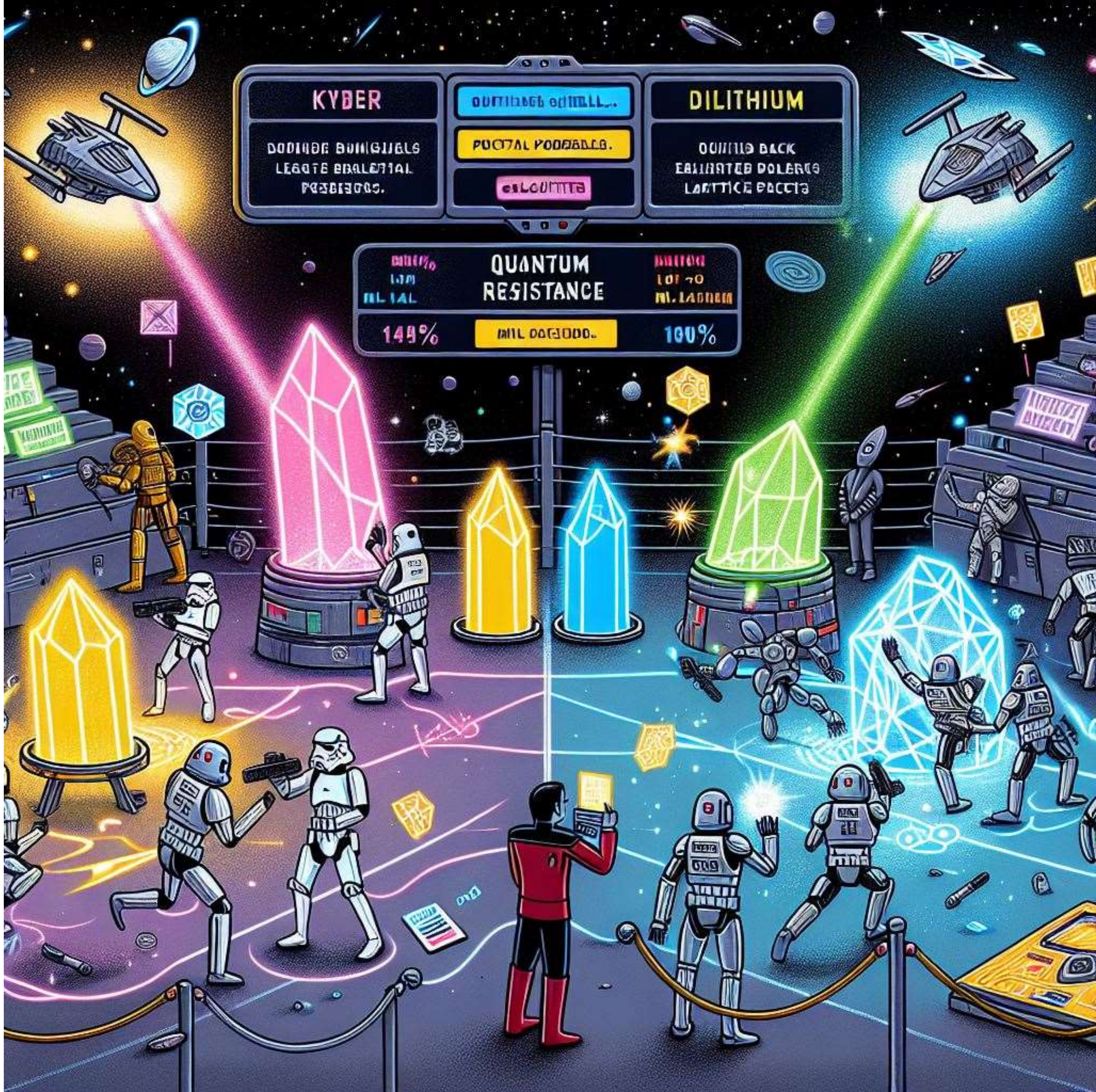
Theory: same building blocks

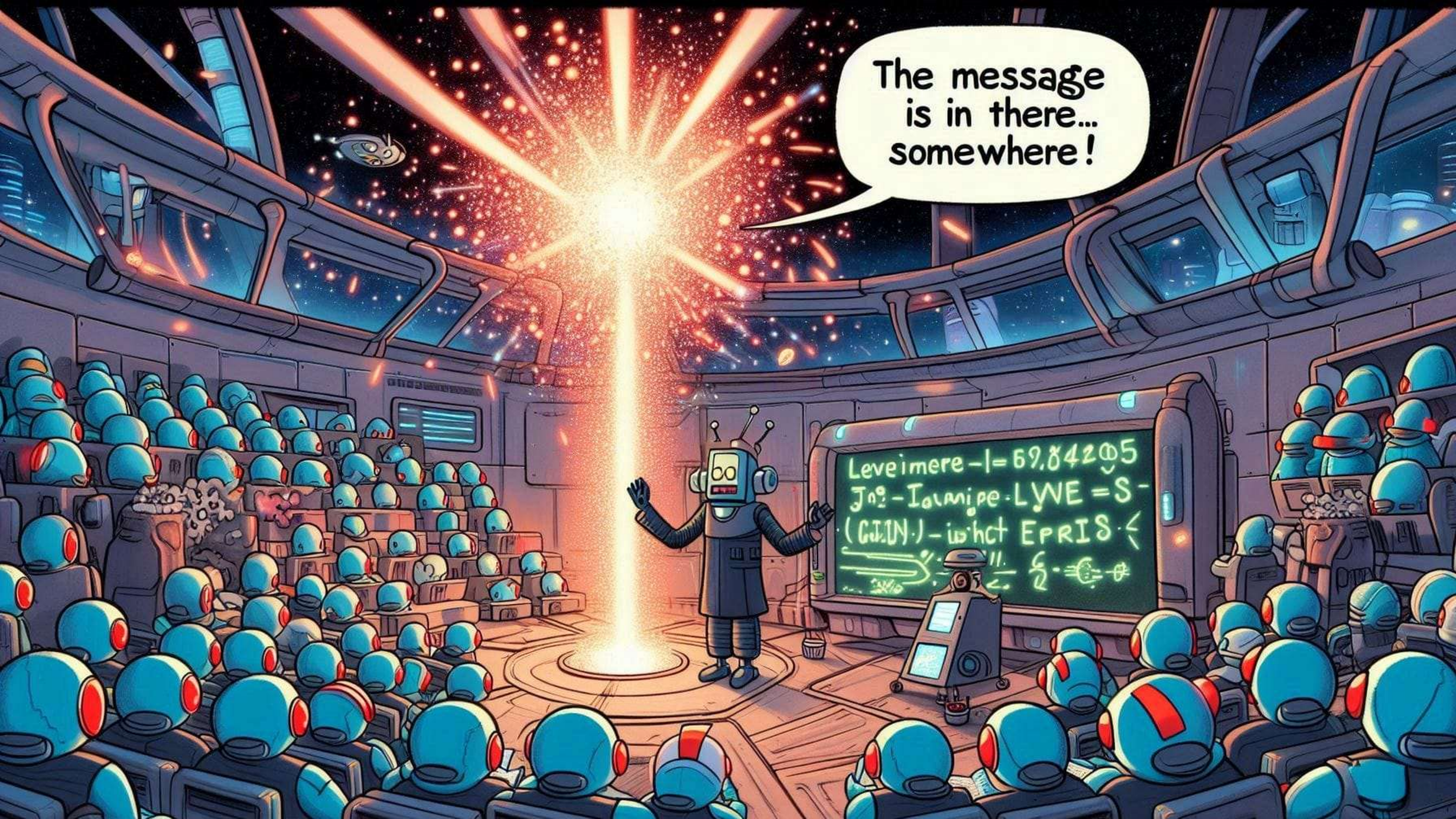
- Module Learning with Errors
- Number-Theoretic Transformations

<u>Many</u> new techniques to deal with!

Kyber uses the ‘Fujisaki-Okamoto Transform’ to get strong security

Dilithium uses ‘Rejection Sampling’ as a core component for producing signatures




The message is in there... somewhere!
Leveimere -I= 69,84205
Jn2-Iamipe-LYWE =S-
(GJIN)-iwhct EPRIS

Solving systems of linear equations

$$\begin{array}{ccccc} \mathbb{Z}_{13}^{7\times 4} & & \overset{\textbf{secret}}{\mathbb{Z}_{13}^{4\times 1}} & & \mathbb{Z}_{13}^{7\times 1} \\ \begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} & \times & \begin{bmatrix} \blacksquare \\ \blacksquare \\ \blacksquare \\ \blacksquare \end{bmatrix} & = & \begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix} \end{array}$$

Linear system problem: given **blue**, find **red**

Solving systems of linear equations

$$\overset{\mathbb{Z}_{13}^{7\times 4}}{\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix}} \times \overset{\overset{\textbf{secret}}{\mathbb{Z}_{13}^{4\times 1}}}{\begin{bmatrix} 6 \\ 9 \\ 11 \\ 11 \end{bmatrix}} = \overset{\mathbb{Z}_{13}^{7\times 1}}{\begin{bmatrix} 4 \\ 8 \\ 1 \\ 10 \\ 4 \\ 12 \\ 9 \end{bmatrix}}$$

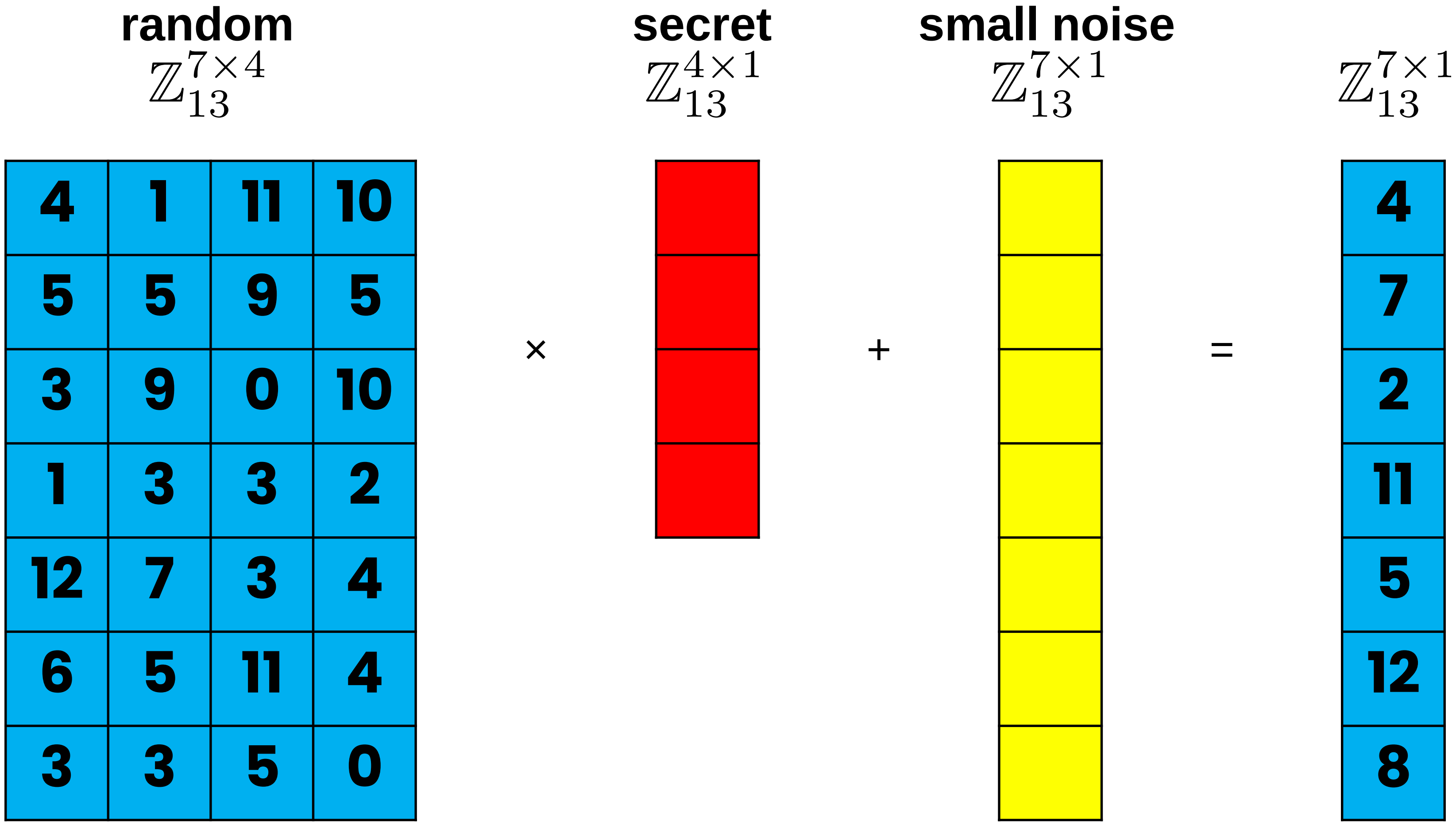
Easily solved using Gaussian elimination (Linear Algebra 101)

Linear system problem: given **blue**, find **red**

Learning with errors problem

$$
\begin{array}{ccccccc}
\textbf{random} & & \textbf{secret} & & \textbf{small noise} & & \\
\mathbb{Z}_{13}^{7\times 4} & & \mathbb{Z}_{13}^{4\times 1} & & \mathbb{Z}_{13}^{7\times 1} & & \mathbb{Z}_{13}^{7\times 1} \\
\begin{bmatrix}
4 & 1 & 11 & 10 \\
5 & 5 & 9 & 5 \\
3 & 9 & 0 & 10 \\
1 & 3 & 3 & 2 \\
12 & 7 & 3 & 4 \\
6 & 5 & 11 & 4 \\
3 & 3 & 5 & 0
\end{bmatrix}
& \times &
\begin{bmatrix}
6 \\ 9 \\ 11 \\ 11
\end{bmatrix}
& + &
\begin{bmatrix}
0 \\ -1 \\ 1 \\ 1 \\ 1 \\ 0 \\ -1
\end{bmatrix}
& = &
\begin{bmatrix}
4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8
\end{bmatrix}
\end{array}
$$

Learning with errors problem



$$\begin{array}{ccccccc}
\text{random} & & \text{secret} & & \text{small noise} & & \\
\mathbb{Z}_{13}^{7\times 4} & & \mathbb{Z}_{13}^{4\times 1} & & \mathbb{Z}_{13}^{7\times 1} & & \mathbb{Z}_{13}^{7\times 1} \\
\begin{bmatrix} 4 & 1 & 11 & 10 \\ 5 & 5 & 9 & 5 \\ 3 & 9 & 0 & 10 \\ 1 & 3 & 3 & 2 \\ 12 & 7 & 3 & 4 \\ 6 & 5 & 11 & 4 \\ 3 & 3 & 5 & 0 \end{bmatrix} & \times & \begin{bmatrix} \; \\ \; \\ \; \\ \; \end{bmatrix} & + & \begin{bmatrix} \; \\ \; \\ \; \\ \; \\ \; \\ \; \\ \; \end{bmatrix} & = & \begin{bmatrix} 4 \\ 7 \\ 2 \\ 11 \\ 5 \\ 12 \\ 8 \end{bmatrix}
\end{array}$$

Computational LWE problem: given **blue**, find **red**

Toy example versus real-world example

$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
5	5	9	5
3	9	0	10
1	3	3	2
12	7	3	4
6	5	11	4
3	3	5	0

$\mathbb{Z}_{2^{15}}^{752\times 8}$

8

752

2738	3842	3345	2979	•••
2896	595	3607		
377	1575			
2760				
•••				

752 × 8 × 15 bits = 11 KiB

Ring learning with errors problem

random

$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
10	4	1	11
11	10	4	1
1	11	10	4
4	1	11	10
10	4	1	11
11	10	4	1

Each row is the cyclic shift of the row above

Ring learning with errors problem

random
$\mathbb{Z}_{13}^{7\times 4}$

4	1	11	10
3	4	1	11
2	3	4	1
12	2	3	4
9	12	2	3
10	9	12	2
11	10	9	12

Each row is the cyclic shift of the row above

…

with a special wrapping rule:
x wraps to –*x* mod 13.

Ring learning with errors problem

random

$\mathbb{Z}_{13}^{7\times 4}$

4	**1**	**11**	**10**

Each row is the cyclic
shift of the row above

…

with a special wrapping rule:
x wraps to –*x* mod 13 (→ $\mathbb{Z}_{13}[x]/\langle x^4+1\rangle$)

So I only need to tell you the first row.

Ring learning with errors problem

$\mathbb{Z}_{13}[x]/\langle x^4 + 1\rangle$

$4 + 1x + 11x^2 + 10x^3$ **random**

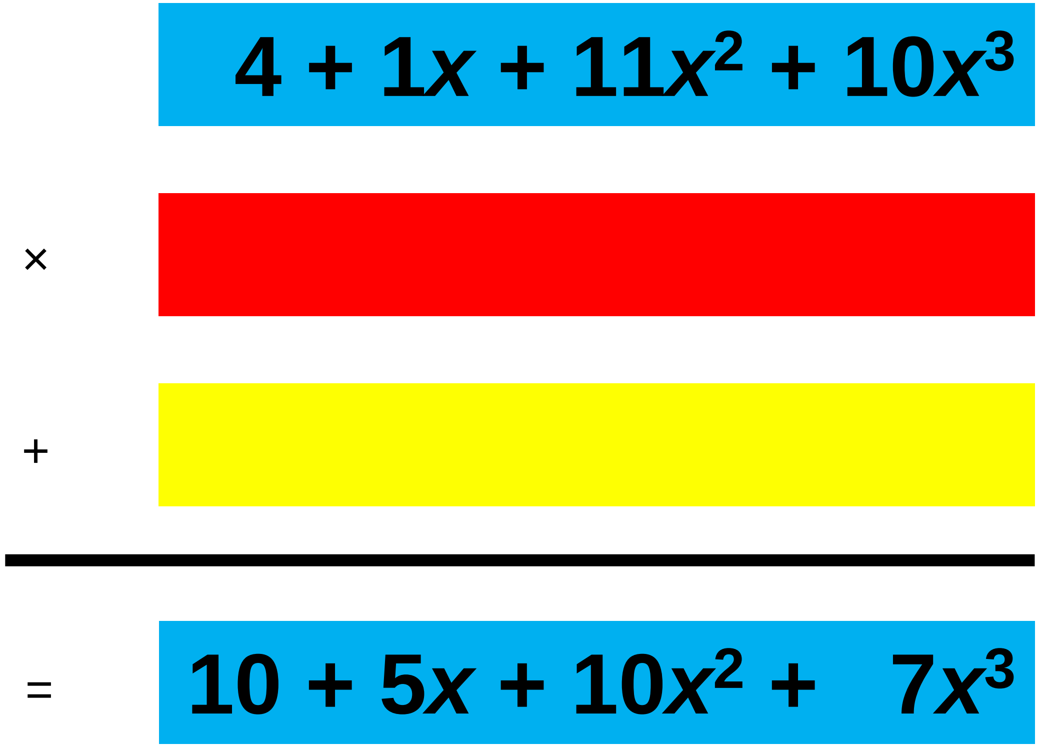
× $6 + 9x + 11x^2 + 11x^3$ **secret**

+ $0 - 1x + 1x^2 + 1x^3$ **small noise**

= $10 + 5x + 10x^2 + 7x^3$

Ring learning with errors problem

$\mathbb{Z}_{13}[x]/\langle x^4 + 1\rangle$



$4 + 1x + 11x^2 + 10x^3$ random

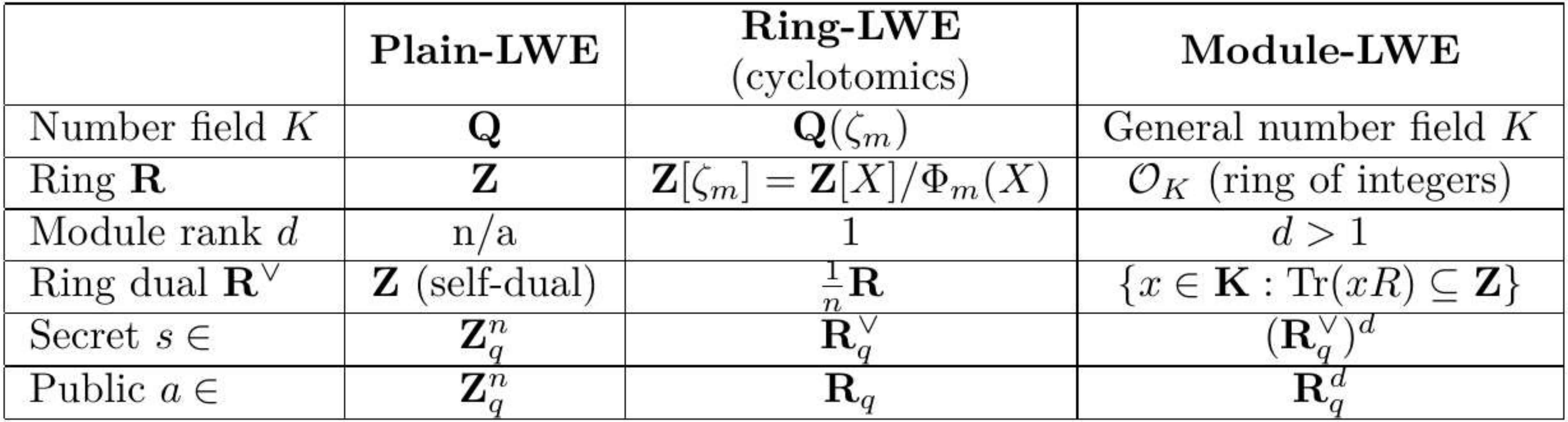
× secret

+ small noise

= $10 + 5x + 10x^2 + 7x^3$

Computational ring-LWE problem: given **blue**, find **red**

Algebraic variants of LWE



	Plain-LWE	**Ring-LWE** (cyclotomics)	**Module-LWE**
Number field K	$\mathbf{Q}$	$\mathbf{Q}(\zeta_m)$	General number field K
Ring $\mathbf{R}$	$\mathbf{Z}$	$\mathbf{Z}[\zeta_m] = \mathbf{Z}[X]/\Phi_m(X)$	$\mathcal{O}_K$ (ring of integers)
Module rank d	n/a	1	$d > 1$
Ring dual $\mathbf{R}^\vee$	$\mathbf{Z}$ (self-dual)	$\frac{1}{n}\mathbf{R}$	$\{x \in \mathbf{K} : \mathrm{Tr}(xR) \subseteq \mathbf{Z}\}$
Secret $s \in$	$\mathbf{Z}_q^n$	$\mathbf{R}_q^\vee$	$(\mathbf{R}_q^\vee)^d$
Public $a \in$	$\mathbf{Z}_q^n$	$\mathbf{R}_q$	$\mathbf{R}_q^d$

Even more variants exist: Polynomial-LWE, order-LWE, middle-product-LWE

Basic ring-LWE-DH key agreement

- Reformulation of Peikert's ring-LWE KEM (*PQCrypto 2014*)

public: "big" a in $R_q = \mathbf{Z}_q[x]/(x^n+1)$

Alice

secret:
random "small" s, e in R_q

Bob

secret:
random "small" s', e' in R_q

$$\xrightarrow{\quad b = a \bullet s + e \quad}$$

$$\xleftarrow{\quad b' = a \bullet s' + e' \quad}$$

Alice

shared secret:
$s \bullet b' = s \bullet (a \bullet s' \bullet e') \approx s \bullet a \bullet s'$

Bob

shared secret:
$b \bullet s' \approx s \bullet a \bullet s'$

These are only approximately equal ⇒ need rounding

What is the impact of PQC on Industrial IoT?

SECURE ELEMENTS AND END-TO-END SERVICES

NXP propels today's on-the-go lifestyle with intelligent mobile solutions that safely connect consumers and their technology to the world around them.

SECURE ELEMENTS AND END-TO-END SERVICES

CUSTOM HIGH-PERFORMANCE INTERFACES

SMART VOICE, AUDIO, AND HAPTIC SOLUTIONS

EFFICIENT CHARGING SOLUTIONS

DEFINING WHAT'S NEXT FOR MOBILE PHONES

NXP has been driving the mobile wallet expansion, advancing analog and charging solutions add more capabilities to mobile phones, notebooks, and tablets.

- NFC, eSE, eSIM, and UWB solutions
- Advanced analog solutions for personal computing
- Fast charging with USB Type-C

WEARABLES

Thanks to secure mobile payments, advanced audio solutions and tailored MCUs, wearables naturally blend into our lives.

- NFC+eSE mobile wallet solutions
- Highly integrated Arm® based MPUs and MCUs
- MiGLO™ NFMI radios for wireless audio

ACCESSORIES

NXP's anti-counterfeiting technology, among others products, support charging cables, power adapters, and wireless charging pads for mobile phones to help OEMs protect their brand and provides safety to their customers by making trusted accessories.

INDUSTRIAL

- Fit-for-purpose Scalable Processors
- Functional Safety & Security
- Industrial Connectivity & Control
- Machine Learning & Vision
- Comprehensive Software

PQC ON EMBEDDED DEVICES

What is embedded?

- NIST has recommended a focus on the Arm Cortex-M4

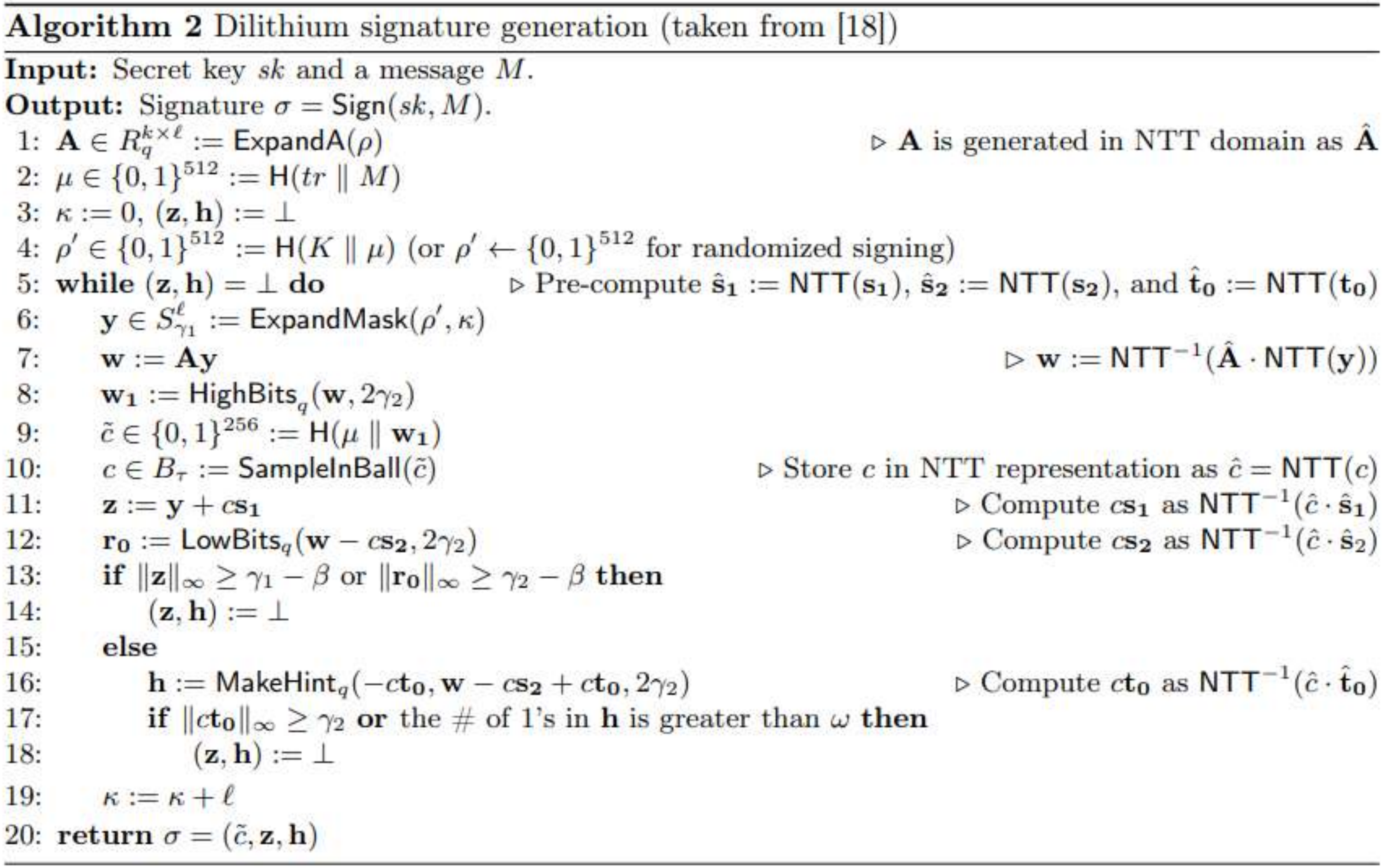
Pqm4: Post-quantum crypto library for the ARM Cortex-M4, STM32F4DISCOVERY
196 KiB of RAM and 1 MiB of Flash ROM

Low-power Edge computing: LPC800 Series

- 8 to 60 MHz Cortex-M0+ core
- $\{4, 8, 16\}$ KiB of SRAM
- $\{16, 32\}$ KiB Flash

The fastest implementations in pqm4 require ≈ 49, ≈ 80 and ≈ 116 KiB memory for Dilithium-{2,3,5}.

DILITHIUM SIGNATURE GENERATION



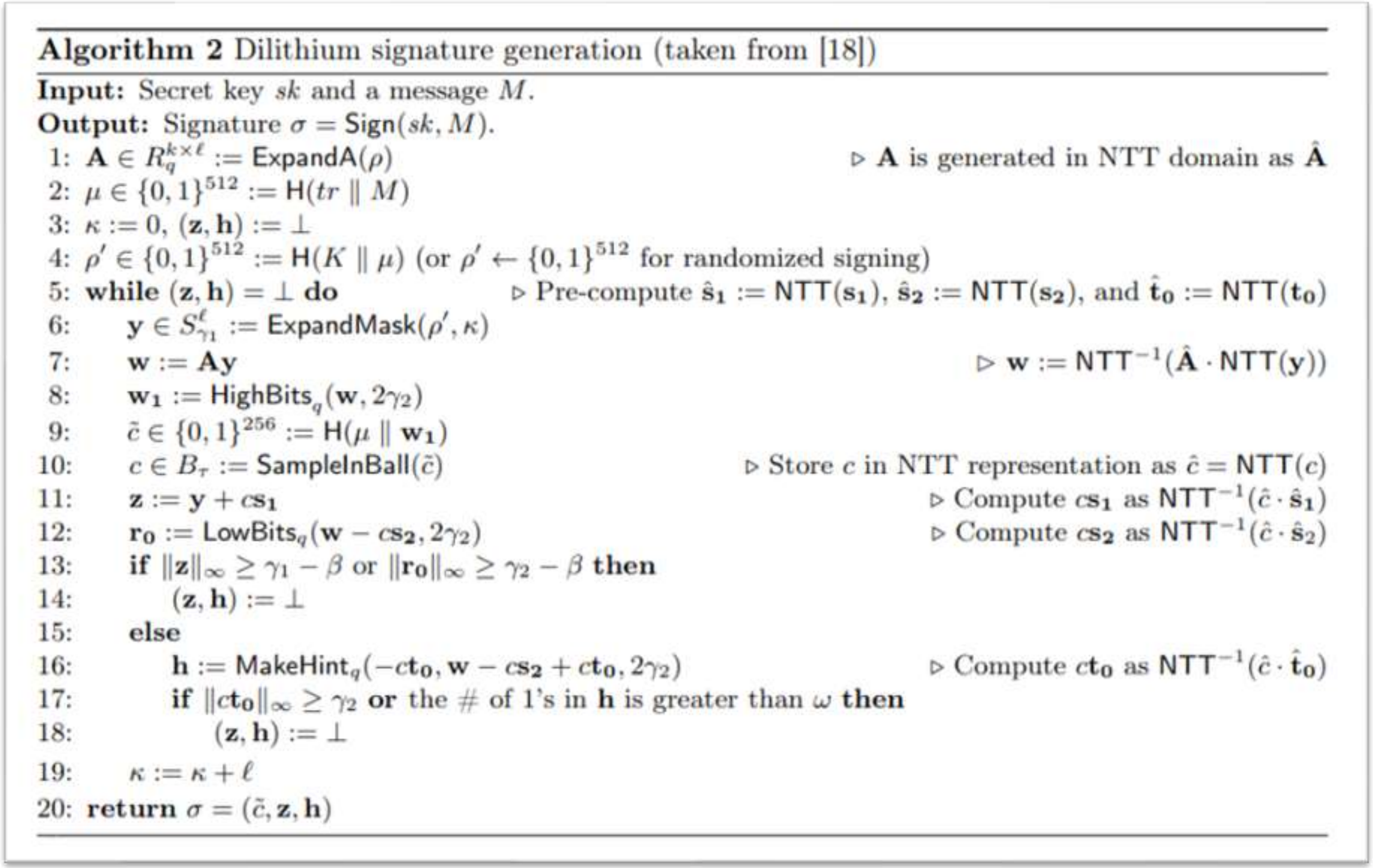
Algorithm 2 Dilithium signature generation (taken from [18])

Input: Secret key sk and a message M.

Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.

```
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \parallel \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \parallel \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

DILITHIUM SIGNATURE GENERATION



Algorithm 2 Dilithium signature generation (taken from [18])

Input: Secret key sk and a message M.

Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.

```
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \,\|\, M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \,\|\, \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \,\|\, \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

Polynomials from

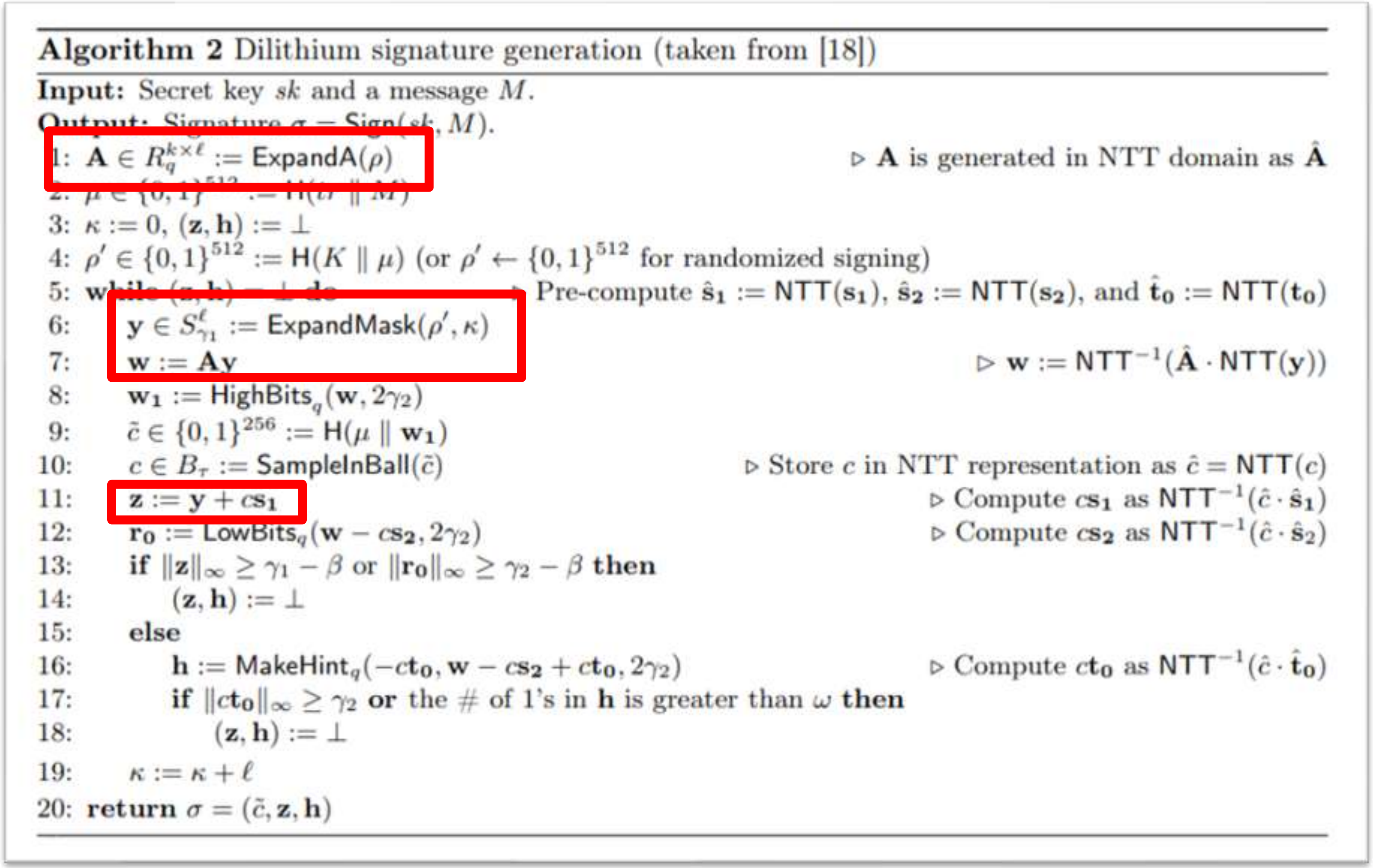
$$R_q = \mathbb{Z}_q[X]/(X^{256} + 1)$$

where $q = 2^{23} - 2^{13} + 1$ and stored as 32-bit values.

→ One R_q elements needs **1KB**

Dilithium-3: $(k, \ell) = (6,5)$

DILITHIUM SIGNATURE GENERATION



```
Algorithm 2 Dilithium signature generation (taken from [18])
Input: Secret key $sk$ and a message $M$.
Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \,\|\, M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \,\|\, \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \,\|\, \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

Polynomials from

$$R_q = \mathbb{Z}_q[X]/(X^{256}+1)$$

where $q = 2^{23} - 2^{13} + 1$ and stored as 32-bit values.

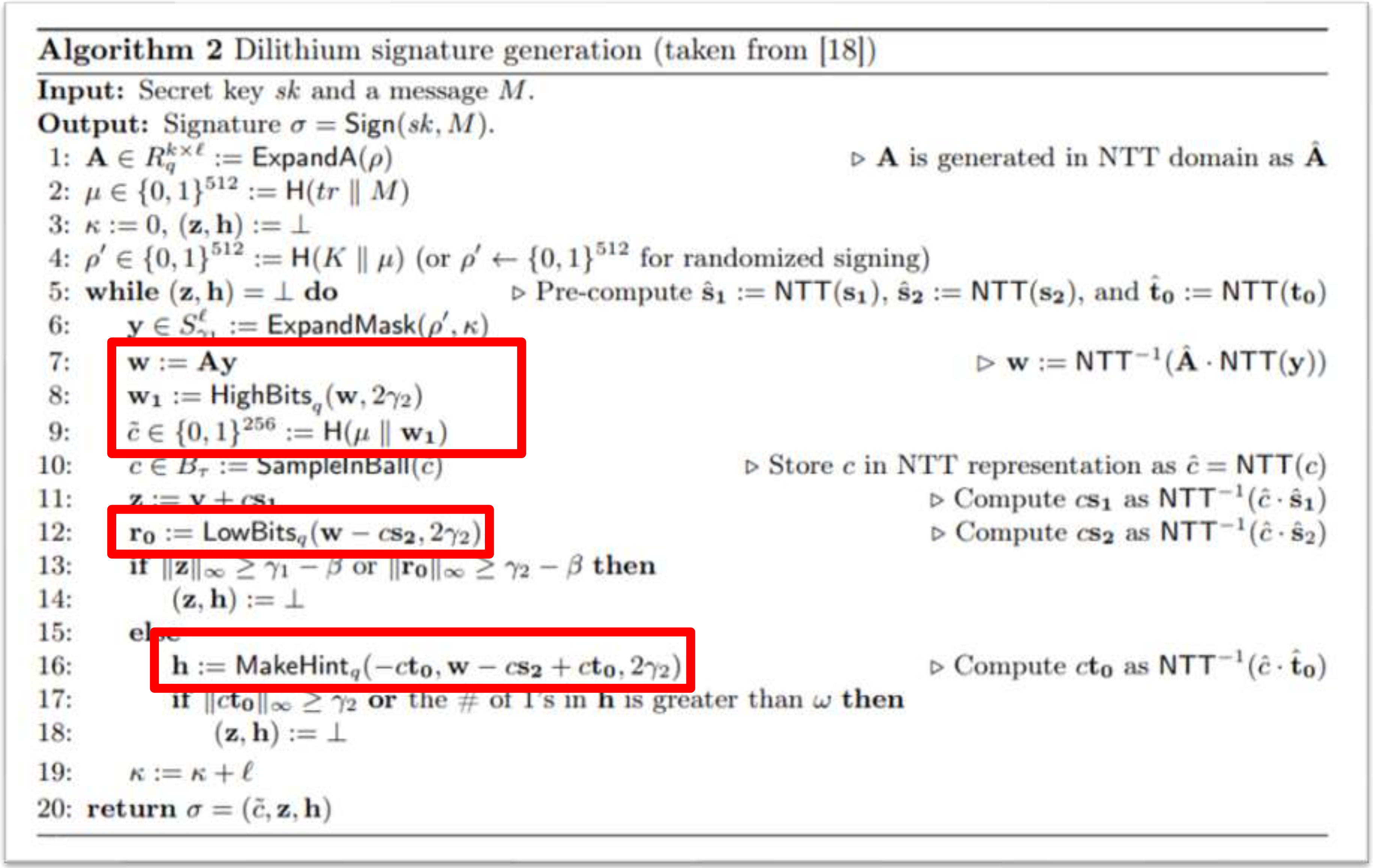
→ One R_q elements needs **1KB**

Dilithium-3: $(k, \ell) = (6,5)$

(Re-)generate matrix A and y on-the-fly

- Reduce by $k \cdot \ell$ KB for A
  → **30 KB**
- Reduce by ℓ KB for y
  → **5 KB**

DILITHIUM SIGNATURE GENERATION



Algorithm 2 Dilithium signature generation (taken from [18])

Input: Secret key sk and a message M.

Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.

```
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \parallel \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \parallel \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

Polynomials from
$$R_q = \mathbb{Z}_q[X]/(X^{256} + 1)$$
where $q = 2^{23} - 2^{13} + 1$ and stored as 32-bit values.
→ One R_q elements needs **1KB**

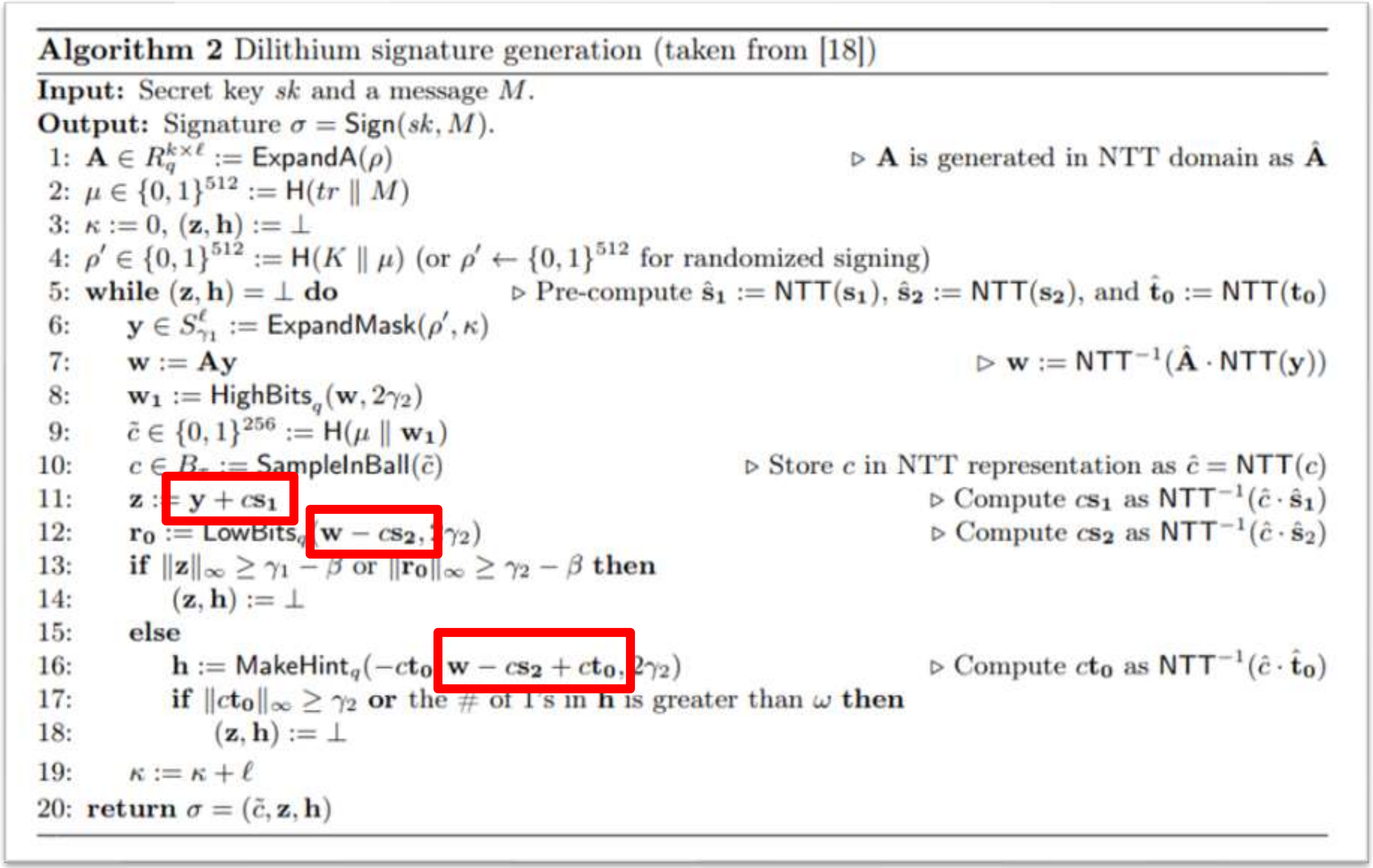
Dilithium-3: $(k, \ell) = (6,5)$

(Re-)generate matrix A and y on-the-fly: ~~80~~, 45 KB

Compress w

- Store values as 24-bit
- One R_q elements needs 768 bytes
- Packing and unpacking is simple and efficient
- Reduces memory by Reduce by $256k$ bytes → 1.5 KB

DILITHIUM SIGNATURE GENERATION



Algorithm 2 Dilithium signature generation (taken from [18])

```
Input: Secret key $sk$ and a message $M$.
Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \parallel \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \parallel \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, \gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

Polynomials from

$$R_q = \mathbb{Z}_q[X]/(X^{256}+1)$$

where $q = 2^{23} - 2^{13} + 1$ and stored as 32-bit values.

→ One R_q elements needs **1KB**

Dilithium-3: $(k, \ell) = (6,5)$

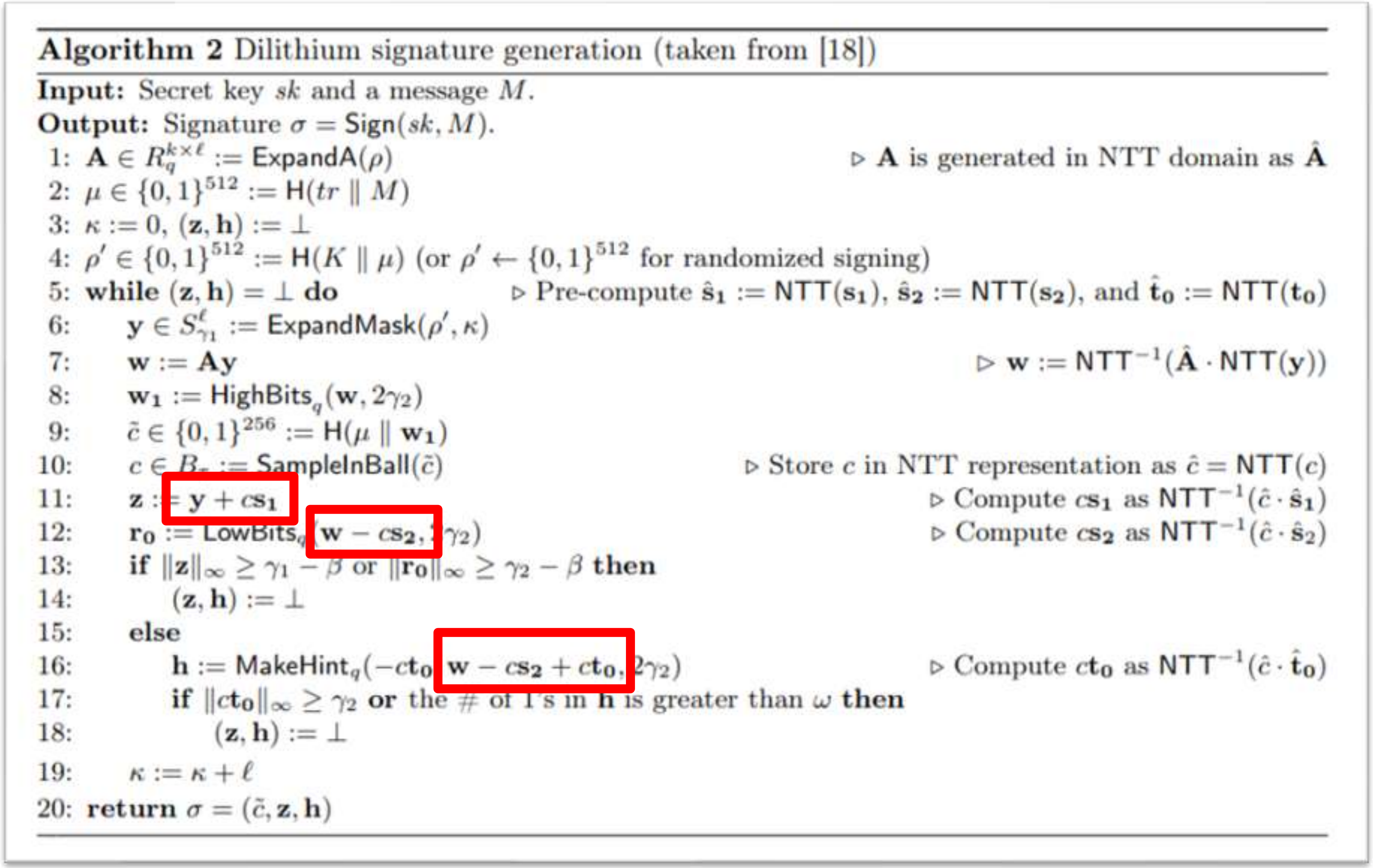
(Re-)generate matrix A and y on-the-fly: ~~80 KB~~ → 45 KB

Compress w: ~~45 KB~~ → 43.5 KB

Compressing multiplications

- NTT used for faster polynomial multiplication
- Secret key coefficient range is much smaller
- Not using NTT reduces by $2k + \ell$ KB → **17 KB**

DILITHIUM SIGNATURE GENERATION



Algorithm 2 Dilithium signature generation (taken from [18])

Input: Secret key sk and a message M.

Output: Signature $\sigma = \mathsf{Sign}(sk, M)$.

```
$\mathbf{A} \in R_q^{k \times \ell} := \mathsf{ExpandA}(\rho)$    ▷ $\mathbf{A}$ is generated in NTT domain as $\hat{\mathbf{A}}$
$\mu \in \{0,1\}^{512} := \mathsf{H}(tr \,\|\, M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho' \in \{0,1\}^{512} := \mathsf{H}(K \,\|\, \mu)$ (or $\rho' \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ do    ▷ Pre-compute $\hat{\mathbf{s}}_1 := \mathsf{NTT}(\mathbf{s}_1)$, $\hat{\mathbf{s}}_2 := \mathsf{NTT}(\mathbf{s}_2)$, and $\hat{\mathbf{t}}_0 := \mathsf{NTT}(\mathbf{t}_0)$
    $\mathbf{y} \in S_{\gamma_1}^{\ell} := \mathsf{ExpandMask}(\rho', \kappa)$
    $\mathbf{w} := \mathbf{A}\mathbf{y}$    ▷ $\mathbf{w} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \cdot \mathsf{NTT}(\mathbf{y}))$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathsf{H}(\mu \,\|\, \mathbf{w}_1)$
    $c \in B_\tau := \mathsf{SampleInBall}(\tilde{c})$    ▷ Store $c$ in NTT representation as $\hat{c} = \mathsf{NTT}(c)$
    $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$    ▷ Compute $c\mathbf{s}_1$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$    ▷ Compute $c\mathbf{s}_2$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ or $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ then
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$    ▷ Compute $c\mathbf{t}_0$ as $\mathsf{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$
        if $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$ or the # of 1's in $\mathbf{h}$ is greater than $\omega$ then
            $(\mathbf{z}, \mathbf{h}) := \perp$
    $\kappa := \kappa + \ell$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$
```

Polynomials from

$$R_q = \mathbb{Z}_q[X]/(X^{256} + 1)$$

where $q = 2^{23} - 2^{13} + 1$ and stored as 32-bit values.

→ One R_q elements needs **1KB**

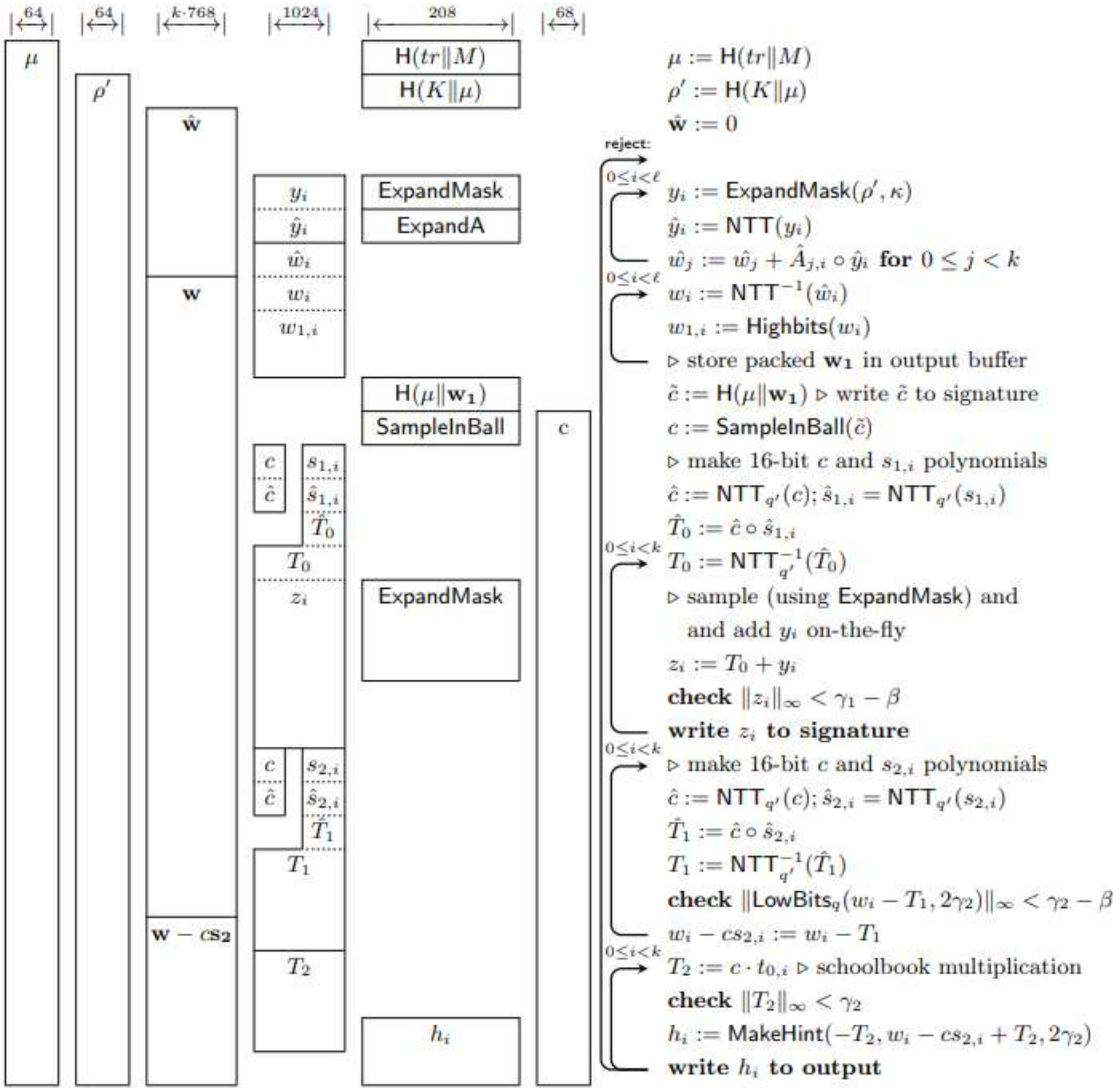
Dilithium-3: $(k, \ell) = (6,5)$

(Re-)generate matrix A and y on-the-fly: ~~80 KB~~ → 45 KB

Compress w: ~~45 KB~~ → 43.5 KB

Compressing multiplications ~~43.5 KB~~ → 26.5 KB

Variable Allocation



```
$\mu := \mathsf{H}(tr\|M)$
$\rho' := \mathsf{H}(K\|\mu)$
$\hat{\mathbf{w}} := 0$
reject:
  $0 \le i < \ell$
    $y_i := \mathsf{ExpandMask}(\rho', \kappa)$
    $\hat{y}_i := \mathsf{NTT}(y_i)$
    $\hat{w}_j := \hat{w}_j + \hat{A}_{j,i} \circ \hat{y}_i$ for $0 \le j < k$
  $0 \le i < \ell$
    $w_i := \mathsf{NTT}^{-1}(\hat{w}_i)$
    $w_{1,i} := \mathsf{Highbits}(w_i)$
    ▷ store packed $\mathbf{w_1}$ in output buffer
  $\tilde{c} := \mathsf{H}(\mu\|\mathbf{w_1})$ ▷ write $\tilde{c}$ to signature
  $c := \mathsf{SampleInBall}(\tilde{c})$
  ▷ make 16-bit $c$ and $s_{1,i}$ polynomials
  $\hat{c} := \mathsf{NTT}_{q'}(c)$; $\hat{s}_{1,i} = \mathsf{NTT}_{q'}(s_{1,i})$
  $\hat{T}_0 := \hat{c} \circ \hat{s}_{1,i}$
  $0 \le i < k$
    $T_0 := \mathsf{NTT}_{q'}^{-1}(\hat{T}_0)$
    ▷ sample (using ExpandMask) and
      and add $y_i$ on-the-fly
    $z_i := T_0 + y_i$
    check $\|z_i\|_\infty < \gamma_1 - \beta$
    write $z_i$ to signature
  $0 \le i < k$
    ▷ make 16-bit $c$ and $s_{2,i}$ polynomials
    $\hat{c} := \mathsf{NTT}_{q'}(c)$; $\hat{s}_{2,i} = \mathsf{NTT}_{q'}(s_{2,i})$
    $\hat{T}_1 := \hat{c} \circ \hat{s}_{2,i}$
    $T_1 := \mathsf{NTT}_{q'}^{-1}(\hat{T}_1)$
    check $\|\mathsf{LowBits}_q(w_i - T_1, 2\gamma_2)\|_\infty < \gamma_2 - \beta$
    $w_i - cs_{2,i} := w_i - T_1$
  $0 \le i < k$
    $T_2 := c \cdot t_{0,i}$ ▷ schoolbook multiplication
    check $\|T_2\|_\infty < \gamma_2$
    $h_i := \mathsf{MakeHint}(-T_2, w_i - cs_{2,i} + T_2, 2\gamma_2)$
    write $h_i$ to output
```

(Re-)generate matrix A and y on-the-fly: ~~80 KB~~ → 45 KB

Compress w: ~~45 KB~~ → 43.5 KB

Compressing multiplications ~~43.5 KB~~ → 26.5 KB

Variable Allocation:

Total of

$64 + 64 + 768k + 1024 + 208 + 68$ bytes → 5268 bytes

In practice: 6.5 KB needed

From Theory to Practice: Small-Memory Implementations

Do these implementations actually run on embedded systems?

		pqm4		NXP PQC [A]		Slower	Smaller
		Runtime	RAM	Runtime	RAM	Runtime	RAM
Dilithium-2	Sign	19 ms	50 kB	61 ms	5 kB	3.2x	10.0x
	Verify	7 ms	11 kB	16 ms	3 kB	2.3x	3.7x
Dilithium-3	Sign	31 ms	69 kB	119 ms	7 kB	3.8x	9.9x
	Verify	12 ms	10 kB	29 ms	3 kB	2.4x	3.3x
Dilithium-5	Sign	42 ms	123 kB	168 ms	8 kB	4.0x	15.4x
	Verify	21 ms	12 kB	50 ms	3 kB	2.4x	4.0x

- **All Dilithium parameter sets will fit on a device with ~8KB memory!**
- ☹ Factor 3 to 4 decrease in performance
- ☺ Hardware accelerators will mitigate this

[A] Bos, Renes, Sprenkels; Dilithium for memory constrained devices; Africacrypt 2022

Example of what we do at NXP

Joppe W. Bos, Joost Renes and Christine van Vredendaal: *Polynomial Multiplication with Contemporary Co-Processors: Beyond Kronecker, Schönhage-Strassen & Nussbaumer*. USENIX Security Symposium 2022.

Implementing Classical cryptography

NIST

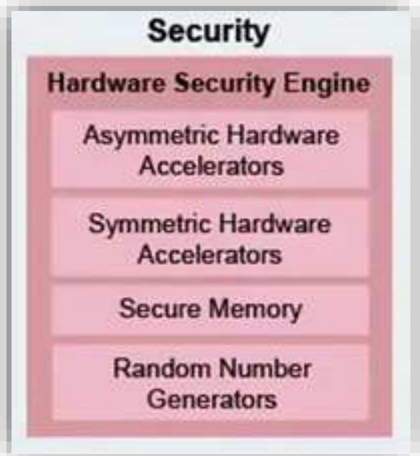
Information Technology Laboratory

COMPUTER SECURITY RESOURCE CENTER

PUBLICATIONS

FIPS 186-5 (Draft)

Digital Signature Standard (DSS)

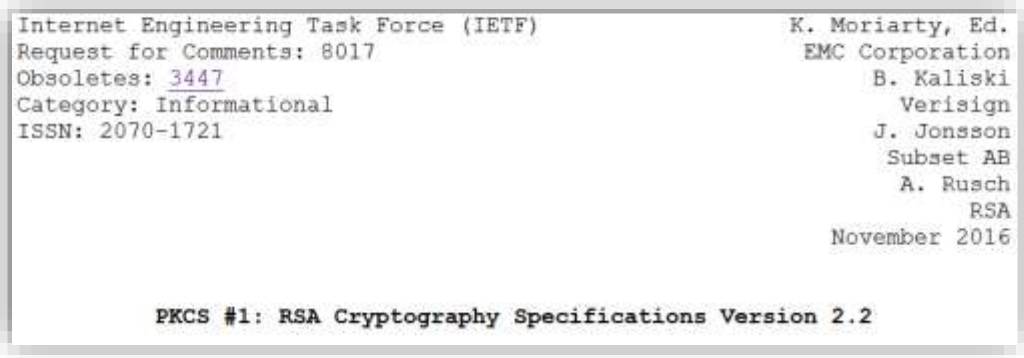


Security

Hardware Security Engine

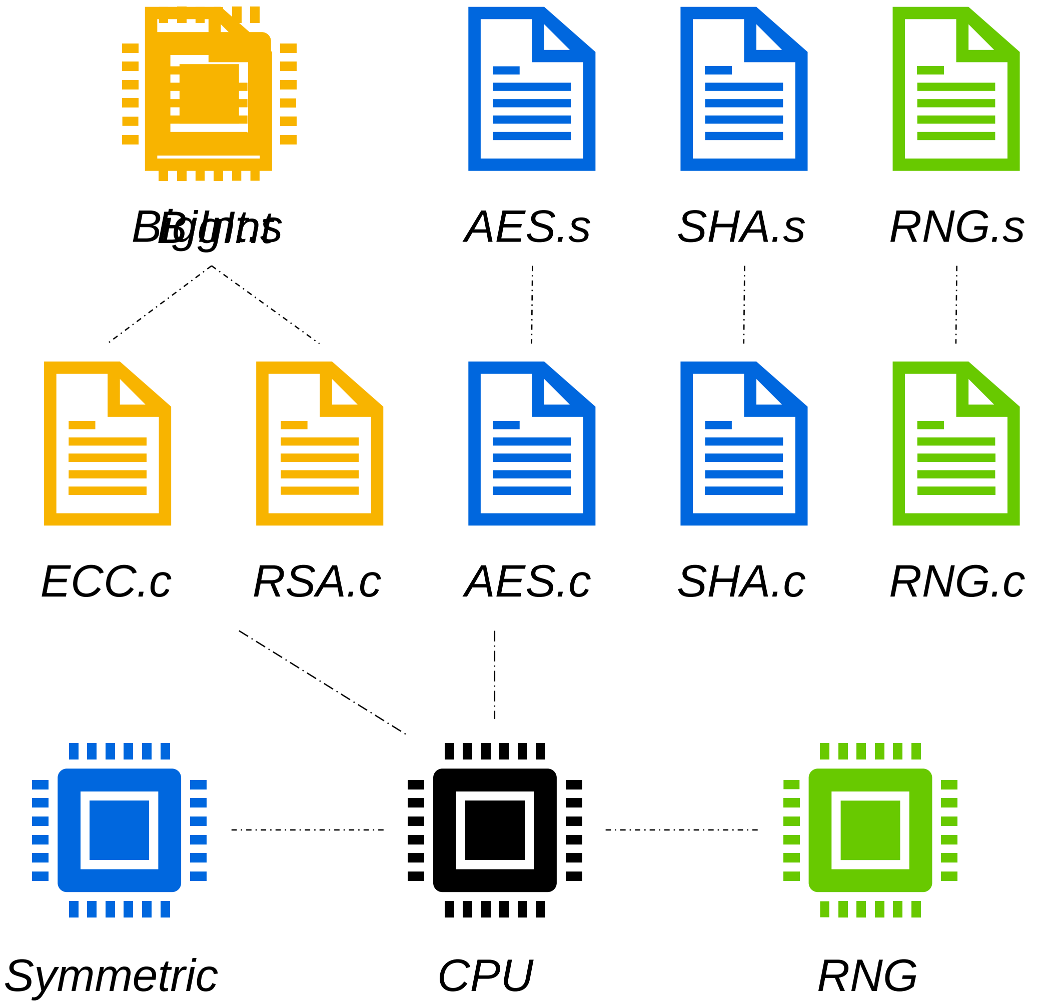
- Asymmetric Hardware Accelerators
- Symmetric Hardware Accelerators
- Secure Memory
- Random Number Generators

S32G2 automotive processor spec



```
Internet Engineering Task Force (IETF)                    K. Moriarty, Ed.
Request for Comments: 8017                                 EMC Corporation
Obsoletes: 3447                                                 B. Kaliski
Category: Informational                                           Verisign
ISSN: 2070-1721                                                 J. Jonsson
                                                                 Subset AB
                                                                  A. Rusch
                                                                       RSA
                                                             November 2016


         PKCS #1: RSA Cryptography Specifications Version 2.2
```



1. https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/s32g-vehicle-network-processors:S32G-PROCESSORS

Implementing post-quantum cryptography

NIST

Information Technology Laboratory

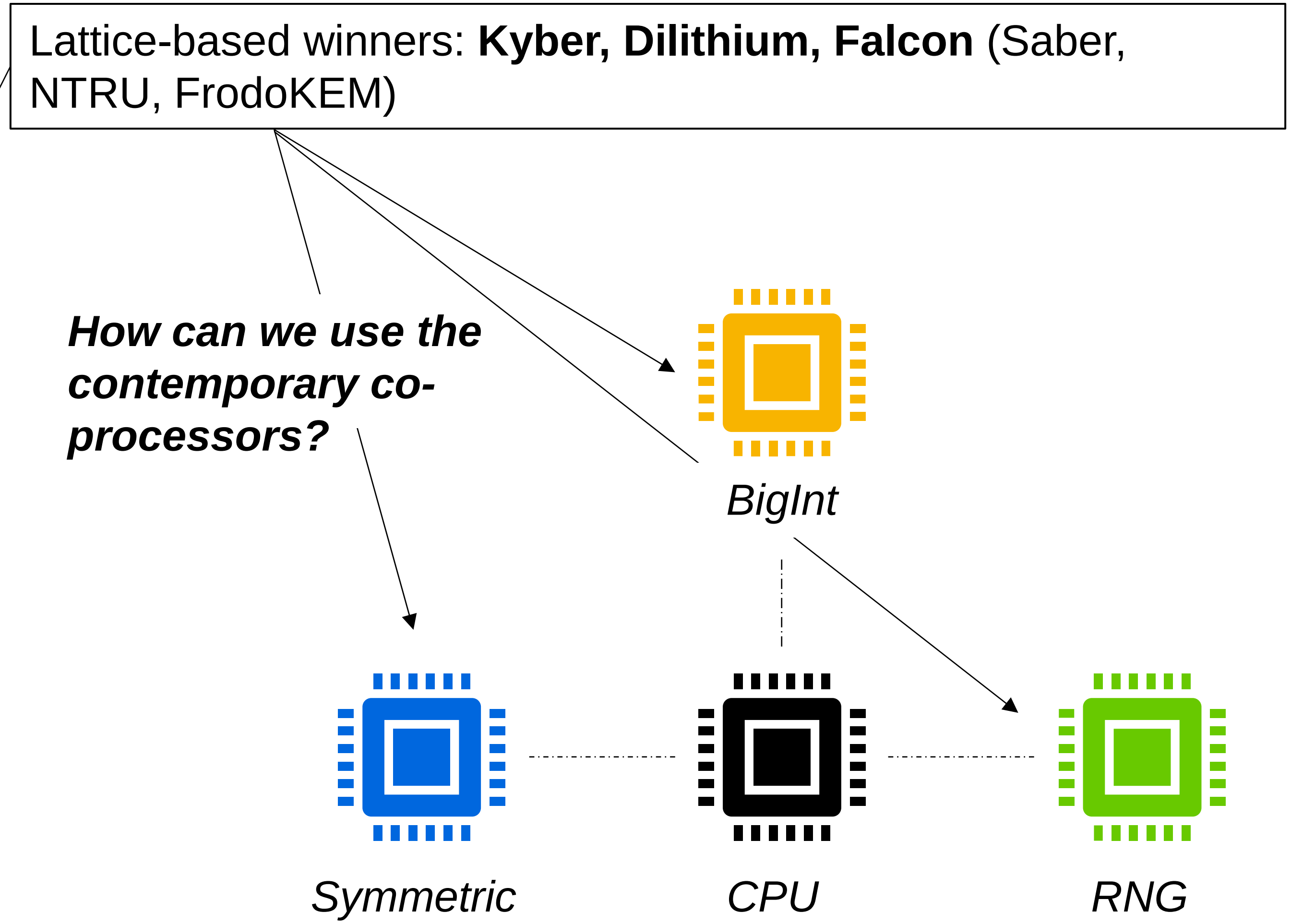
COMPUTER SECURITY RESOURCE CENTER

PROJECTS

Post-Quantum Cryptography PQC

Project Overview

NIST has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. **Full details can be found in the** Post-Quantum Cryptography Standardization **page.**



Lattice-based winners: **Kyber, Dilithium, Falcon** (Saber, NTRU, FrodoKEM)

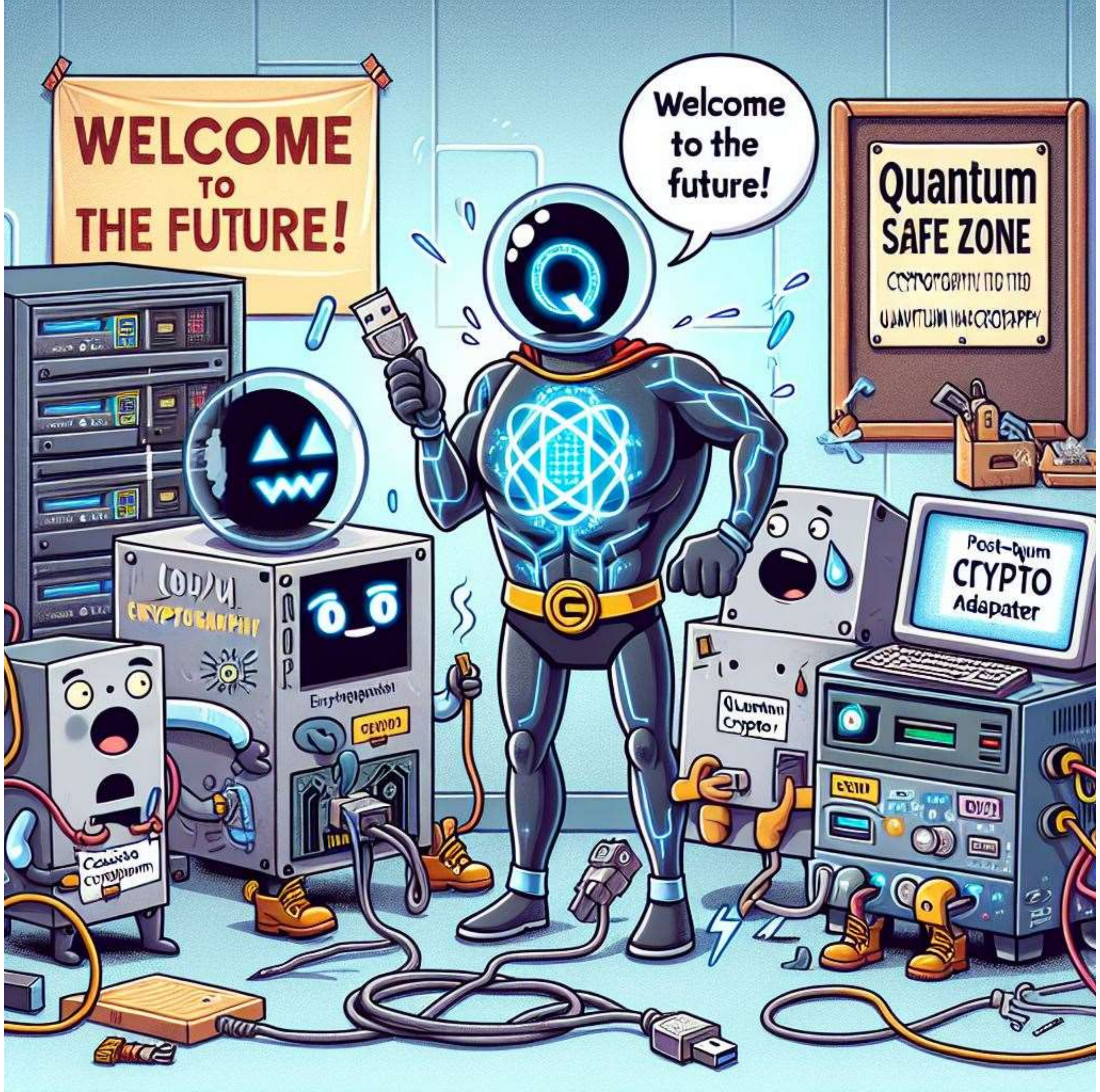
How can we use the contemporary co-processors?

Re-using existing HW

Approach	Core	Structure	Size
RSA	Modular multiplication	$(\mathbb{Z}/n\mathbb{Z})^*$	n is 3072-bit
ECC	Elliptic curve scalar multiplication	$\mathrm{E}(\mathbb{F}_p)$	p is 256-bit
Lattice	Polynomial multiplication	$(\mathbb{Z}/q\mathbb{Z})[X]/(X^n + 1)$	q is 16-bit n is 256

Co-pro present in chips

Can we use this?


WELCOME TO THE FUTURE!
Welcome to the future!
Quantum SAFE ZONE
Post-quum CRYPTO Adapater
Quantum Crypto!

Kronecker substitution

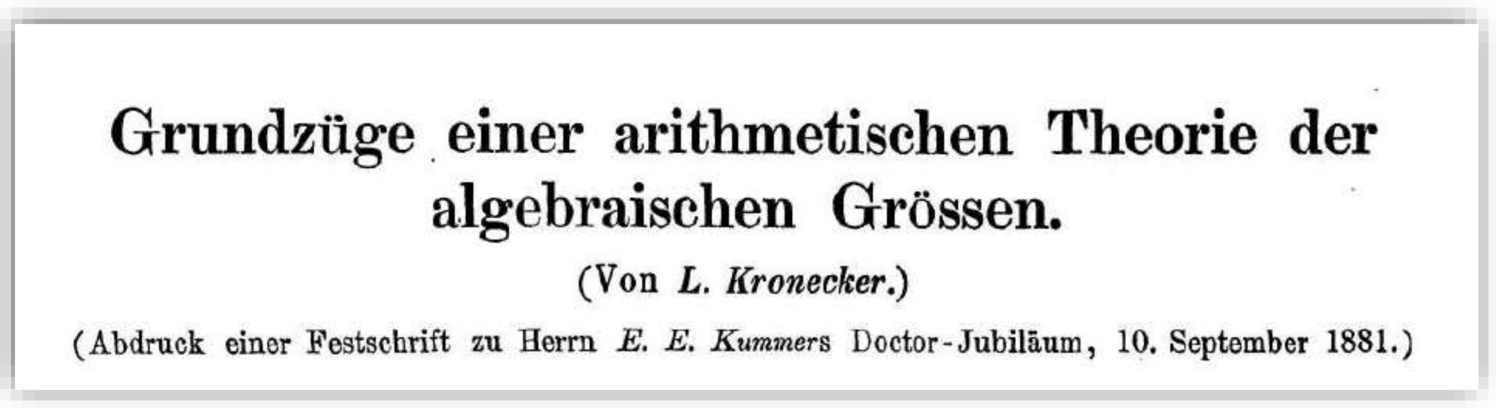
Polynomial domain

$$f = 1 + 2x + 3x^2 + 4x^3$$

$$g = 5 + 6x + 7x^2 + 8x^3$$

$$\times$$

$$fg = \underline{5} + \underline{16}x + \underline{34}x^2 + \underline{60}x^3 + \underline{61}x^4 + \underline{52}x^5 + \underline{32}x^6$$



Grundzüge einer arithmetischen Theorie der algebraischen Grössen.

(Von *L. Kronecker.*)

(Abdruck einer Festschrift zu Herrn *E. E. Kummers* Doctor-Jubiläum, 10. September 1881.)

Kronecker domain (with evaluation point 100)

$$f(100) = 4030201$$

$$g(100) = 8070605$$

$$\times$$

$$fg(100) = \underline{32}\,\underline{52}\,\underline{61}\,\underline{60}\,\underline{34}\,\underline{16}\,\underline{05}$$

Polynomial multiplication techniques

Kronecker evaluation at 2^{32}

Multiplication with a 256-bit multiplier

$$\begin{tikzcd}
\frac{\mathbb{Z}[X]}{(X^{256}+1)} \arrow[r, dashed, "\textit{Kronecker}"] \arrow[d] \arrow[dd, dashed, bend right, "\textit{Nussbaumer}"'] & \frac{\mathbb{Z}}{(2^{8192}+1)} \arrow[r, dashed, "\textit{Schön.-Strassen}"] & \frac{\mathbb{Z}/(2^{544}+1)[X]}{(X^{32}+1)} \\
\frac{\mathbb{Z}[Y]/(Y^{8}+1)[X]}{(X^{32}-Y)} \arrow[r, dashed, "\textit{Kronecker}"] \arrow[d] & \frac{\mathbb{Z}/(2^{256}+1)[X]}{(X^{32}-2^{32})} \arrow[r, dashed, "\textit{Twist}"] & \frac{\mathbb{Z}/(2^{256}+1)[X]}{(X^{32}-1)} \\
\frac{\mathbb{Z}[Y]/(Y^{8}+1)[X]}{(X^{64}-1)} \arrow[r, dashed, "\textit{Kronecker}"] & \frac{\mathbb{Z}/(2^{256}+1)[X]}{(X^{64}-1)} &
\end{tikzcd}$$

Kronecker+

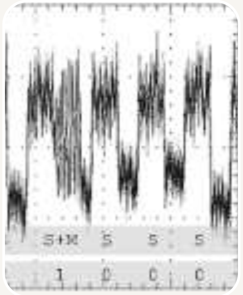
Algorithm	# Muls	# Bits
Kron. + Schoolbook	1024	256
Kron. + Karatsuba	243	256
Kron. + Toom-Cook	63	256
Kron. + Schön.-Strassen	32	544
Nussbaumer + Kron.	64	256
Kronecker+	**32**	**256**

[A] Albrecht, Hanser, Hoeller, Pöppelmann, Virdia, Wallner; Implementing RLWE-based schemes using an RSA co-processor. TCHES 2019

[B] Harvey. Faster polynomial multiplication via multipoint Kronecker substitution. J. of Sym. Comp. 2009.

[C] Bos, Renes, van Vredendaal: Polynomial Multiplication with Contemporary Co-Processors: Beyond Kronecker, Schönhage-Strassen & Nussbaumer. USENIX Security Symposium 2022.

Resistance against physical & logical attacks

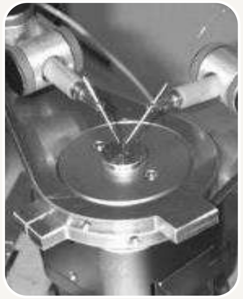


Side-channel attacks

- Power analysis (SPA, DPA)
- Electromagnetic analysis (SEMA, DEMA)
- Timing Analysis
- Photo-emission microscopy (high-end)
- Profiled, unprofiled and ML-assisted variants

Fault injection attacks

- Voltage or clock glitching
- Electromagnetic fault injection (EMFI)
- Body bias injection
- Laser fault injection
- Single and multi-shot scenarios



Invasive attack

- Focused Ion Beam (FIB) modifications
- Micro/Nano-probing of internal signals
- Signal forcing
- Delayering
- Reverse-engineering

Embedded cryptography and implementation attacks

Attacks

Deep understanding in both academia and industry.

⚠ ✓

Classic Cryptography

AES 3DES
DSA… ECDSA
RSA ECC

⚠ ✓

Countermeasures

Practically secure and certified implementations.

Embedded cryptography and implementation attacks

Attacks

Deep understanding in both academia and industry.

⚠ ✓

Classic Cryptography

AES 3DES
DSA… ECDSA
RSA ECC

⚠ ✓

Countermeasures

Practically secure and certified implementations.

Post-Quantum Cryptography

Active research area resulting in increasingly powerful attacks.

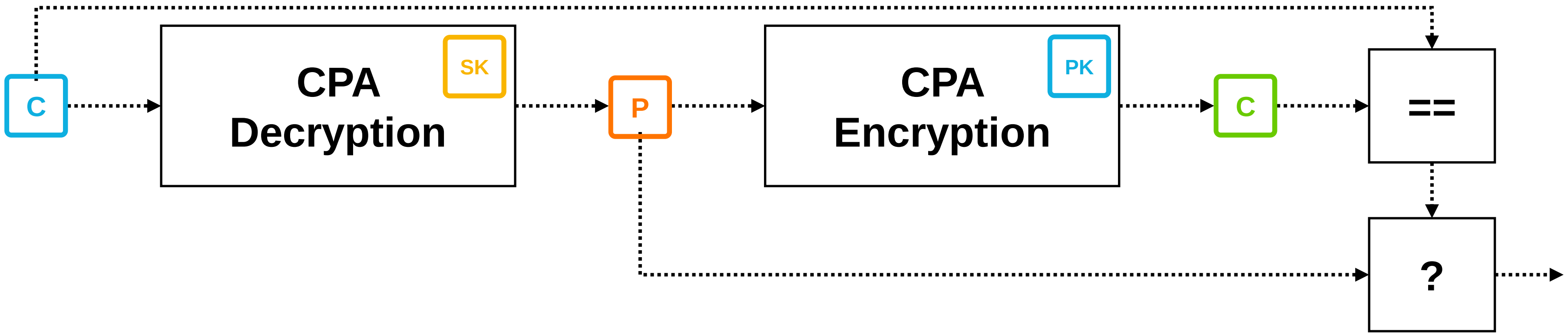
⚠ ?

Kyber
Dilithium …
SPHINCS+
XMSS

⚠ ?

Early stage of academic research. Limited industrial results.

Fujisaki Okamoto transform



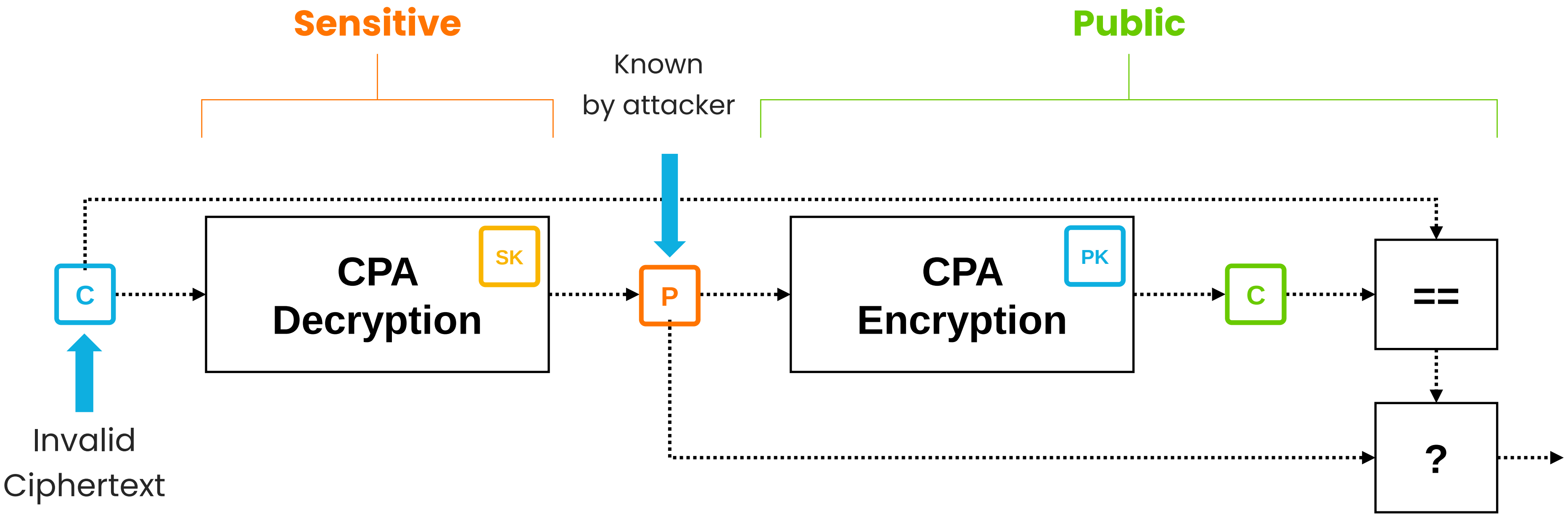
Transform a scheme which achieves **IND-CPA** (“chosen plaintext attack”) security to reach **IND-CCA** (“indistinguishability against chosen-ciphertext attacks”) security

Fujisaki, E. and Okamoto T., Secure integration of asymmetric and symmetric encryption schemes, CRYPTO 1999 and JoC 2013

The SCA Problem of the FO-Transform

Attack 1: Chosen Plaintext

- Attacker inputs only valid ciphertexts
- Attack focuses on **CPA Decryption**, everything after (and including) P is public
- Only need to protect **CPA Decryption**

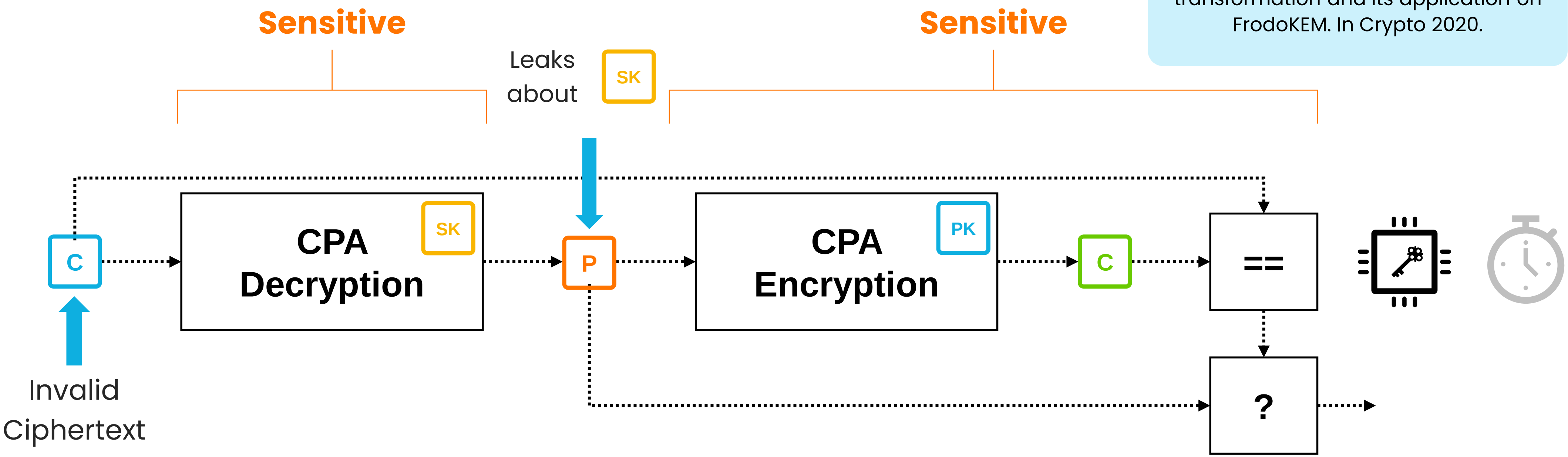


The SCA Problem of the FO-Transform

Attack 2: Chosen Ciphertext

- Attacker inputs specially-crafted invalid ciphertexts
- Attack focuses on **CPA Decryption** + everything after (and including) P is potentially sensitive
- Potentially all (or most) modules need to be hardened

Guo, Johansson, Nilsson. A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. In Crypto 2020.



From Theory to practice: Secure implementations (NXP PQC Team)

Only with carefully managed maximum number of issued signatures

First completely masked implementation of Kyber / FIPS 203 !

Year	Venue	FIPS 203	FIPS 204	Title
2021	TCHES	🟩		Masking Kyber: First- and Higher-Order Implementations
2021	RWC	🟩		Post-Quantum Crypto: The Embedded Challenge
2022	TCHES	🟩	🟦	Post-Quantum Authenticated Encryption against Chosen-Ciphertext SCA
2022	RWC	🟩		Surviving the FO-calypse: Securing PQC Implementations in Practice
2023	TCHES		🟦	From MLWE to RLWE: A Differential Fault Attack on Randomized & Deterministic Dilithium
2023	TCHES		🟦	Protecting Dilithium Against Leakage Revisited Sensitivity Analysis
2024	RWC		🟦	Lessons Learning from Protecting CRYSTALS-Dilithium
2024	TCHES		🟦	Exploiting Small-Norm Polynomial Multiplication with Physical Attacks
2024	RWC	🟩	🟦	Challenges of Migration to PQ Secure Embedded Systems

Completely masked implementation of Dilithium / FIPS 204 !

[A] Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process

Industrial & IoT

i.MX 94 Family of Applications Processors Delivers Safe and Secure Industrial and Automotive Connectivity with Real-Time Control

Samples available in 1H, 2025

Security

- **First NXP apps processor supporting Post-Quantum Cryptography**
- EdgeLock Secure Enclave with Cyber Resilience Recovery Module

Including
✓ Secure boot,
✓ Secure update
✓ Secure debug
of the processor based on PQC

Customer products in-the-field for 10-15 years, PQC a wanted feature!

NXP S32G2 vehicle network processor with PQC integration

Our target platform: S32G274A

- 3 Lockstep Arm® Cortex®-M7 Microcontrollers
- 4 Cluster Lockstep Cortex-A53 Microprocessors
- 8 MB of System RAM
- Network Accelerators (LLCE/PFE)
- **Hardware Security Engine (HSE)**
- ASIL D Functional Safety Support

Post-Quantum Crypto

- Integrate PQC secure signature verification
- Protection against Fault Attacks
- Enable PQC secure boot
- Secure Over-the-Air (OTA) updates
- Secure vehicle and driver data

www.nxp.com/S32G2

SAFE ASSURE™ by NXP

EDGELOCK™ ASSURANCE by NXP

Benchmarks for authentication of FW signature on the S32G2

Alg.	Size		Performance (ms)			
			1 KB		128 KB	
	PK	Sig.	Inst.	Boot	Inst.	Boot
RSA 4K	512	512	2.6	0.0	2.7	0.2
ECDSA-p256	64	64	6.2	0.0	6.4	0.2
Dilithium-3	1952	3293	16.7	0.0	16.9	0.2

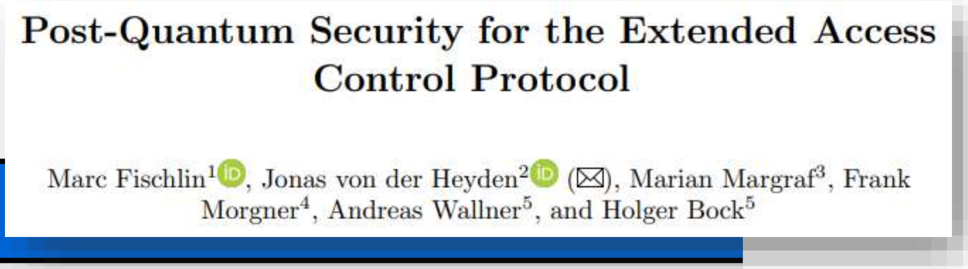
Demonstrator only, further optimizations are possible (such as hardware accelerated SHA-3)

Signature verification only required once for installation!

During boot the signature verification can be replaced with a check of the Reference Proof of Authenticity

Bos, Carlson, Renes, Rotaru, Sprenkels, Waters;
Post-Quantum Secure Boot on Vehicle Network Processors. Embedded Security in Cars. Escar 2022

eMRTD PQC Migration



Post-Quantum Security for the Extended Access Control Protocol

Marc Fischlin[1], Jonas von der Heyden[2] (✉), Marian Margraf[3], Frank Morgner[4], Andreas Wallner[5], and Holger Bock[5]

Protocol	Goal
Passive Authentication	Authenticate to check integrity of the data stored in the chip
PACE	Set up a communication channel between chip and terminal
Terminal Authentication	Authorize terminal to view sensitive biometrics
Chip Authentication	Prevent sensitive data copy and prove chip authentication

Extended Access Control (EAC)

NXP Recommendations (ICAO)

- Migrate country signing certificates with highest priority
- CSCA / CVCA: **SP 800-208**
- Document signers: **ML-DSA**
- Consider ML-KEM based EAC to avoid variable signing time and decrease key size
- PACE migration lower priority

Working with Darmstadt University of Applied Sciences on a PQC PAKE proof of concept for SmartMX P71
See: https://eprint.iacr.org/2025/812

Many more activities ongoing with GP, GSMA (eSIM), TCG, Javacard, etc. !

Summary

Migration recommended & requested by ecosystem

- Harvest-now, decrypt-later
- Software/firmware signing
- More use cases in a phased / hybrid migration!

Many practical challenges & solutions

- Algorithm design (ML-KEM)
- Low-memory implementations
- Protection against side-channel analysis (SCA) and Fault Injection (FI)
- Hardware acceleration (SHA-3)

First Post-Quantum Cryptography standards ready for adoption (ML-KEM, ML-DSA, SLH-DSA, LMS/XMSS)

Exciting times to work in cryptography!

Get in touch!

Joppe W. Bos

joppe.bos@nxp.com

nxp.com

NXP

Brighter Together