



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

| Public | NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2025 NXP B.V. Vers





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

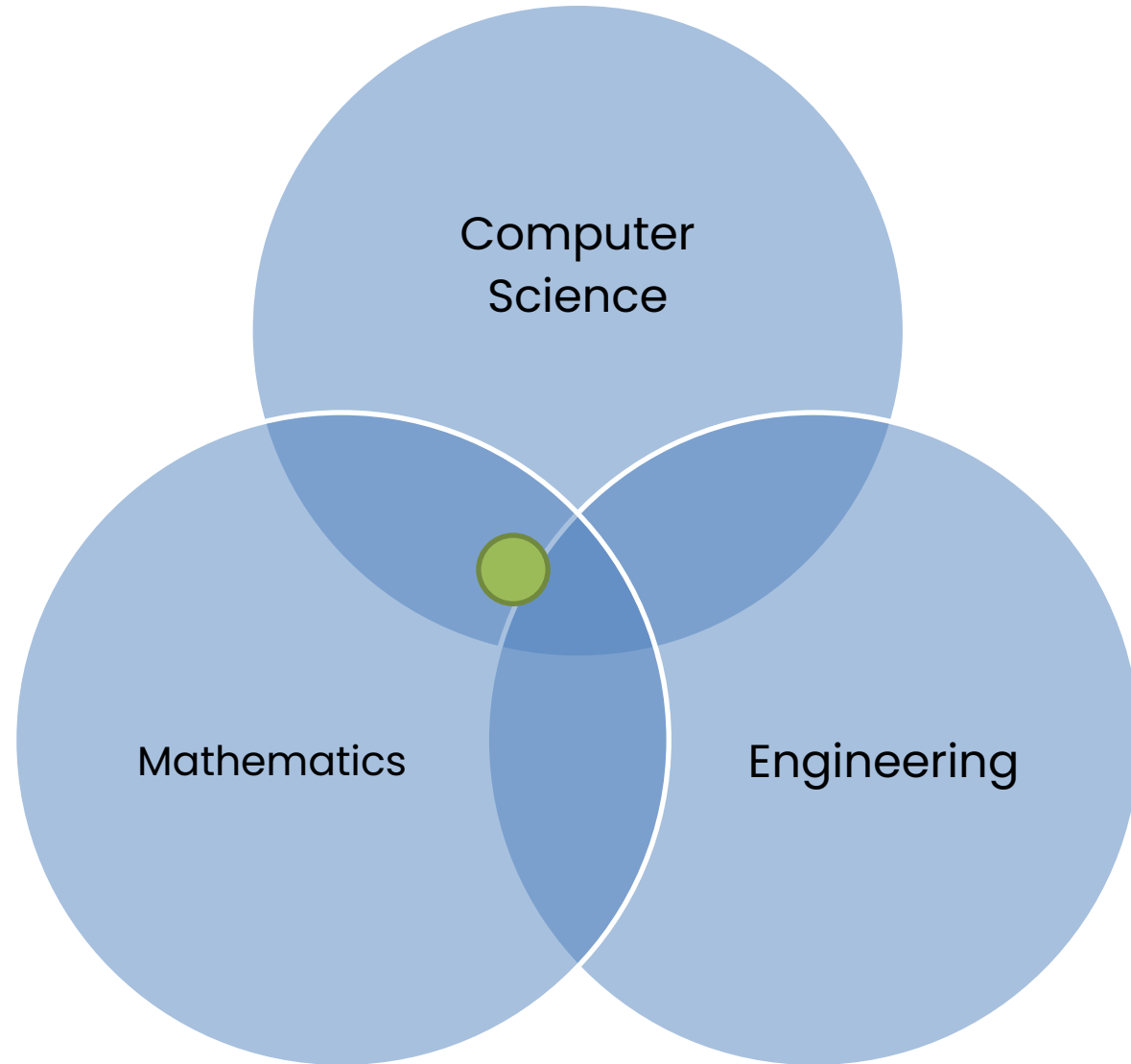
WHOAMI

- 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





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

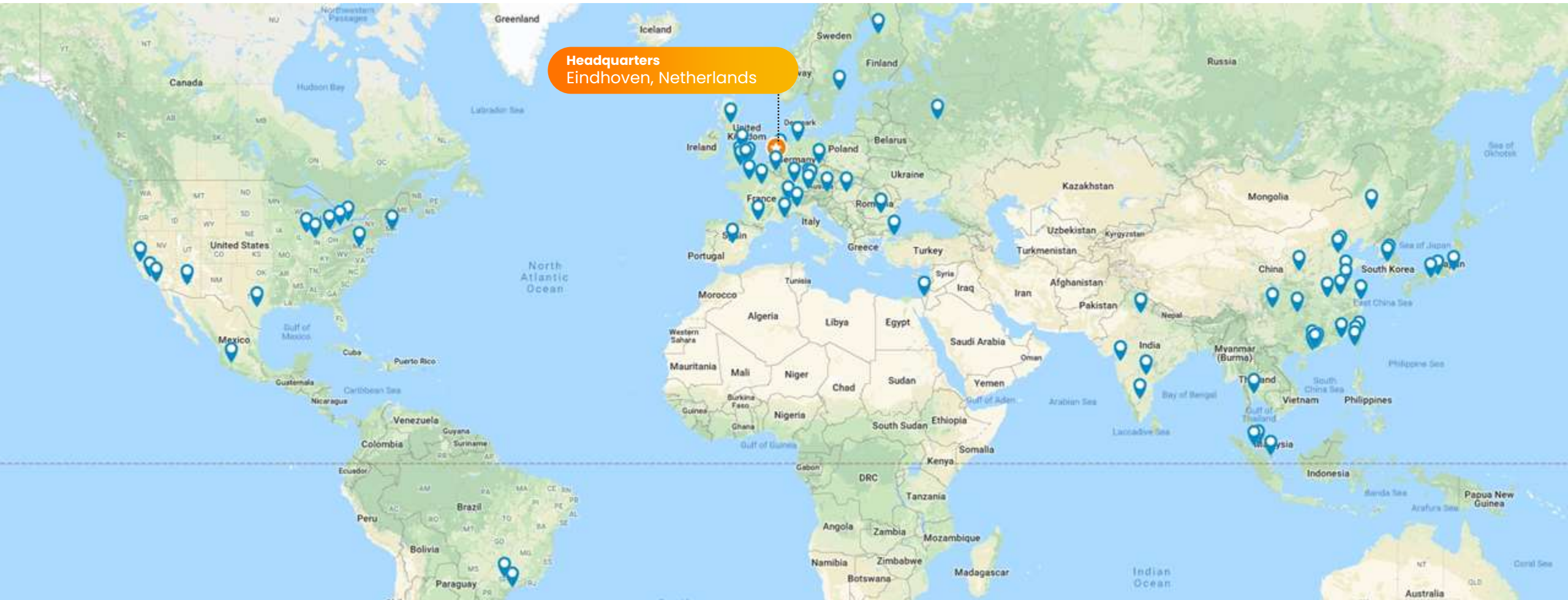


Communication
Infrastructure



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



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

10's to 100's Millions

Network Edge
Network computing



Billions

Application Edge
IoT end points



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”

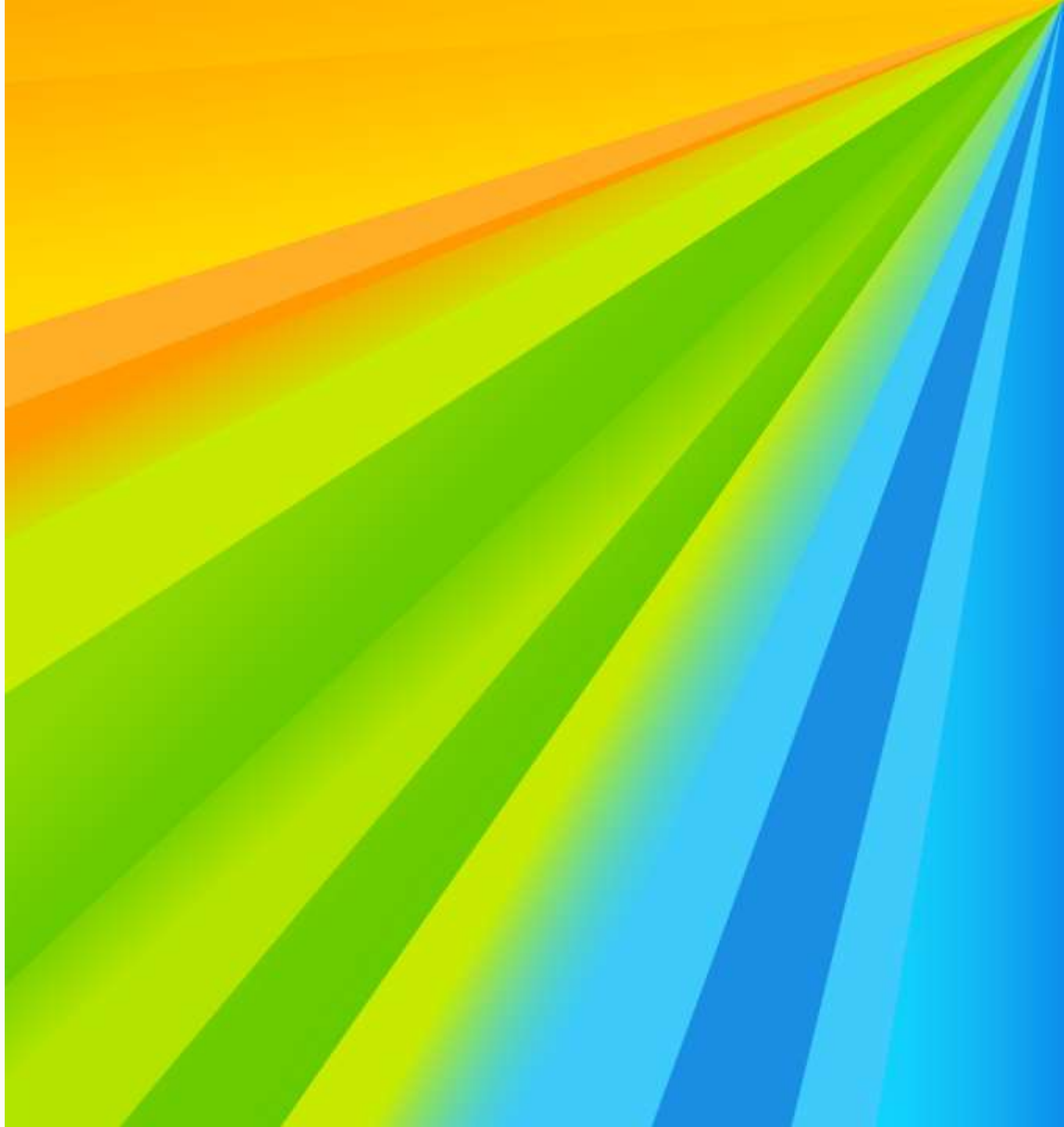
Increase reliability for consumers

Ensure security and privacy

Led by global brands and 200+ companies



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

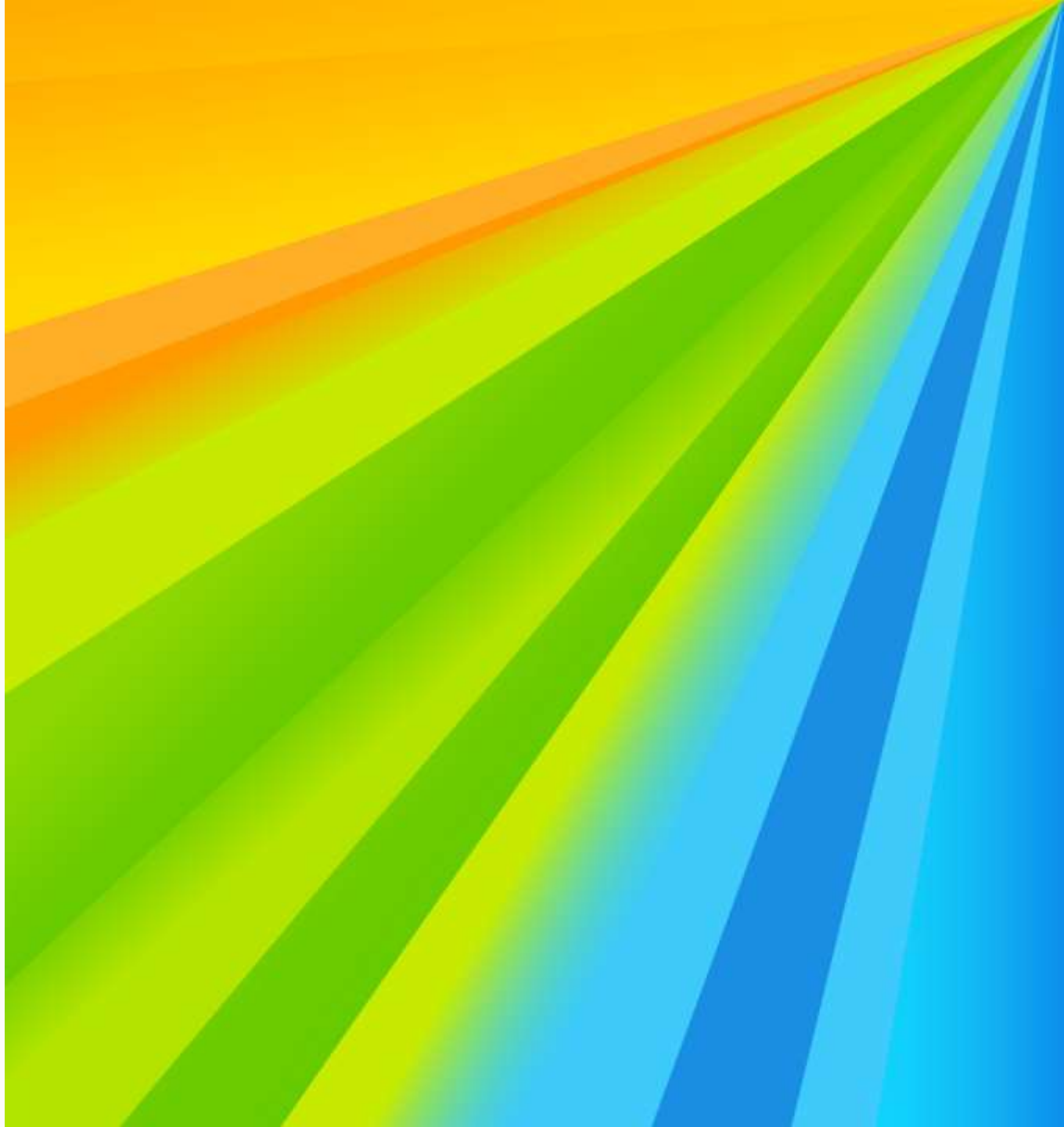
RSA $\rightarrow (\mathbb{Z}/N\mathbb{Z})^\times \rightarrow$ integers $[1, 2, \dots, N - 1]$ which are co-prime to N

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

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

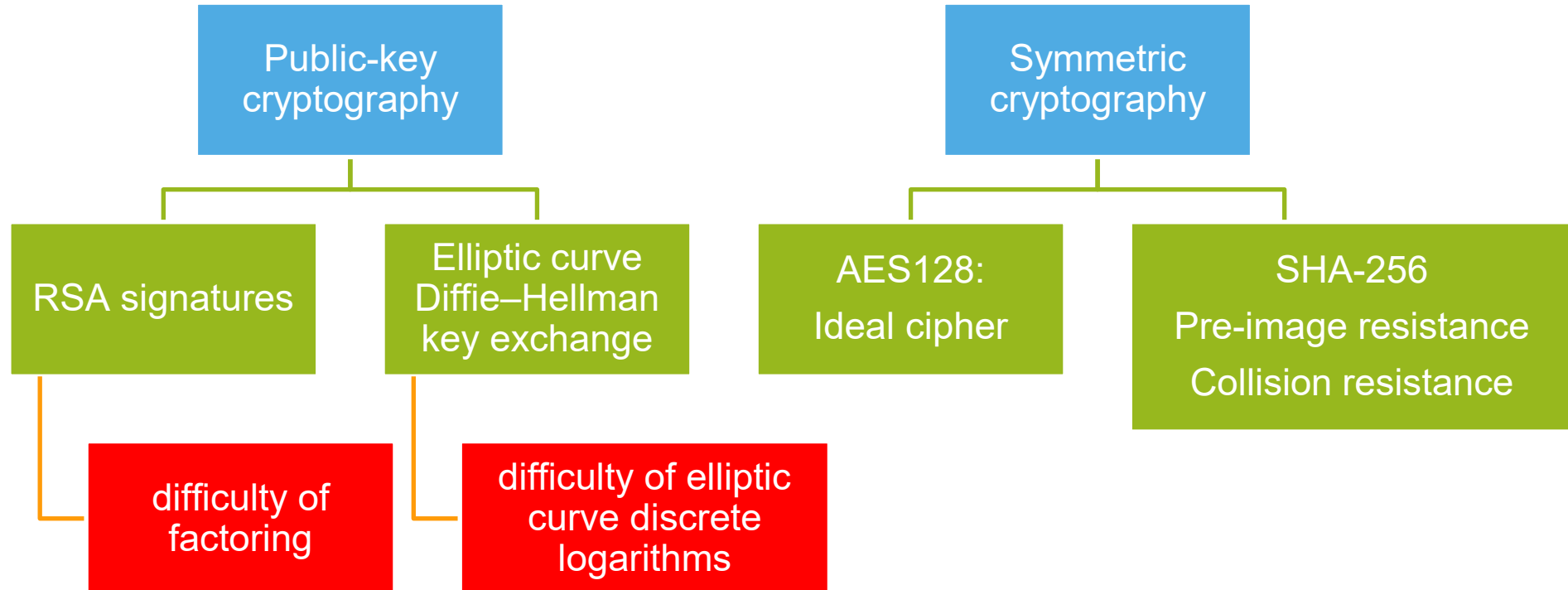
Application	Encryption Scheme, Signature Scheme, Identification Scheme, etc.		
Cryptosystem	DSA, ElGamal, Schnorr, etc.		RSA, Rabin, etc.
Computational Problem	The Discrete Logarithm Problem in a Group of prime Order		The Factoring Problem
Algebraic Structure	The multiplicative group of integers modulo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

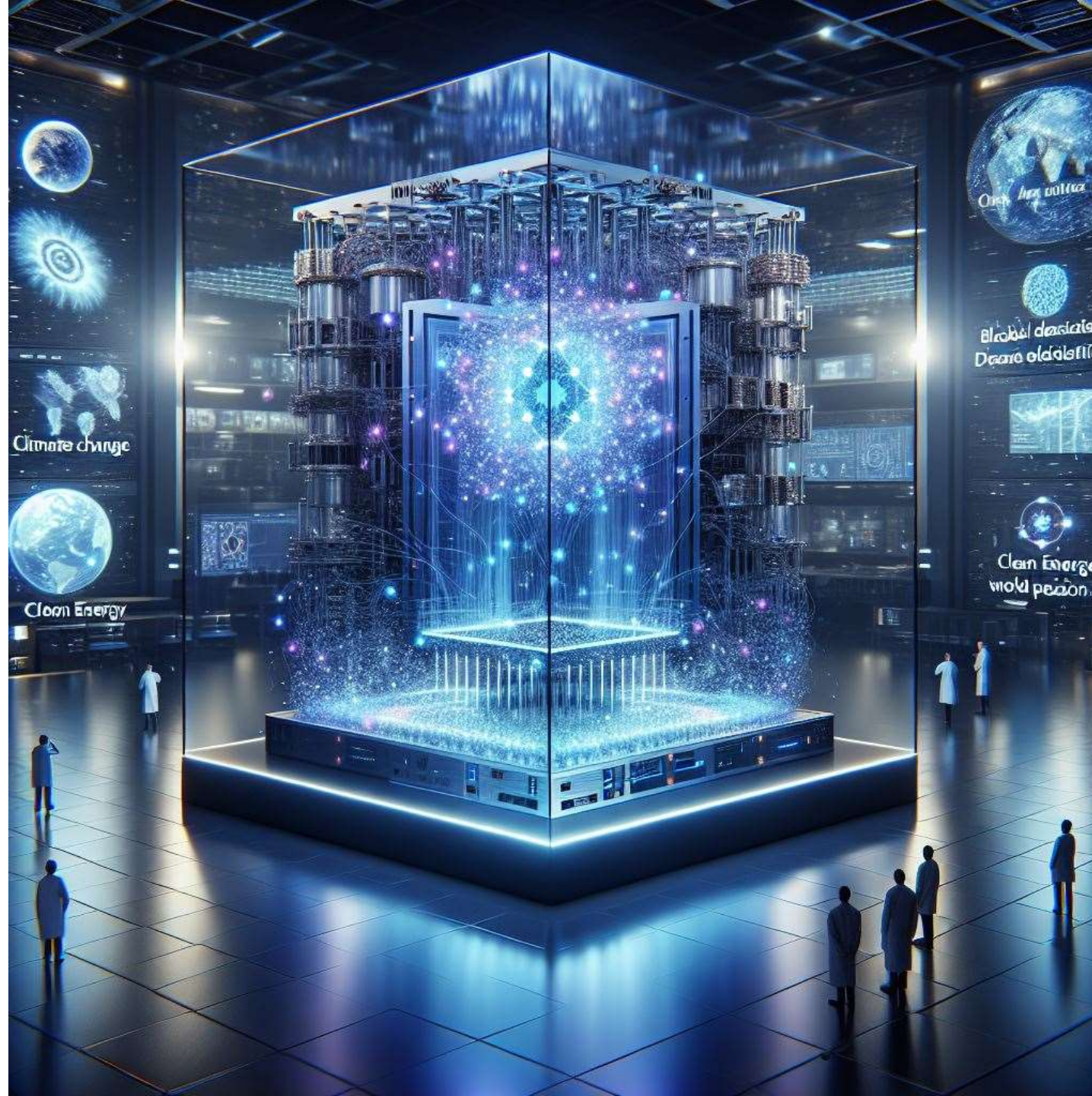
Post-Quantum Cryptography



Contemporary Cryptography

TLS-ECDHE-RSA-AES128-GCM-SHA256





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



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.

quantum
new chip was
performance.



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

Quantum error correction below the surface code threshold

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

SHARE



in

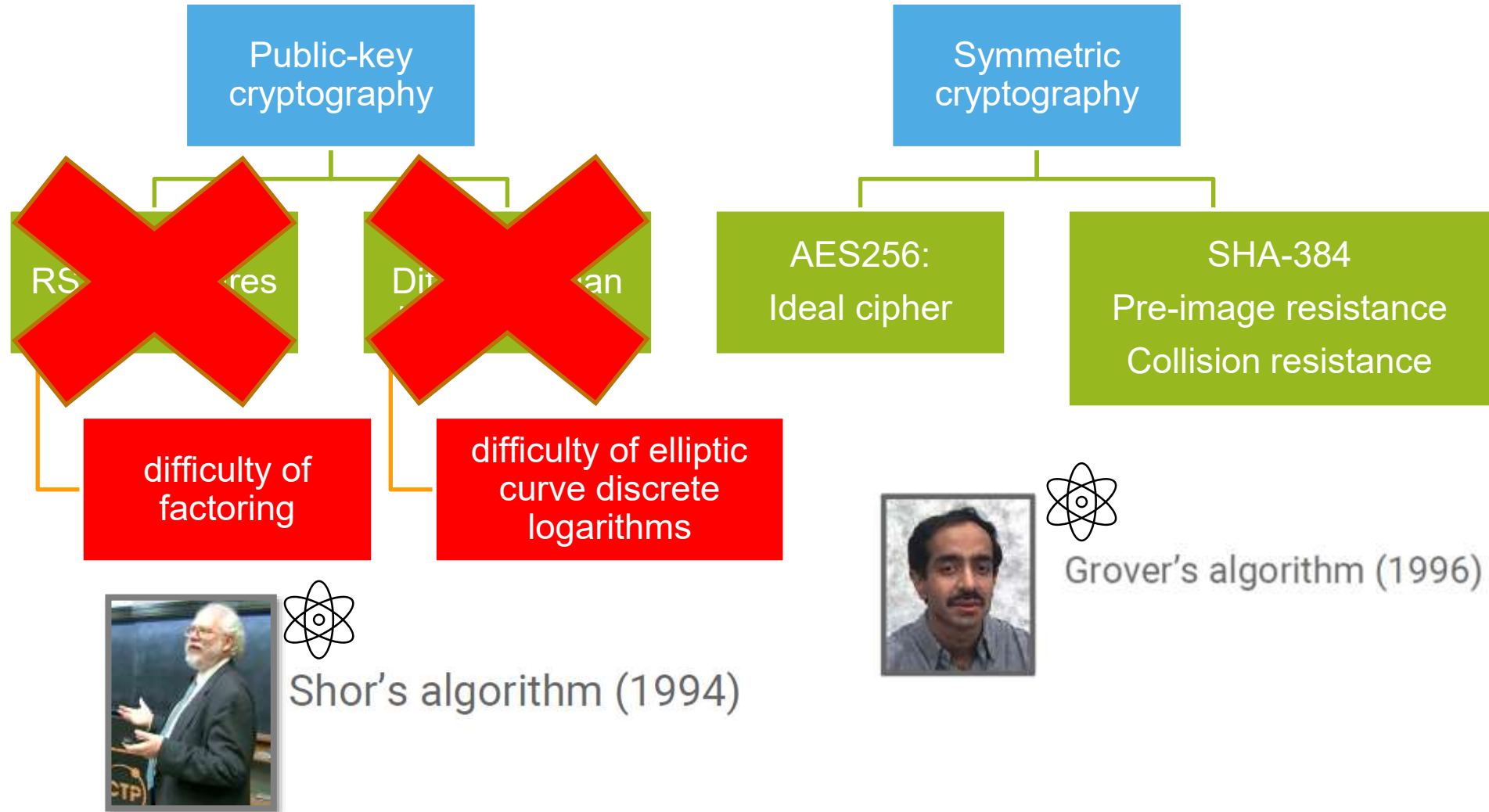


- NXP, eleQtron and ParityQC reveal their first quantum computing demonstrator for the DLR Quantum Computing Initiative.
- It was commissioned by the DLR Quantum Computing Initiative (DLR QCI) to expand the quantum expertise of its partners from research and industry

Contemporary cryptography

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

“Double” the key sizes



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

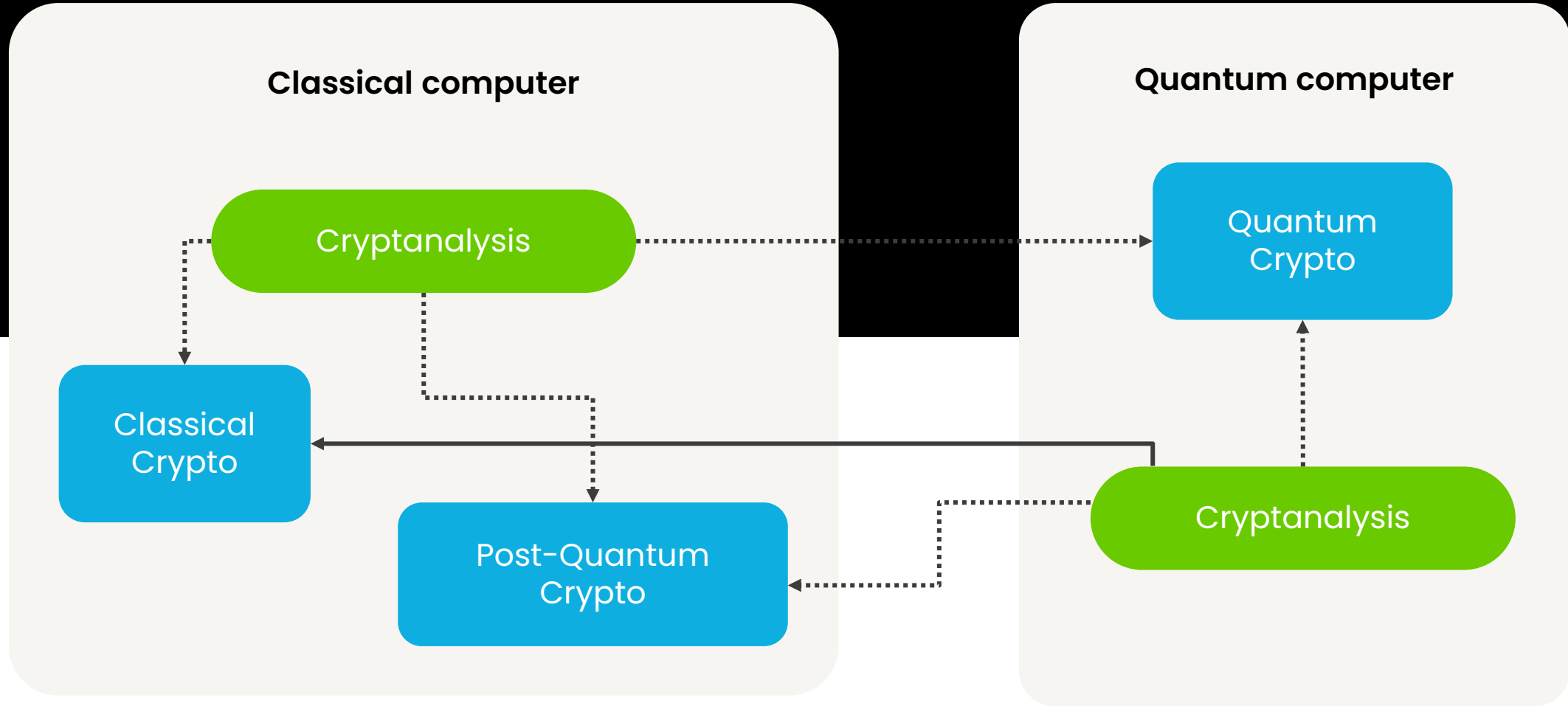


The integrity of blockchains

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

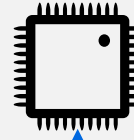


NIST



Crypto Agility

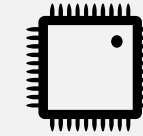
PQC RoT



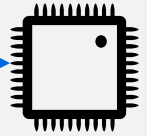
Secure Updates



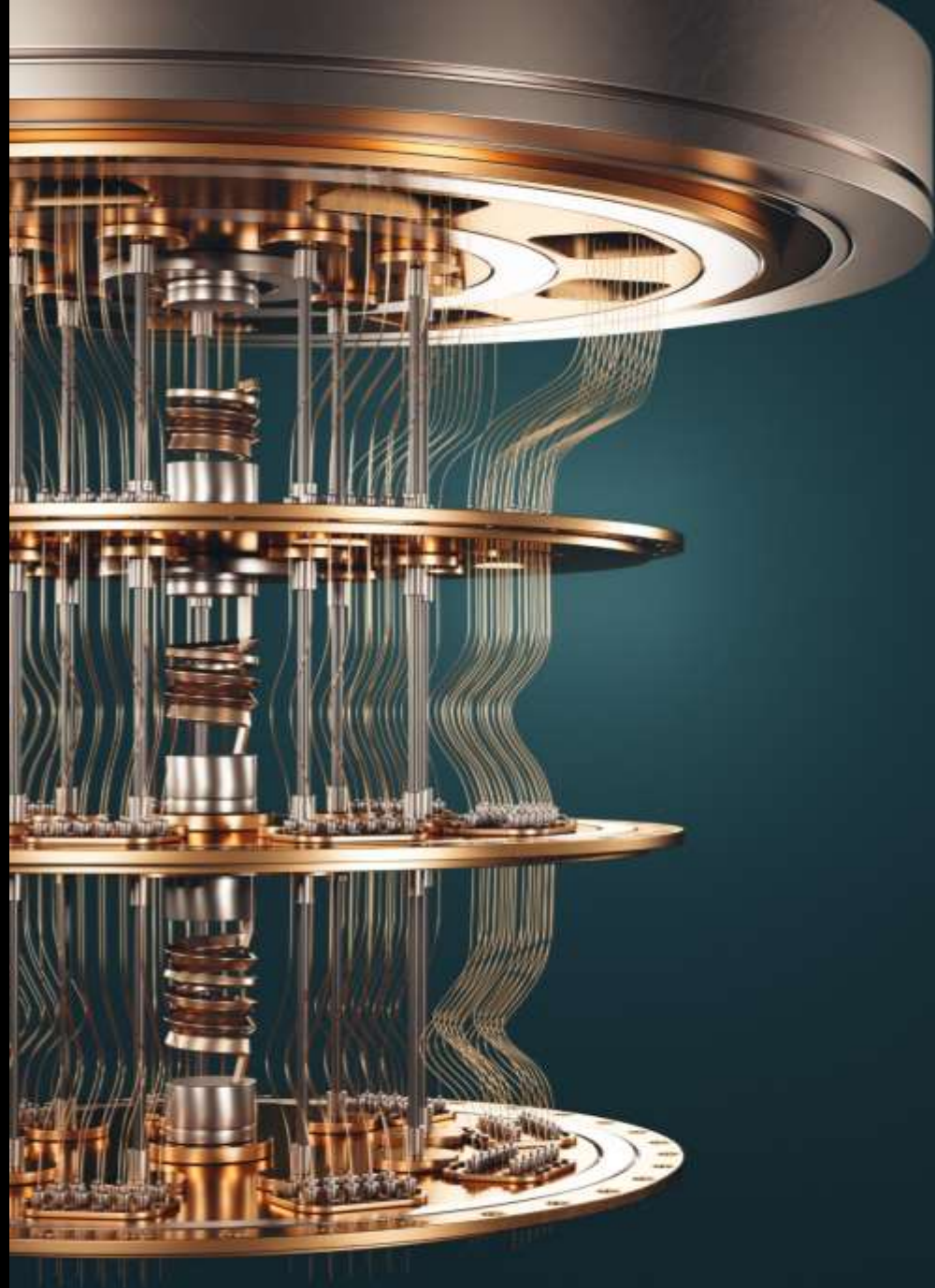
Store Now Decrypt Later



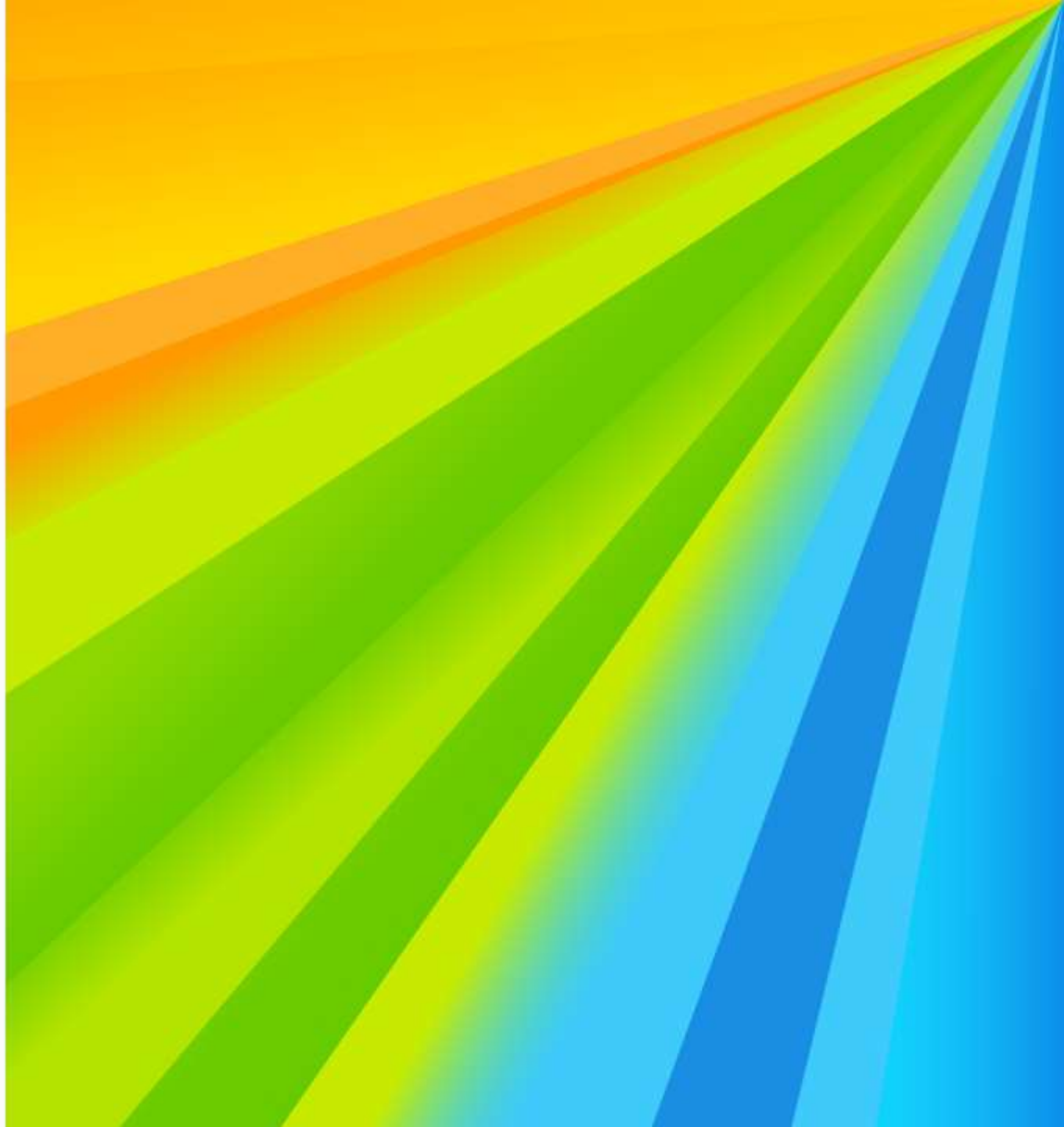
TLS 1.3



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



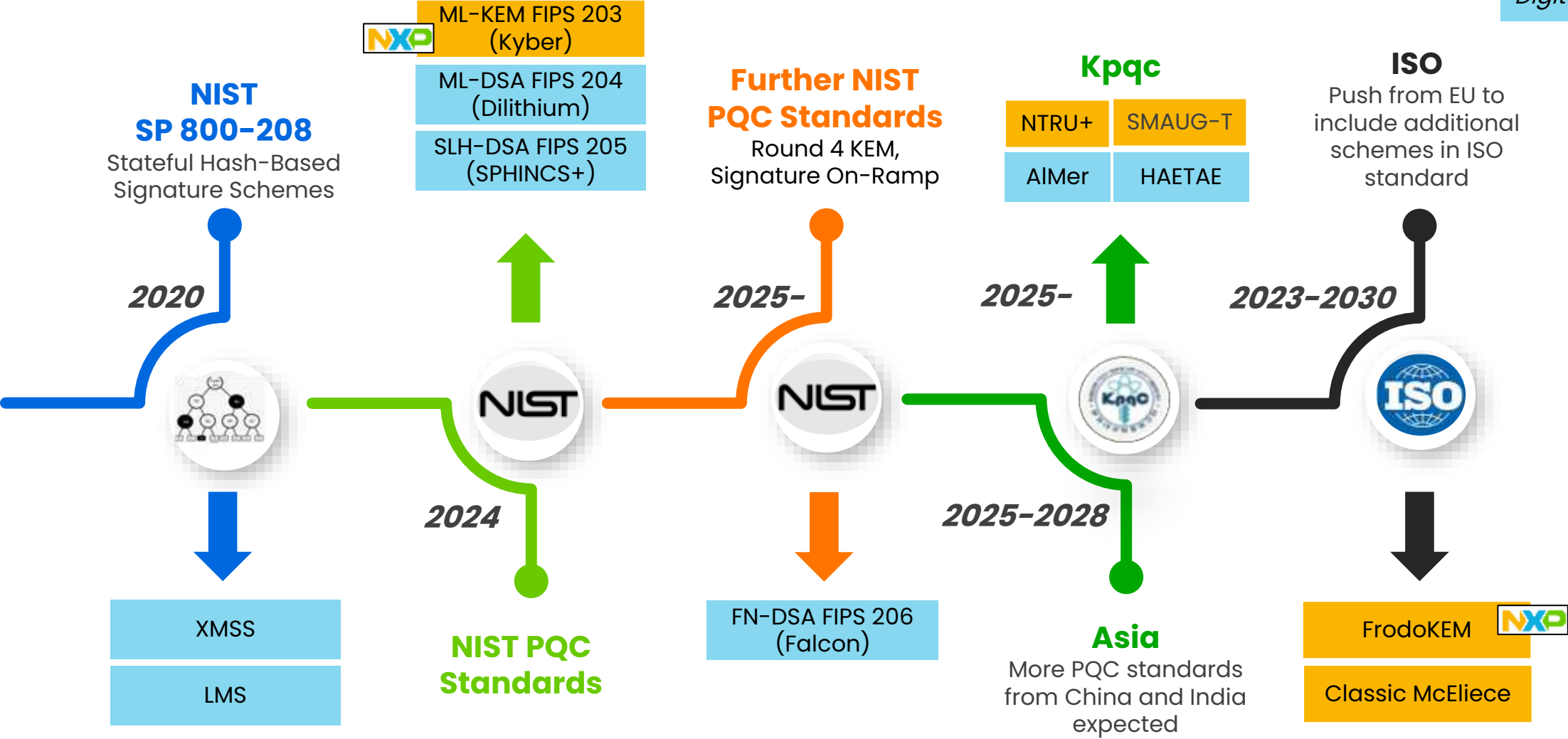
PQC Standards



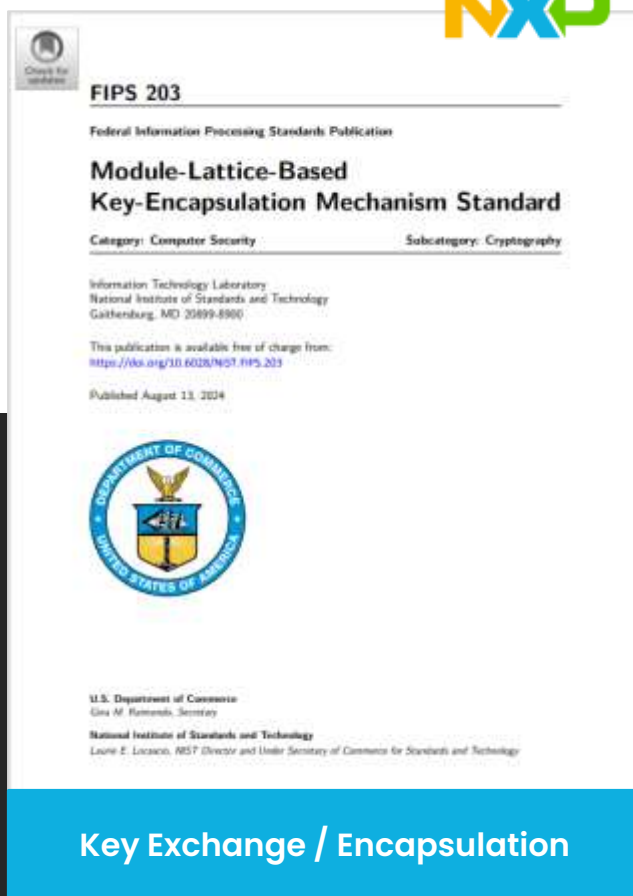
PQC standards

Key Exchange

Digital Signature



New algorithms and standards



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)

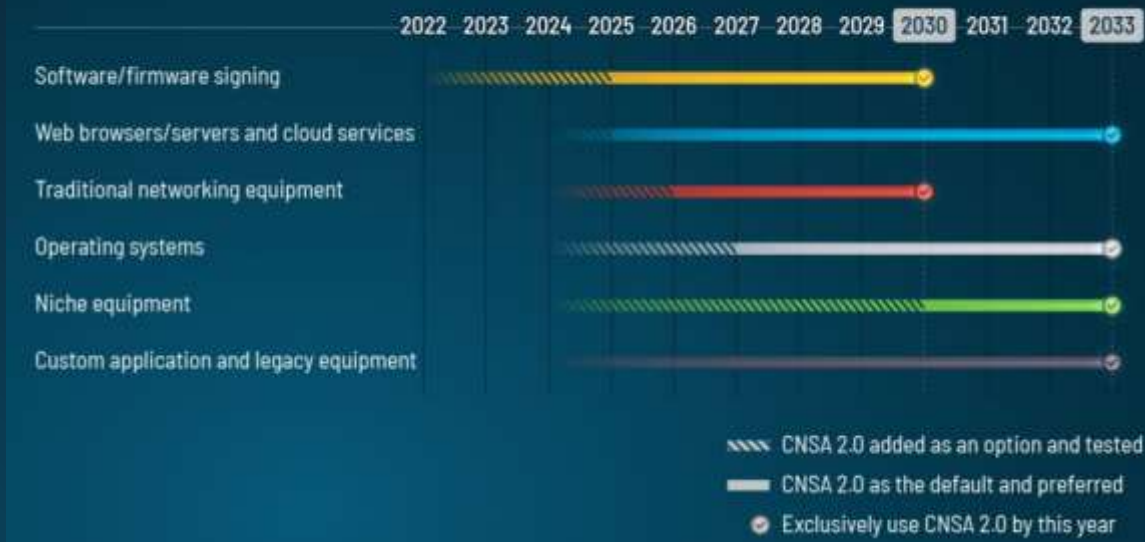
- [BSI first recommendation](#) (English)
- [BSI considerations](#) (German)
- Expectation is that beginning of 2030s, a relevant quantum computer is available to be a threat for high-secure applications
- “QKD is only suitable for specific use cases”



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

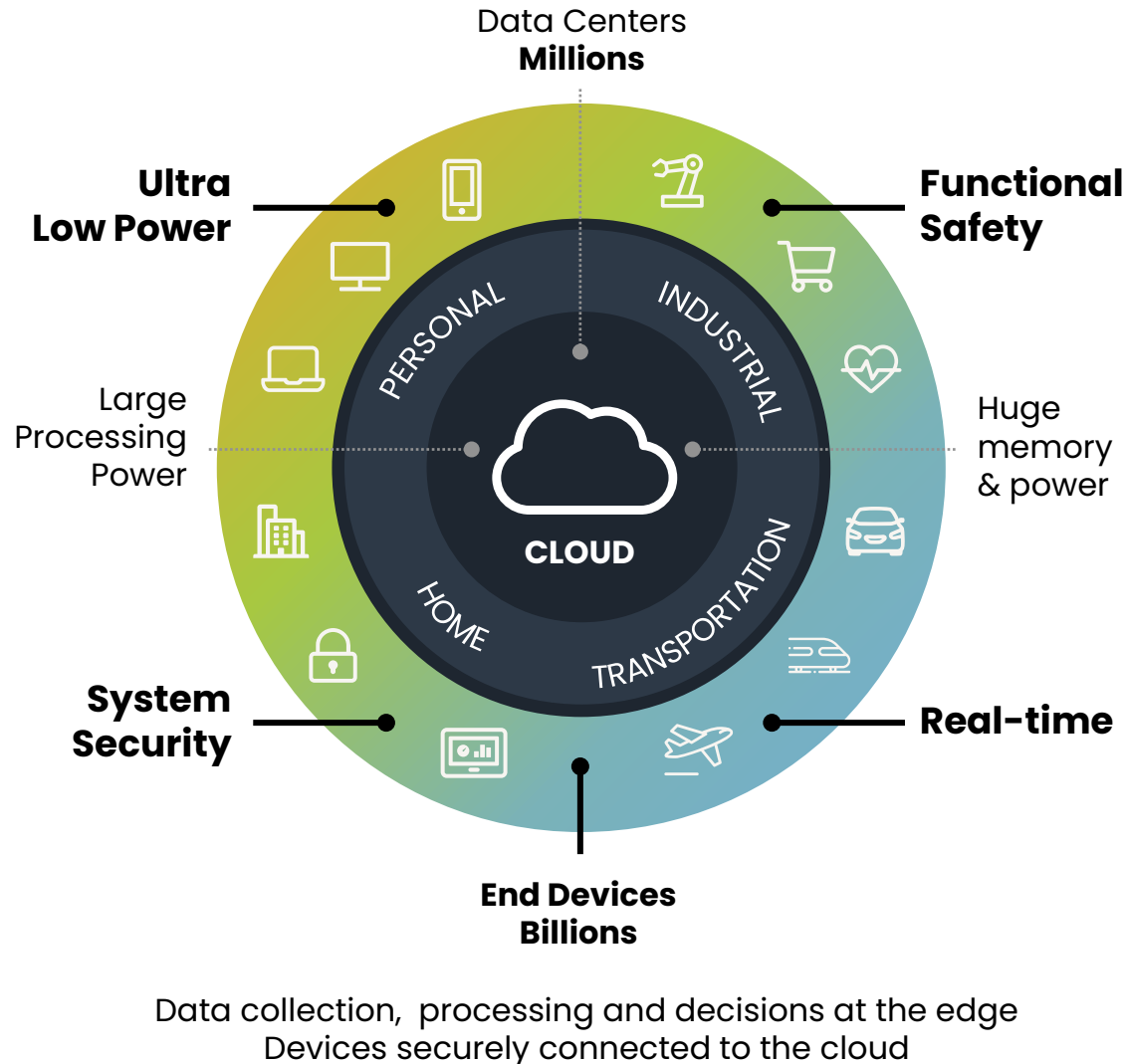
CNSA 2.0 Timeline



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
	≥ 128 bits of security strength	Disallowed after 2035
Elliptic Curve DH and MQC [SP80056A]	112 bits of security strength	Deprecated after 2030
	≥ 128 bits of security strength	Disallowed after 2035
RSA [SP80056B]	112 bits of security strength	Deprecated after 2030
	≥ 128 bits of security strength	Disallowed after 2035

Impact PQC on our eco-system



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!

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

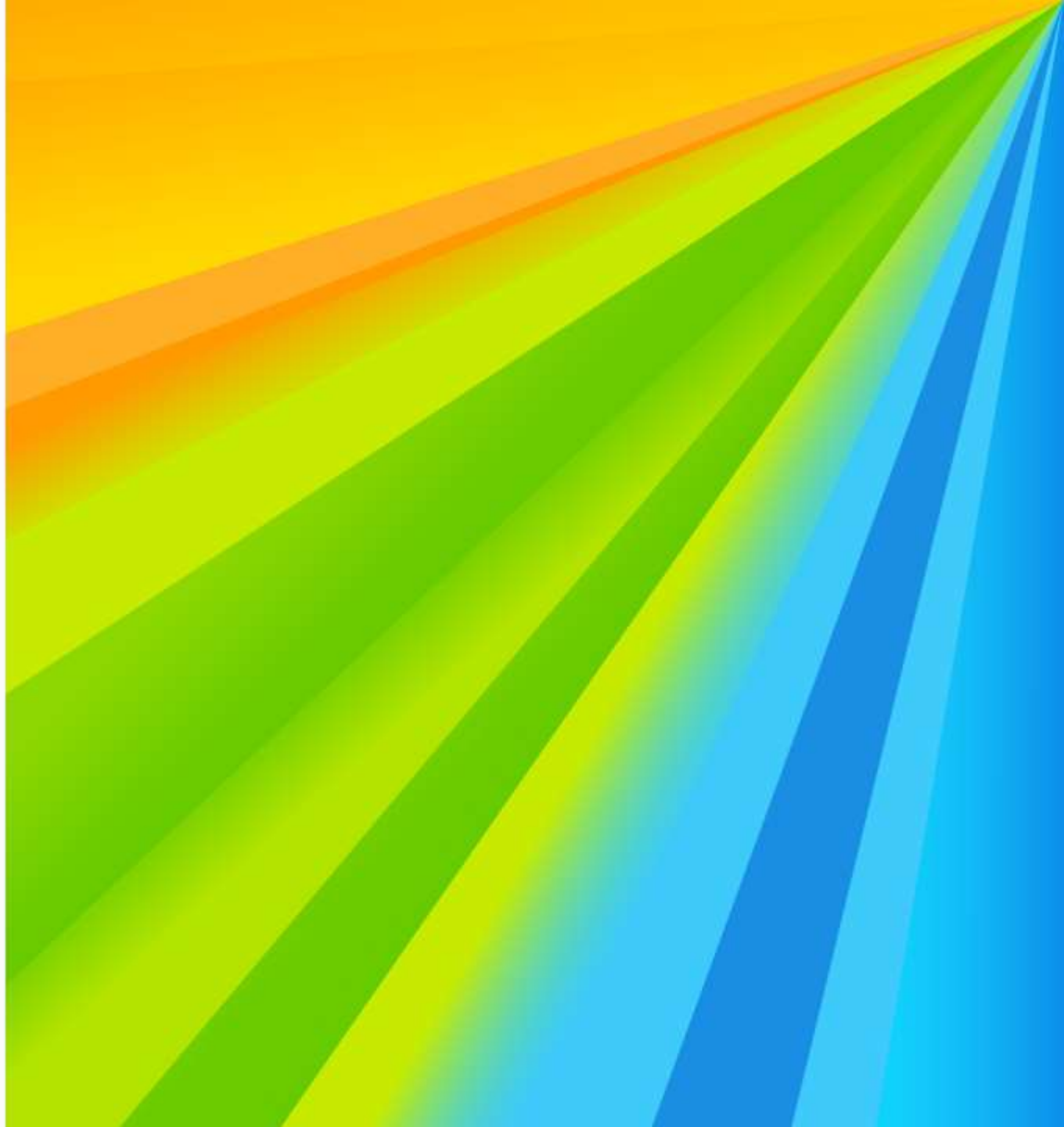


Readers



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**

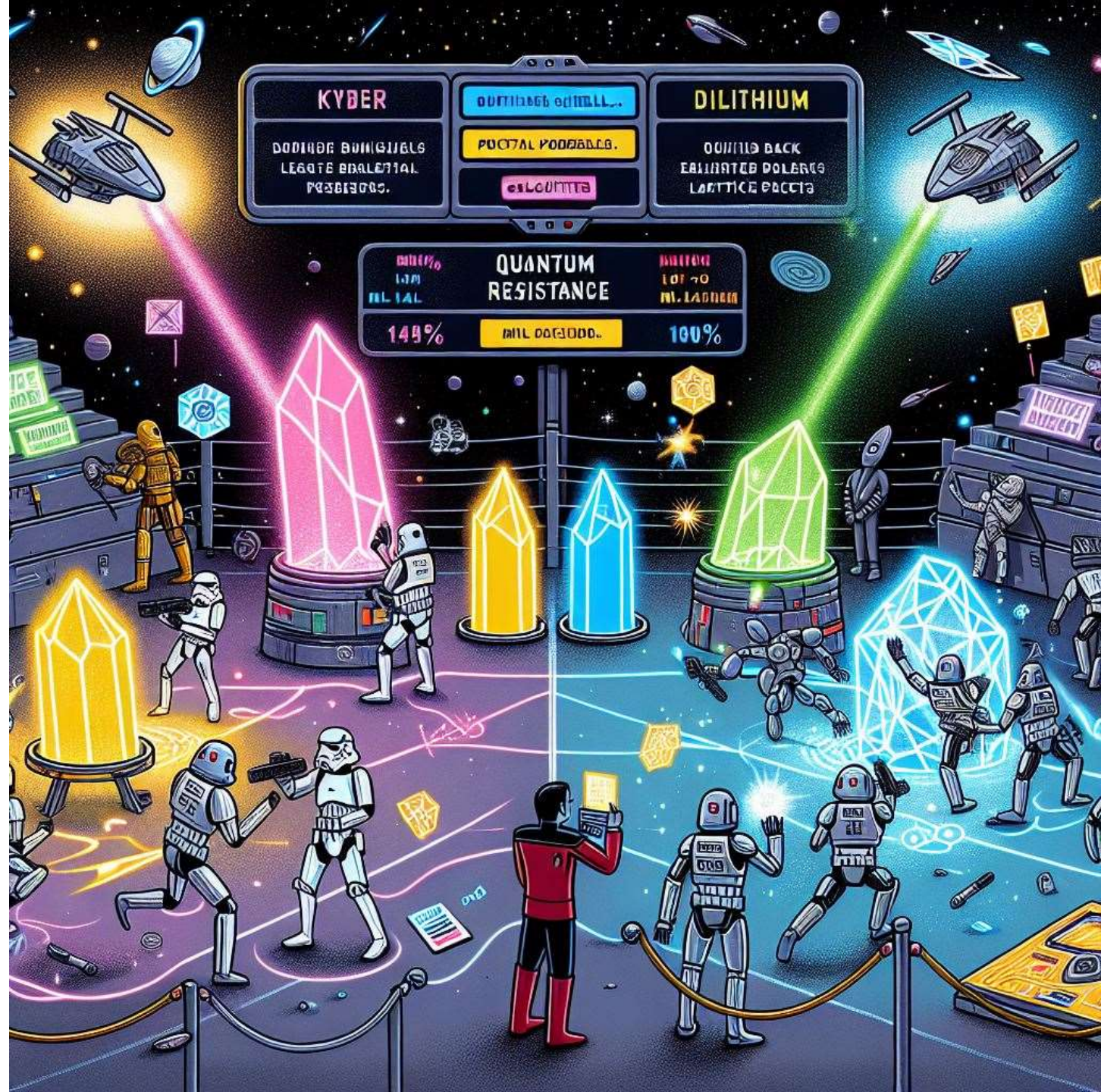
Theory: same building blocks

- Module Learning with Errors
- Number-Theoretic Transformations

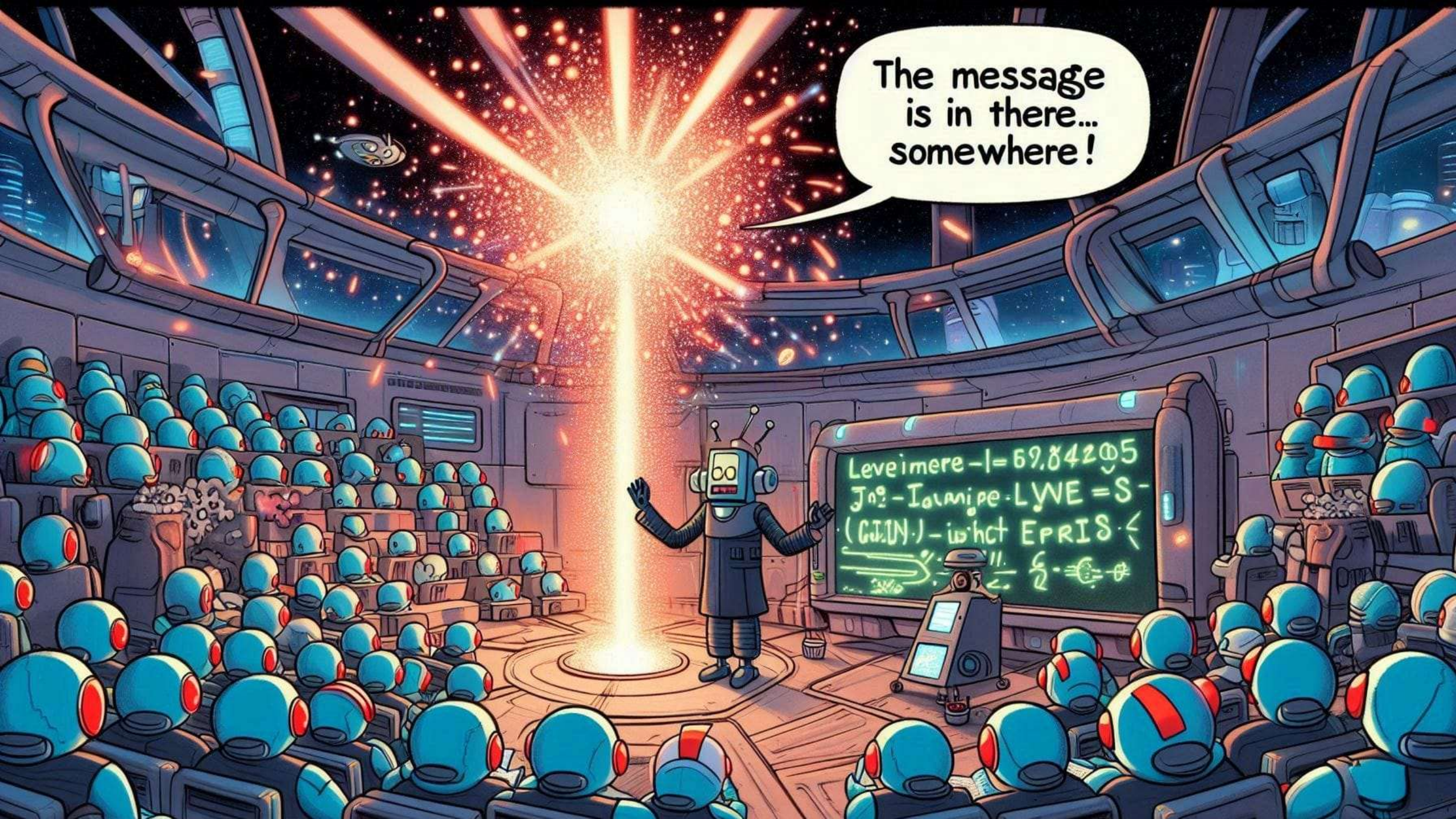
Many new techniques to deal with!

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!



Solving systems of linear equations

$$\begin{matrix} \mathbb{Z}_{13}^{7 \times 4} \\ \begin{array}{|c|c|c|c|} \hline 4 & 1 & 11 & 10 \\ \hline 5 & 5 & 9 & 5 \\ \hline 3 & 9 & 0 & 10 \\ \hline 1 & 3 & 3 & 2 \\ \hline 12 & 7 & 3 & 4 \\ \hline 6 & 5 & 11 & 4 \\ \hline 3 & 3 & 5 & 0 \\ \hline \end{array} \end{matrix} \times \begin{matrix} \text{secret} \\ \mathbb{Z}_{13}^{4 \times 1} \\ \begin{array}{|c|} \hline \\ \hline \\ \hline \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbb{Z}_{13}^{7 \times 1} \\ \begin{array}{|c|} \hline 4 \\ \hline 8 \\ \hline 1 \\ \hline 10 \\ \hline 4 \\ \hline 12 \\ \hline 9 \\ \hline \end{array} \end{matrix}$$

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

Solving systems of linear equations

$$\begin{matrix} \mathbb{Z}_{13}^{7 \times 4} \\ \begin{array}{|c|c|c|c|} \hline 4 & 1 & 11 & 10 \\ \hline 5 & 5 & 9 & 5 \\ \hline 3 & 9 & 0 & 10 \\ \hline 1 & 3 & 3 & 2 \\ \hline 12 & 7 & 3 & 4 \\ \hline 6 & 5 & 11 & 4 \\ \hline 3 & 3 & 5 & 0 \\ \hline \end{array} \end{matrix} \times \begin{matrix} \text{secret} \\ \mathbb{Z}_{13}^{4 \times 1} \\ \begin{array}{|c|} \hline 6 \\ \hline 9 \\ \hline 11 \\ \hline 11 \\ \hline \end{array} \end{matrix} = \begin{matrix} \mathbb{Z}_{13}^{7 \times 1} \\ \begin{array}{|c|} \hline 4 \\ \hline 8 \\ \hline 1 \\ \hline 10 \\ \hline 4 \\ \hline 12 \\ \hline 9 \\ \hline \end{array} \end{matrix}$$

Easily solved using
Gaussian elimination
(Linear Algebra 101)

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

Learning with errors problem

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

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

\times

secret
 $\mathbb{Z}_{13}^{4 \times 1}$

6
9
11
11

$+$

small noise
 $\mathbb{Z}_{13}^{7 \times 1}$

0
-1
1
1
1
0
-1

$=$

$\mathbb{Z}_{13}^{7 \times 1}$

4
7
2
11
5
12
8

Learning with errors problem

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

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

\times

secret
 $\mathbb{Z}_{13}^{4 \times 1}$

$+$

small noise
 $\mathbb{Z}_{13}^{7 \times 1}$

$=$

$\mathbb{Z}_{13}^{7 \times 1}$

4
7
2
11
5
12
8

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

752

$$752 \times 8 \times 15 \text{ bits} = 11 \text{ KiB}$$

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 $-x \bmod 13$.

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 $-x \bmod 13$ ($\rightarrow \mathbb{Z}_{13}[x]/\langle x^4 + 1 \rangle$)

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

×

$$\text{secret}$$

secret

+

$$\text{small noise}$$

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)	$\frac{1}{n}\mathbf{R}$	$\{x \in \mathbf{K} : \text{Tr}(xR) \subseteq \mathbf{Z}\}$
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 $R_q = \mathbb{Z}_q[x]/(x^n+1)$

Alice

secret:

random "small" s, e in R_q

Bob


secret:

random "small" s', e' in R_q

$$b = a \cdot s + e$$


$$b' = a \cdot s' + e'$$


shared secret:

$$s \cdot b' = s \cdot (a \cdot s' + e') \approx s \cdot a \cdot s'$$


shared secret:

$$b \cdot s' \approx s \cdot a \cdot s'$$


These are only approximately equal \Rightarrow 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.



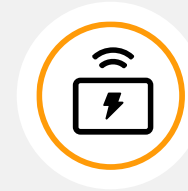
SECURE ELEMENTS
AND END-TO-END
SERVICES



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.

A vertical strip of four images on the left side of the slide. From top to bottom: 1. A robotic arm in a factory setting. 2. A control room with multiple computer monitors displaying data. 3. A large industrial facility with orange overhead cranes. 4. Two autonomous mobile robots (AMRs) in a warehouse environment.

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 $\sigma = \text{Sign}(sk, M)$.

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := H(tr \parallel M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := H(K \parallel \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\gamma_1}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := H(\mu \parallel \mathbf{w}_1)$ 
10:   $c \in B_\tau := \text{SampleInBall}(\tilde{c})$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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 $\sigma = \text{Sign}(sk, M)$.

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := \text{H}(tr \parallel M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := \text{H}(K \parallel \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\gamma_1}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := \text{H}(\mu \parallel \mathbf{w}_1)$ 
10:   $c \in B_\tau := \text{SampleInBall}(\tilde{c})$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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 $\sigma = \text{Sign}(sk, M)$.

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := H(K \| M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := H(K \| \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\gamma_1}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := H(\mu \| \mathbf{w}_1)$ 
10:   $c \in B_\tau := \text{SampleInBall}(\tilde{c})$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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 \mathbf{A} and \mathbf{y} on-the-fly

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

DILITHIUM SIGNATURE GENERATION

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

Input: Secret key sk and a message M .

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

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := H(tr \parallel M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := H(K \parallel \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\kappa}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := H(\mu \parallel \mathbf{w}_1)$ 
10:   $c \in B_\tau := \text{SampleInBall}(c)$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{v} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, 2\gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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 \mathbf{A} and \mathbf{y} on-the-fly: 80, 45 KB

Compress \mathbf{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 $\sigma = \text{Sign}(sk, M)$.

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := H(tr \parallel M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := H(K \parallel \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\gamma_1}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := H(\mu \parallel \mathbf{w}_1)$ 
10:   $c \in B_{\sigma} := \text{SampleInBall}(\tilde{c})$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, \gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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 \mathbf{A} and \mathbf{y} on-the-fly: ~~80 KB~~ → 45 KB

Compress \mathbf{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 $\sigma = \text{Sign}(sk, M)$.

```

1:  $\mathbf{A} \in R_q^{k \times \ell} := \text{ExpandA}(\rho)$  ▷  $\mathbf{A}$  is generated in NTT domain as  $\hat{\mathbf{A}}$ 
2:  $\mu \in \{0, 1\}^{512} := H(tr \parallel M)$ 
3:  $\kappa := 0, (\mathbf{z}, \mathbf{h}) := \perp$ 
4:  $\rho' \in \{0, 1\}^{512} := H(K \parallel \mu)$  (or  $\rho' \leftarrow \{0, 1\}^{512}$  for randomized signing)
5: while  $(\mathbf{z}, \mathbf{h}) = \perp$  do ▷ Pre-compute  $\hat{\mathbf{s}}_1 := \text{NTT}(\mathbf{s}_1)$ ,  $\hat{\mathbf{s}}_2 := \text{NTT}(\mathbf{s}_2)$ , and  $\hat{\mathbf{t}}_0 := \text{NTT}(\mathbf{t}_0)$ 
6:    $\mathbf{y} \in S_{\gamma_1}^\ell := \text{ExpandMask}(\rho', \kappa)$ 
7:    $\mathbf{w} := \mathbf{A}\mathbf{y}$  ▷  $\mathbf{w} := \text{NTT}^{-1}(\hat{\mathbf{A}} \cdot \text{NTT}(\mathbf{y}))$ 
8:    $\mathbf{w}_1 := \text{HighBits}_q(\mathbf{w}, 2\gamma_2)$ 
9:    $\tilde{c} \in \{0, 1\}^{256} := H(\mu \parallel \mathbf{w}_1)$ 
10:   $c \in B_{\sigma} := \text{SampleInBall}(\tilde{c})$  ▷ Store  $c$  in NTT representation as  $\hat{c} = \text{NTT}(c)$ 
11:   $\mathbf{z} := \mathbf{y} + c\mathbf{s}_1$  ▷ Compute  $c\mathbf{s}_1$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_1)$ 
12:   $\mathbf{r}_0 := \text{LowBits}_q(\mathbf{w} - c\mathbf{s}_2, \gamma_2)$  ▷ Compute  $c\mathbf{s}_2$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{s}}_2)$ 
13:  if  $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$  or  $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$  then
14:     $(\mathbf{z}, \mathbf{h}) := \perp$ 
15:  else
16:     $\mathbf{h} := \text{MakeHint}_q(-c\mathbf{t}_0, \mathbf{w} - c\mathbf{s}_2 + c\mathbf{t}_0, 2\gamma_2)$  ▷ Compute  $c\mathbf{t}_0$  as  $\text{NTT}^{-1}(\hat{c} \cdot \hat{\mathbf{t}}_0)$ 
17:    if  $\|c\mathbf{t}_0\|_\infty \geq \gamma_2$  or the # of 1's in  $\mathbf{h}$  is greater than  $\omega$  then
18:       $(\mathbf{z}, \mathbf{h}) := \perp$ 
19:   $\kappa := \kappa + \ell$ 
20: 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**

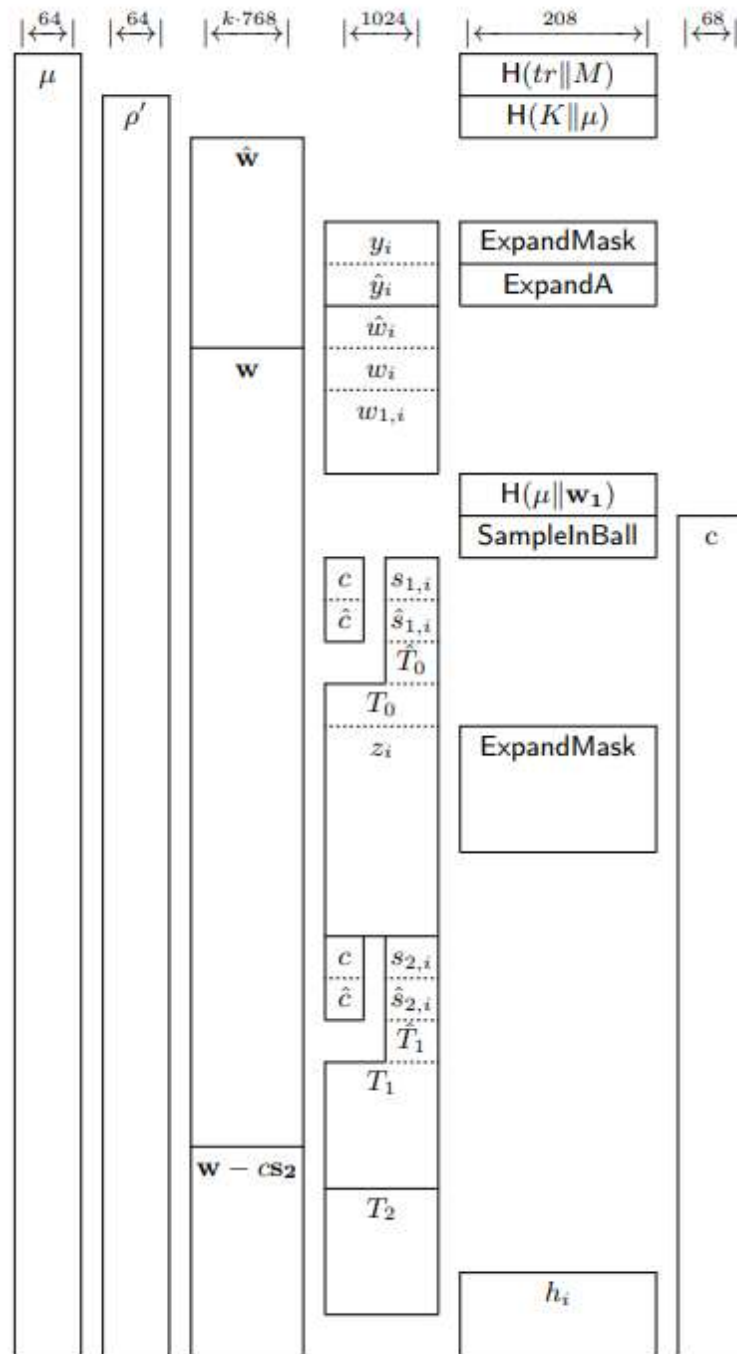
Dilithium-3: $(k, \ell) = (6, 5)$

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

Compress \mathbf{w} : ~~45 KB~~ → 43.5 KB

Compressing multiplications ~~43.5 KB~~ → 26.5 KB

Variable Allocation



```

 $\mu := H(tr \| M)$ 
 $\rho' := H(K \| \mu)$ 
 $\hat{w} := 0$ 

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

```

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

Compress w: ~~45 KB~~ \rightarrow 43.5
KB

Compressing multiplications
~~43.5 KB~~ \rightarrow 26.5 KB

Variable Allocation:

Total of

$64 + 64 + 768k + 1024 +$
 $208 + 68 \text{ bytes} \rightarrow 5268 \text{ bytes}$

In practice: 6.5 KB needed

From Theory to Practice: Small-Memory Implementations

Do these implementations actually run on embedded systems?

		pqm4	
		Runtime	RAM
Dilithium-2	Sign	19 ms	50 kB
	Verify	7 ms	11 kB
Dilithium-3	Sign	31 ms	69 kB
	Verify	12 ms	10 kB
Dilithium-5	Sign	42 ms	123 kB
	Verify	21 ms	12 kB

NXP PQC [A]		Slower	Smaller
Runtime	RAM	Runtime	RAM
61 ms	5 kB	3.2x	10.0x
16 ms	3 kB	2.3x	3.7x
119 ms	7 kB	3.8x	9.9x
29 ms	3 kB	2.4x	3.3x
168 ms	8 kB	4.0x	15.4x
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



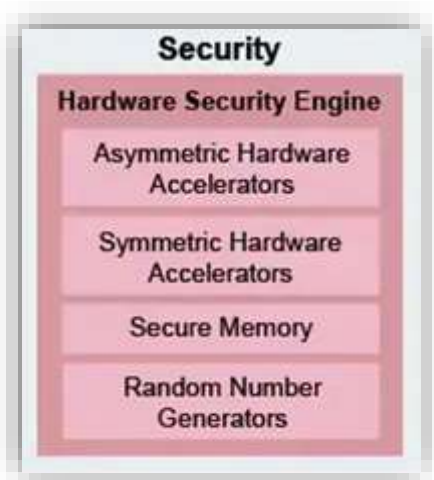
Hardware accelerators will mitigate this



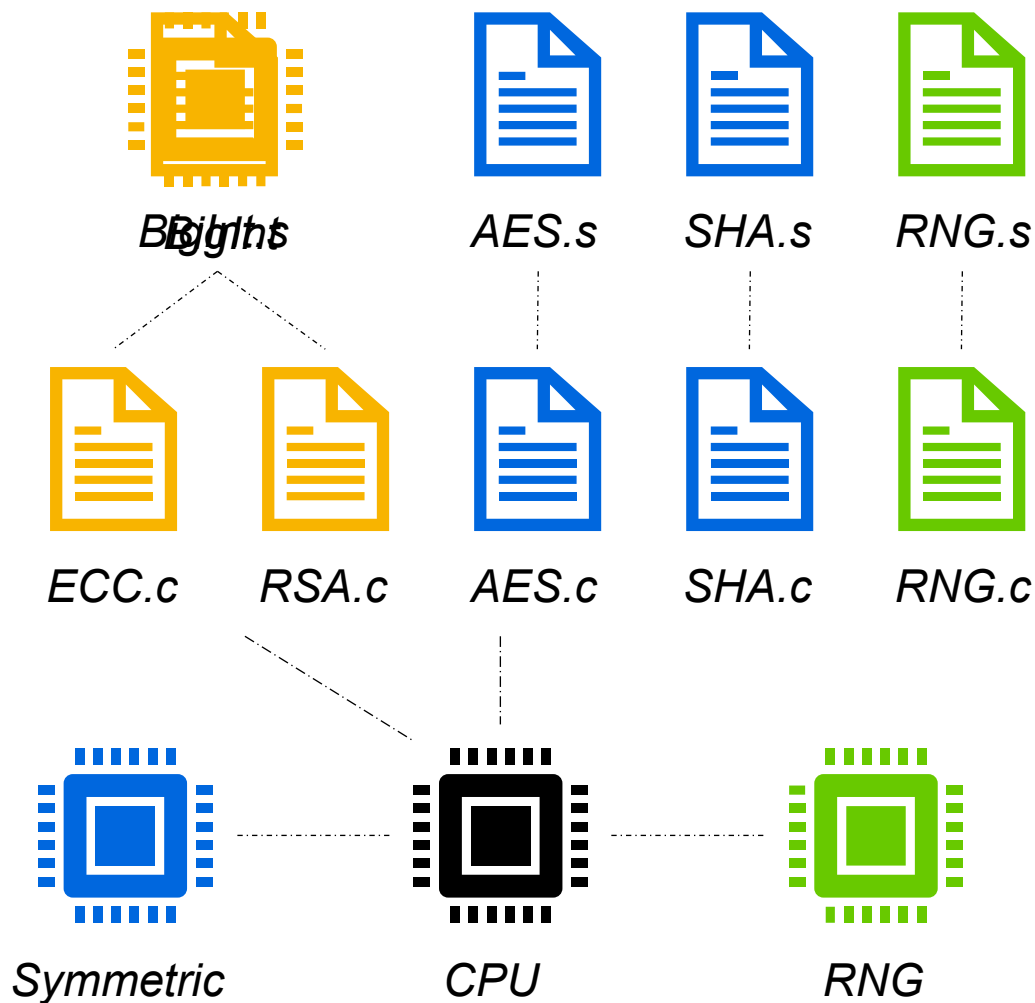
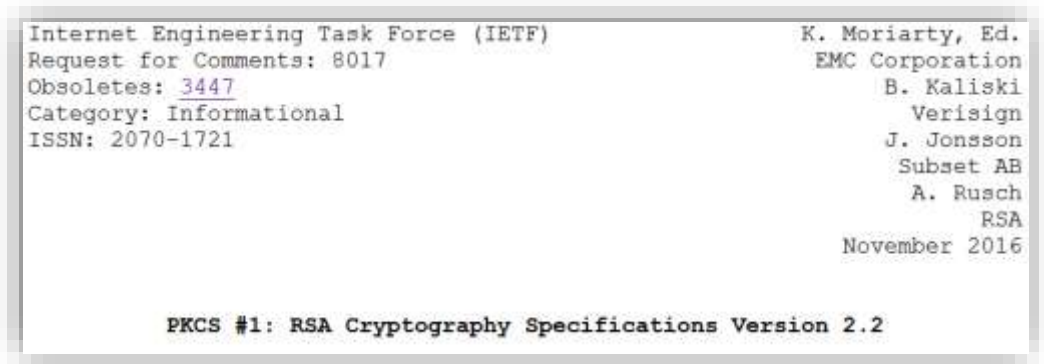
Example of what we do at NXP

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

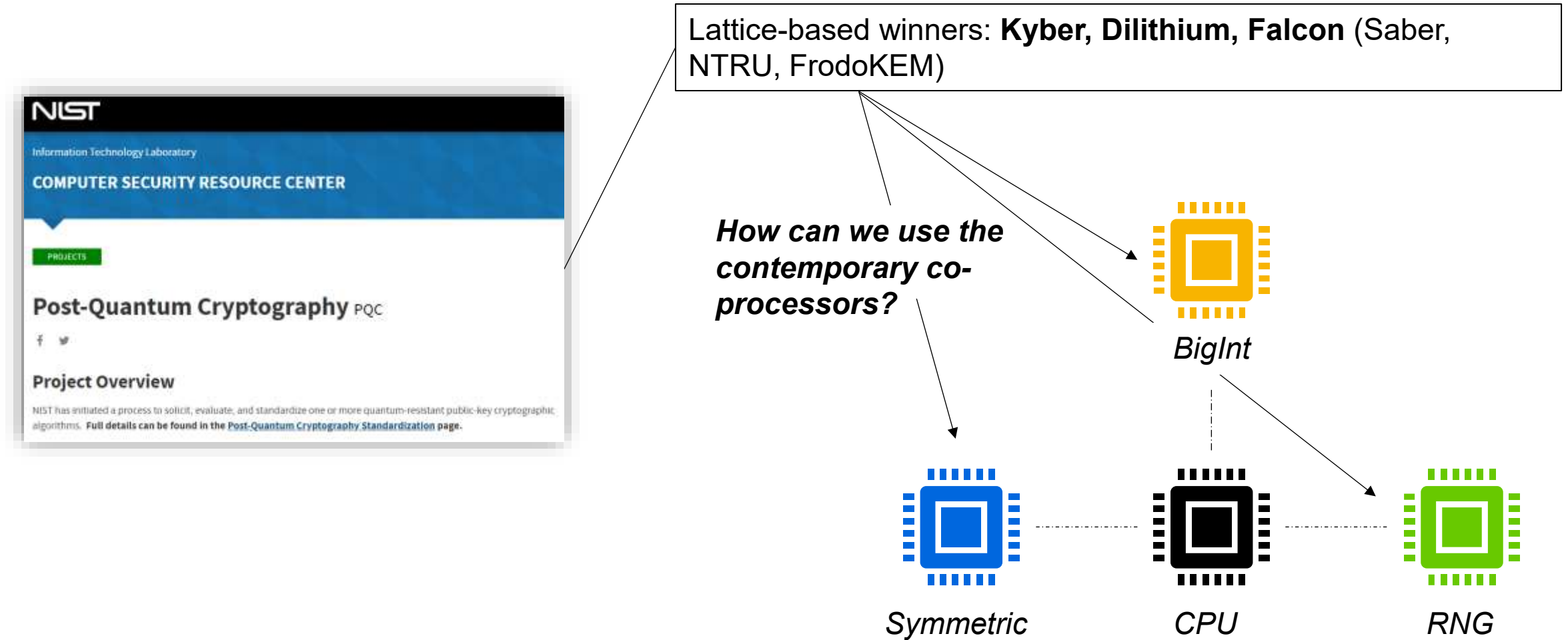
Implementing Classical cryptography



S32G2 automotive processor spec

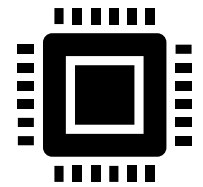


Implementing post-quantum cryptography



Re-using existing HW

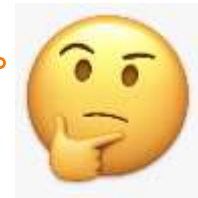
Approach	Core	Structure	Size
RSA	Modular multiplication	$(\mathbb{Z}/n\mathbb{Z})^*$	n is 3072-bit
ECC	Elliptic curve scalar multiplication	$E(\mathbb{F}_p)$	p is 256-bit
Lattice	Polynomial multiplication	$(\mathbb{Z}/q\mathbb{Z})[X]/(X^n + 1)$	q is 16-bit n is 256

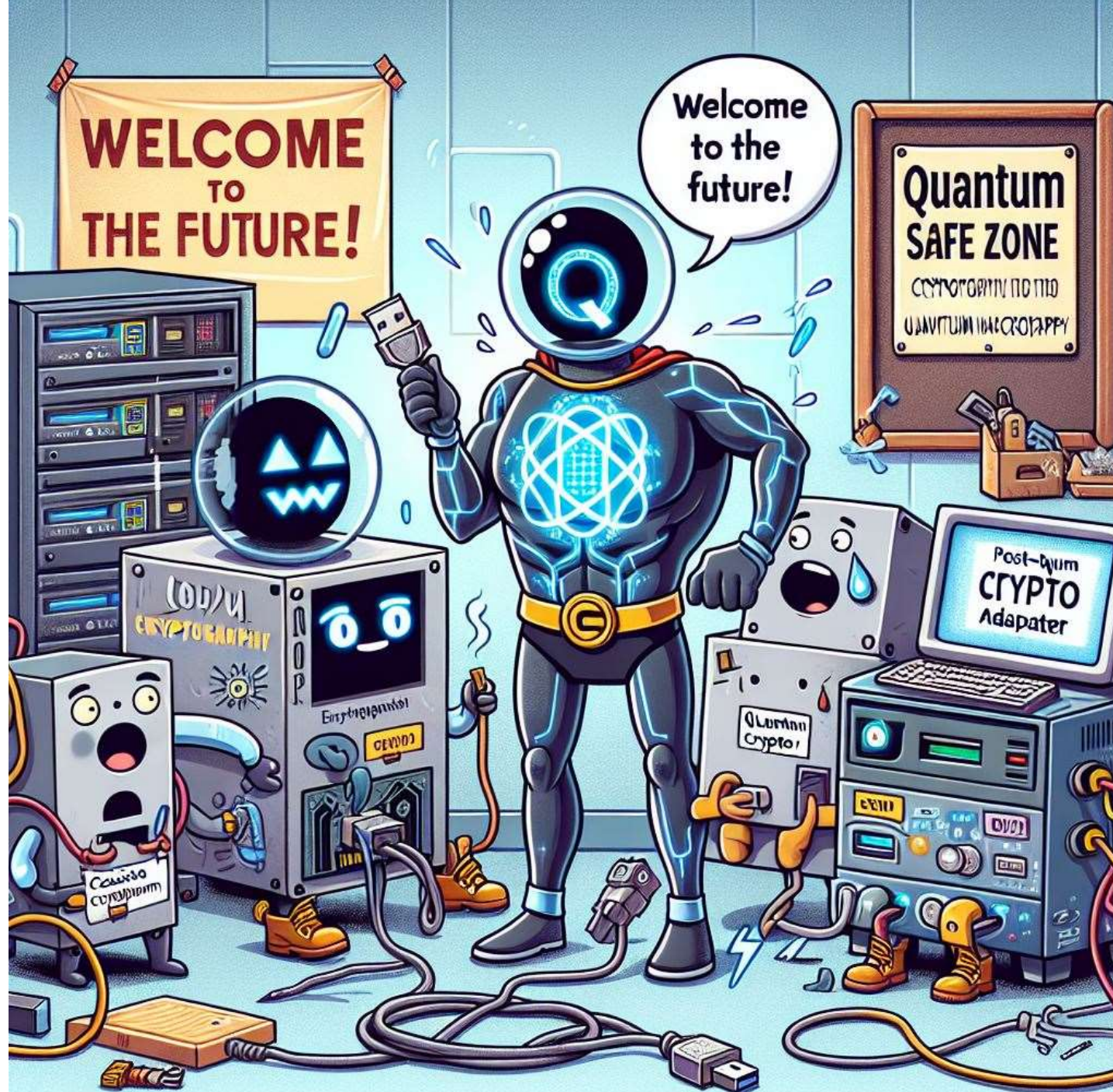


Co-pro present in chips



Can we use this?





Kronecker substitution

Polynomial domain

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

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

×

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

Kronecker domain (with evaluation point 100)

$$f(100) = 4030201$$

$$g(100) = 8070605$$

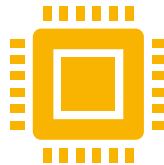
×

$$fg(100) = \underline{32526160341605}$$

Grundzüge einer arithmetischen Theorie der
algebraischen Grössen.

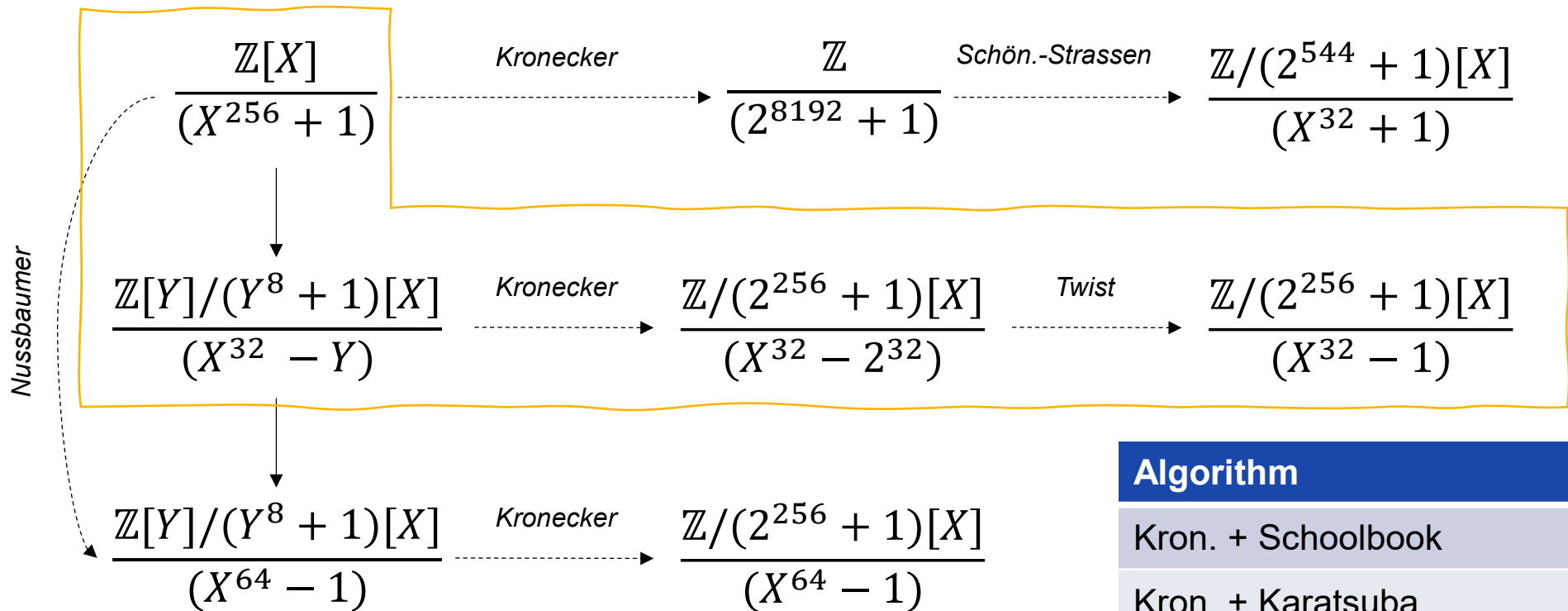
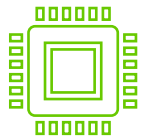
(Von *L. Kronecker*.)

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



Polynomial multiplication techniques

Kronecker evaluation at 2^{32}
 Multiplication with a **256-bit** multiplier



Kronecker+

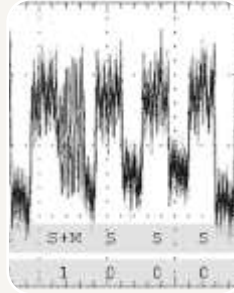
Algorithm	# Muls	# Bits
Kron. + Schoolbook	1024	256
Kron. + Karatsuba	243	256
Kron. + Toom-Cook	63	256
Kron. + Schön.-Strassen	32	544
Nussbaumer + Kron.	64	256
Kronecker+	32	256

[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

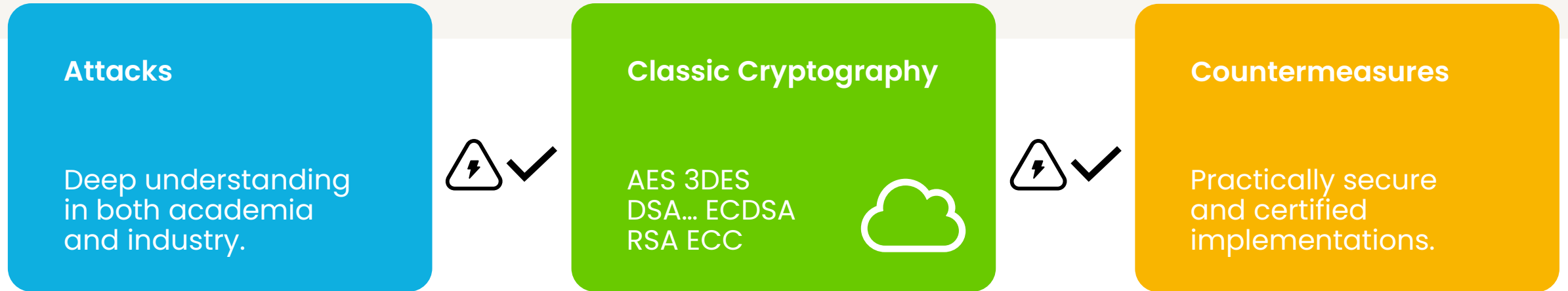
- Voltage or clock glitching
- Electromagnetic fault injection (EMFI)
- Body bias injection
- Laser fault injection
- Single and multi-shot scenarios



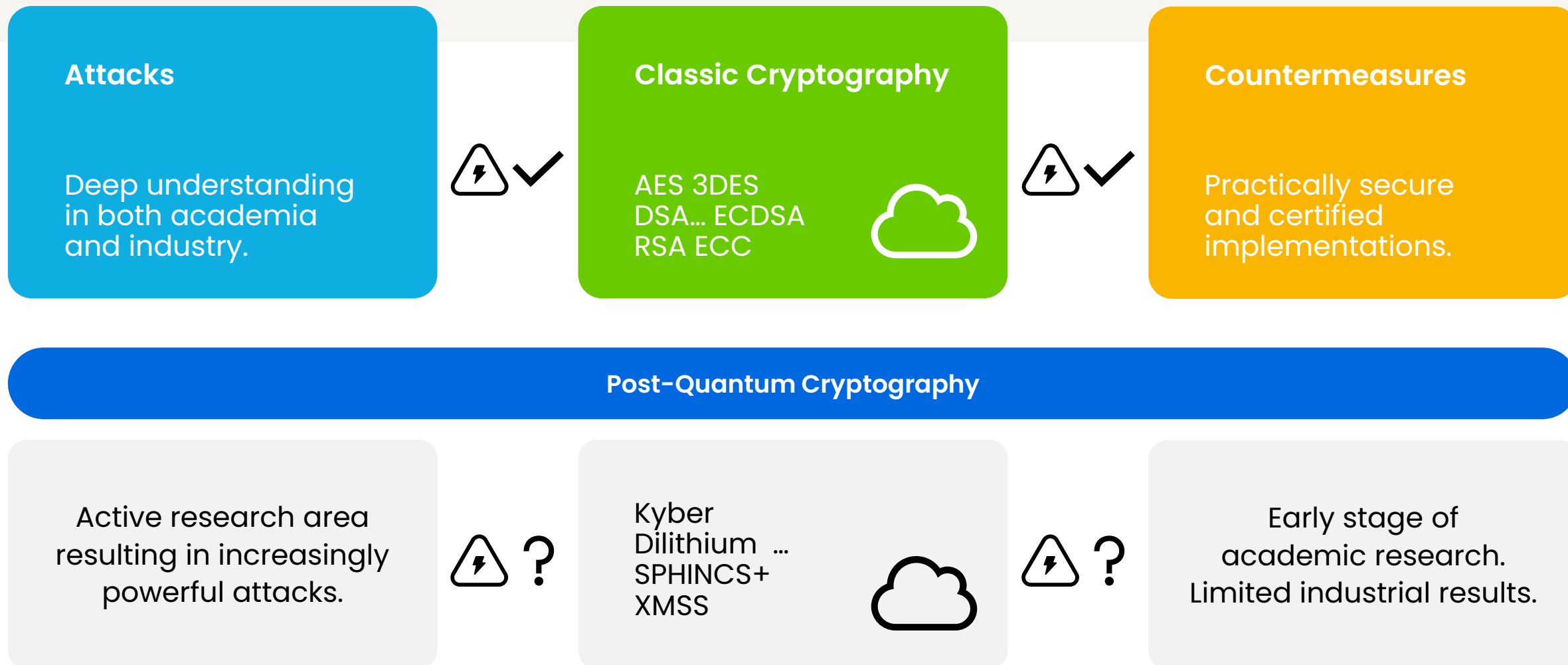
Invasive attack

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

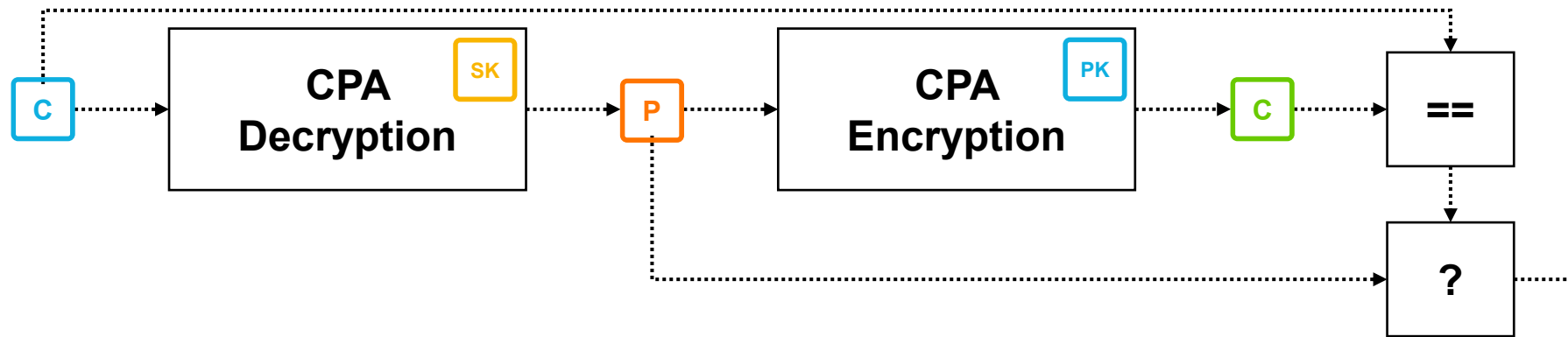
Embedded cryptography and implementation attacks



Embedded cryptography and implementation attacks



Fujisaki Okamoto transform



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

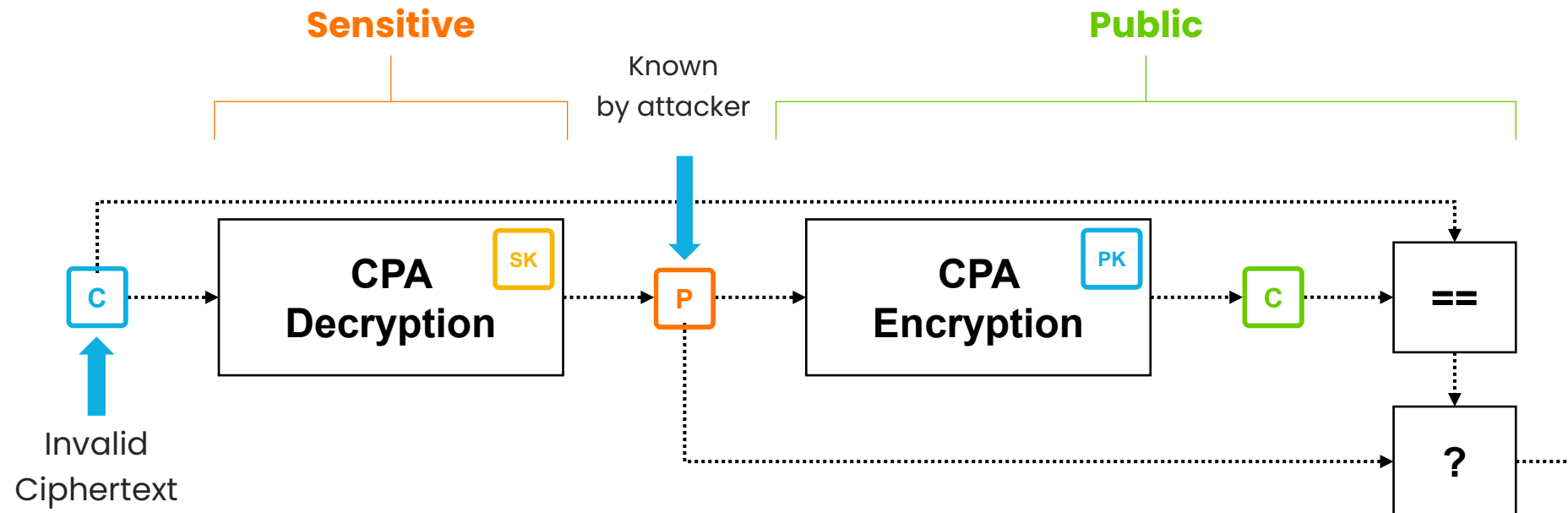
Fujisaki, E. and Okamoto
T., Secure integration of
asymmetric and symmetric
encryption schemes, CRYPTO
1999 and JoC 2013

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**

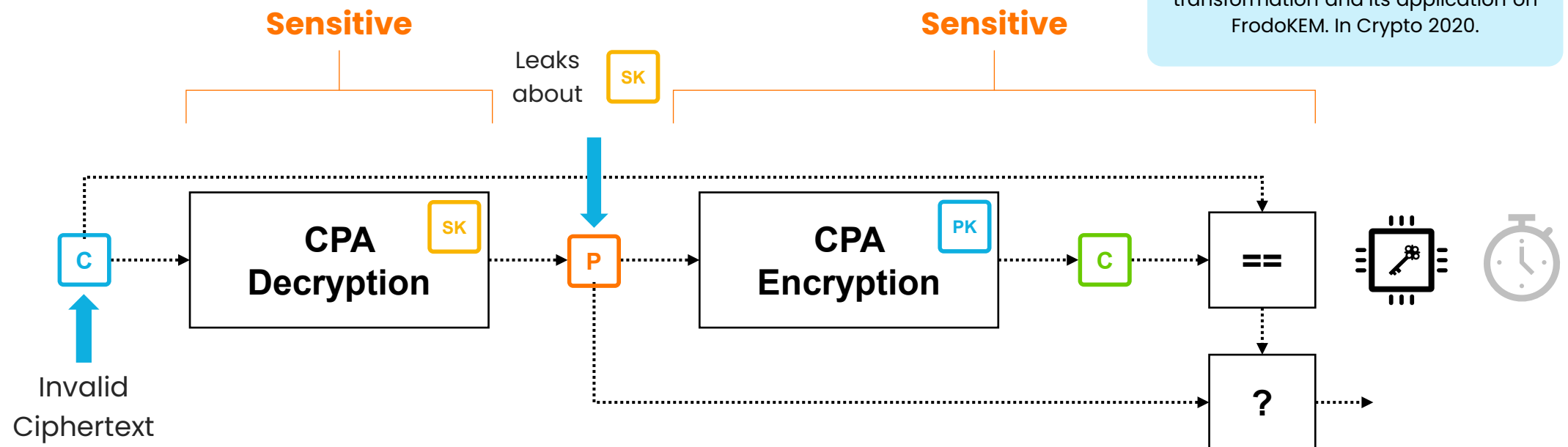


The SCA Problem of the FO-Transform



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



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 !

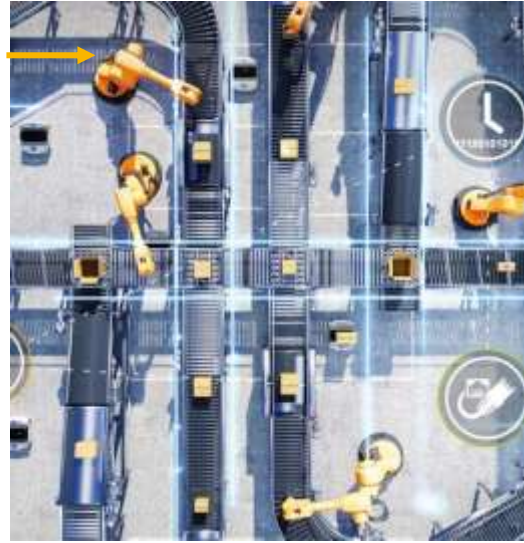
Year	Venue	FIPS 203	FIPS 204	Title
2021	TCHES			Masking Kyber: First- and Higher-Order Implementations
2021	RWC			Post-Quantum Crypto: The Embedded Challenge
2022	TCHES			Post-Quantum Authenticated Encryption against Chosen-Ciphertext SCA
2022	RWC			Surviving the FO-calypse: Securing PQC Implementations 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 !

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

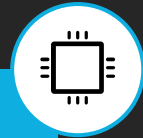
- ✓ Secure boot,
- ✓ Secure update
- ✓ Secure debug of the processor based on PQC

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



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



Marc Fischlin¹, Jonas von der Heyden² (✉), Marian Margraf³, Frank Morgner⁴, Andreas Wallner⁵, and Holger Bock⁵

Extended Access
Control (EAC)

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

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](https://www.nxp.com)



Brighter Together