From Theory to Practice

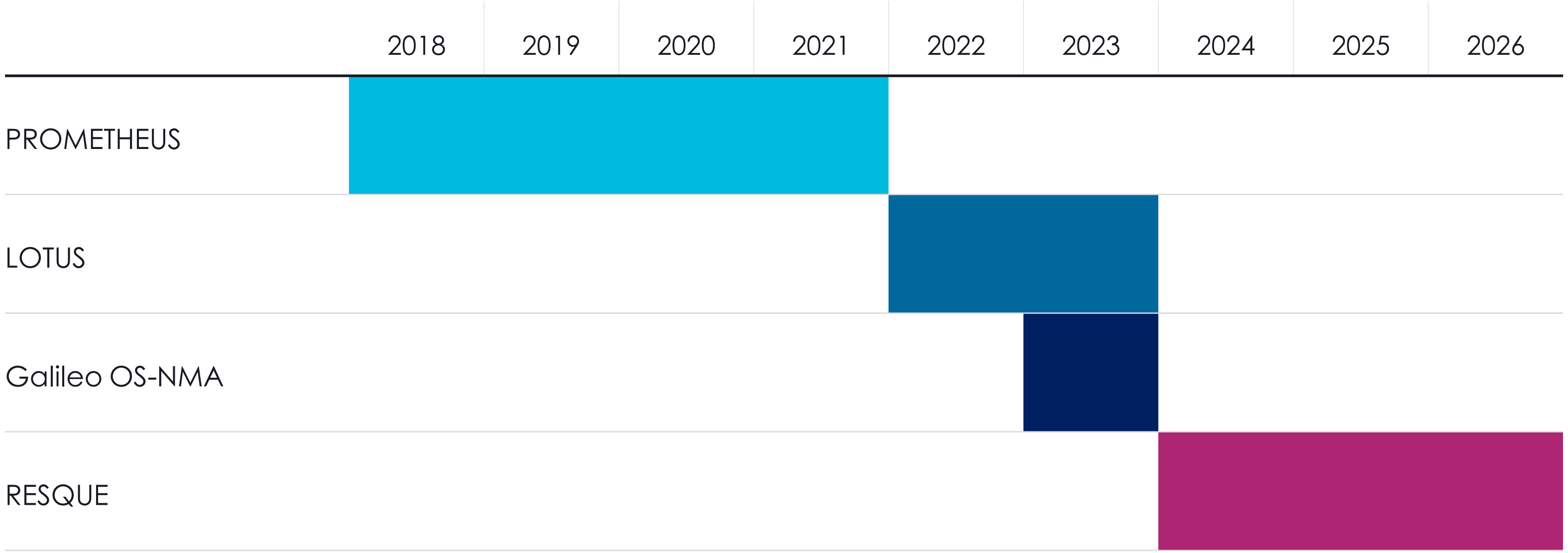
Integrating Post-Quantum Solutions in Real-World Systems

ECW 2024

www.thalesgroup.com

THALES
Building a future we can all trust

Introduction



Part I: Applications

Post-quantum Anonymous Credentials
Hybrid Authenticated Key Exchange
Post-quantum mechanisms in Galileo OS-NMA

Part II: Implementations

Crypto-agility in crypto software libraries
Side-channels attacks and their countermeasures
Hybrid hardware-software architectures

APPLICATIONS OF POST-QUANTUM CRYPTOGRAPHY

Post-quantum Anonymous Credentials

Hybrid Authenticated Key Exchange

Post-quantum mechanisms in Galileo OS-NMA

PROMETHEUS
PRivacy preserving pOst-quantuM systEms from advanced crypTograpHic mEchanisms Using latticeS

PROMETHEUS – H2020 EU project 2018-2023

- EU collaborative project on **post-quantum protection of personal data** :
  - 8 academic / 4 industrial partners,
  - France/Germany/United Kingdom/Spain/Netherlands/Israel

- Scope : protect **children privacy** when browsing a **video streaming service.**

- Goal : to allow a user to **gain access to age-appropriate video** without having to **disclose personal information** to the service.

> Our solution QPACE – Quantum Proof Anonymous Credential Engine

Preserving privacy with post-quantum anonymous credentials

The **user** aims to prove legitimate access to a **service** that enforces a **verification** (identity, age, etc.) while preserving its **privacy**.

Step 1 : the user requests a **credential** that will be delivered by a trusted third party.
Step 2 : credentials are used by the QPACE to generate **ephemeral tokens** which will be verified by a service.

Lattice-Based Group signature **[dPLS18]**
RSIS/RLWE/NTRU assumptions → quantum resistant

NIZK Proof system **[ALS20]**

RSIS = (Ring)-Short Integer Solution
RLWE = (Ring)-Learning With Errors
NTRU-based cryptosystem

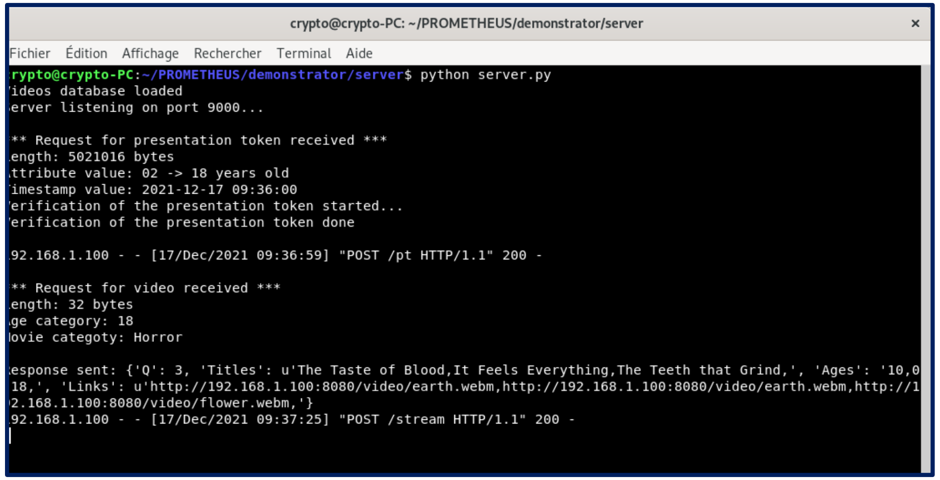
[dPLS18] Rafaël del Pino, Vadim Lyubashevsky and Gregor Seiler. Lattice-Based Groupe Signatures and Zero-Knowledge Proofs of Automorphism Stability. CCS 2018.

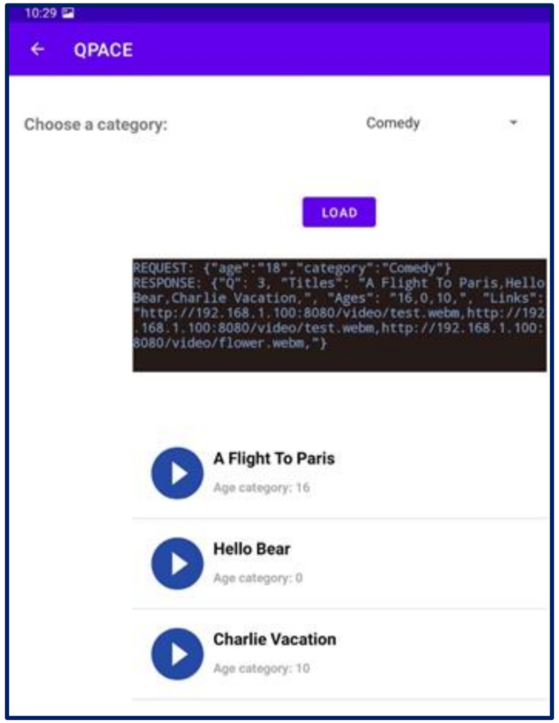
[ALS20] Thomas Attema, Vadim Lyubashevsky and Gregor Seiler. Practical Product Proofs for Lattice Commitments. CRYPTO 2020.

Performances of our QPACE demonstrator

Transmission + data parsing
~ few seconds

Average: ~ 5 seconds
Worst-case ~10 seconds





Presentation Token (average size 5 MB)	CPU cycles (min / average / max)	Seconds (min / average / max)
Generation	1 / **5** / 20	0.5 / **2** / 6
Verification	5.4 / **5.43** / 5.5	0.169 / **0.17** / 0.174
Total	6.4 / **10.43** / 25.5	0.679 / **2.17** / 6.174

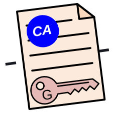
Room for improvements
- ✓ faster proof system
- ✓ better data parsing
- ✓ only a demonstrator

RESQUE – BPI France 2023-2026

HSM

- **Infrastructures protection**
- *Performances ++*
- *Protection against Side Channel Attacks*

certificates provider



VPN gateway / encryptor

- **Protection of communications**
- *Post-quantum evolution of IPsec IKEv2 standard*
- *Integration of an hybrid AKE*

Thales MISTRAL + Hybrid AKE solution

Thales DIS dedicated co-processor (HSM Luna) for HW accelerations

Algorithms :
- Dilithium (**ML-DSA**)
- Kyber (**ML-KEM**)

ML = Module-Lattice-Based

Ipsec tunnel

VPN client

- **Remote workers protection**

Efficient **hybrid solution** toward a safe, progressive post-quantum cryptography to **protect companies, administration and remote workers.**

Hybrid Authenticated Key Exchange in RESQUE

Aim : Adapt the IPsec DR specification of IKEv2 to add Hybrid Key Exchange and Hybrid Authentication

ANSSI views on PQC transition [1] :

Hybrid Key Exchange: ETSI's [2] hybridation modes for Key Encapsulation Mechanisms are recommended

Hybrid Authentication:

- Hybridation is made by combining classical and PQ signatures
- No choice of hybrid certificates : « designs and security proofs of such hybrid certificate protocols are still currently moving »

[1] ANSSI views on the Post-Quantum Cryptography transition (2023 follow up) (December 21, 2023)

[2] ETSI. Quantum-safe hybrid key exchanges

Hybrid AKE in RESQUE – our solutions

Hybrid Key Exchange: one solution

Proposed Standard RFC 9370 proposes a solution to add multiple key exchanges on IKEv2

→ We **adapted it** by replacing hybridation modes by ETSI's ones.

Hybrid authentication: three solutions

At the protocol level, by using **experimental RFC 4379** to allows multiple authentication exchanges

At primitive level, by using **hybridized certificates**:

- One adopted by the **LAMPS working group**, not yet formalized as an RFC (draft-ietf-lamps-pq-composite-sigs-02)
- One based on the **multiple-algorithm certificate** defined in "X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks"

Post-quantum mechanisms in Galileo OS-NMA (2023)

> Context:

- Project **E-GIANTS** : **E**uropean **G**NSS[1] **I**mproved **A**uthentication **S**olutions
- Requested by the European Commission
- 6 industrial partners

> Our mission: Cryptographic support for the Galileo OS-NMA mechanism

- Review of the current OS-NMA[2] mechanism
- Identification of possible cryptographic improvements for the long term (> 2030)

→ **Consider post-quantum mechanisms**

1. **GNSS**: **G**lobal **N**avigation **S**atellite **S**ystem
2. **OS-NMA**: Galileo **O**pen **S**ervice **N**avigation **M**essage **A**uthentication

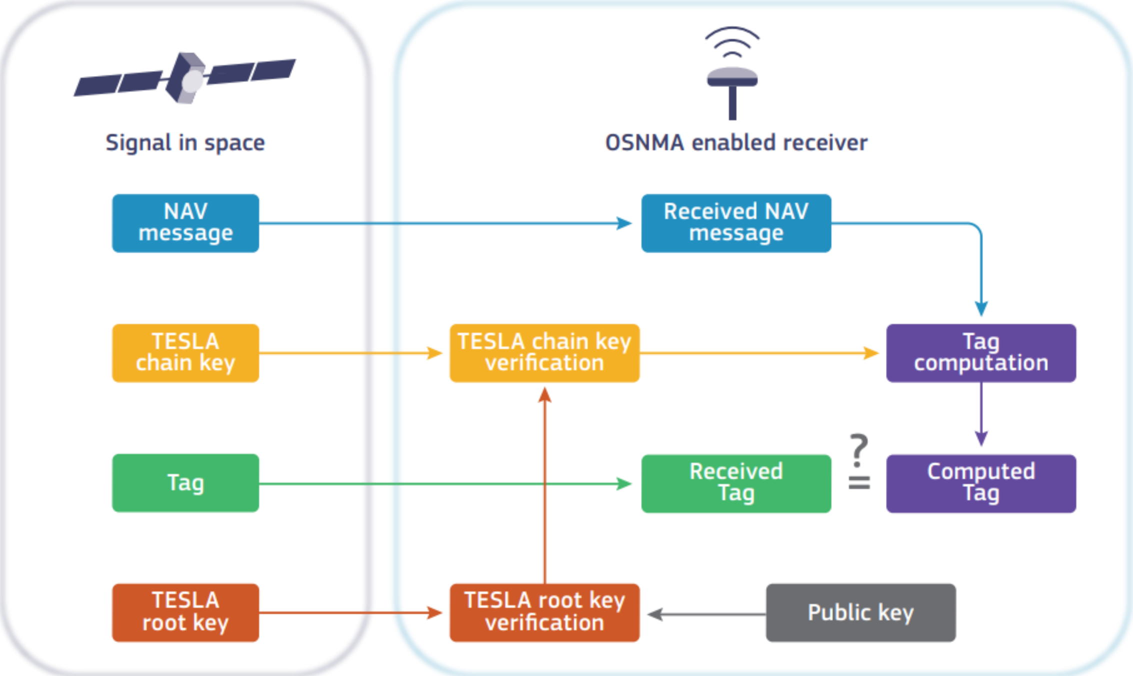
Post-quantum mechanisms in Galileo OS-NMA

> OS-NMA:

- Open, free of charge positioning and timing service
- Authenticates the messages broadcast by the satellites
- Based on **TESLA protocol** [RFC 4082]

TESLA chain derivation
SHA-256
SHA3-256

TESLA root key verification
ECDSA-256
ECDSA-512



Tag computation
HMAC-SHA-256
AES-CMAC

Public key verification
Merkle tree
(SHA256 / SHA3-256)
Or certificates (ECDSA-256)

Source: Galileo High Accuracy Service (HAS) Info Note. © European Union Agency for the Space Programme, 2021

Post-quantum mechanisms in Galileo OS-NMA

> Suggested improvements:

- Symmetric algorithms: increase the parameter size
- Digital signatures: use post-quantum schemes

} According to the recommandations (hybridation, use of standard PQC schemes and parameter sets)

> Principal constraint: The Signal in Space has a very limited bandwidth

- Transmit PQ signatures within the Signal in Space seems impossible

> Possible solutions:

- Increase the Signal In Space bandwidth
- Preload some data and signatures if possible

No specific algorithms selected yet, work still in progress!

Part I: applications – what we learned so far

> Anonymous Credentials

PQ algorithms **not mature yet** → **waiting time** before accessing the service, need to **reduce PT generation time** (improve primitives and performances?)

> Hybrid KEMs and certificates

Lot of (too much?) solutions but:

- not always fully **mature**, need **adaptations**
- not always **compliant** with **official recommendations** (hybrid KEMs ≠ ANSSI recos)

> PQ for space communications

Need to **increase bandwidth** or **pre-load** as much data as possible

Only **preliminary studies**, lack of **mature solutions** for space systems

IMPLEMENTATIONS OF POST-QUANTUM CRYPTOGRAPHY

Crypto-agility in crypto software libraries

Side-channels attacks and their countermeasures

Hybrid hardware-software architectures

What is crypto-agility ?

Generic interface providing services: encryption, signature, hashing, KEM, etc.

✓ Different **algorithms, parameters, modes** and **implementations** → flexible input/output sizes, internal state, etc.

SHA256, SHA512, SHAKE256_384, HMAC-SHA256, etc. → **HASHFunction**

XMSS_2/5/10/14_SHAKE256_256, XMSS_2/16_SHAKE256_512 → **Signature**

Regular implementation, specific instruction set enabled, masked version against SCAs, etc.

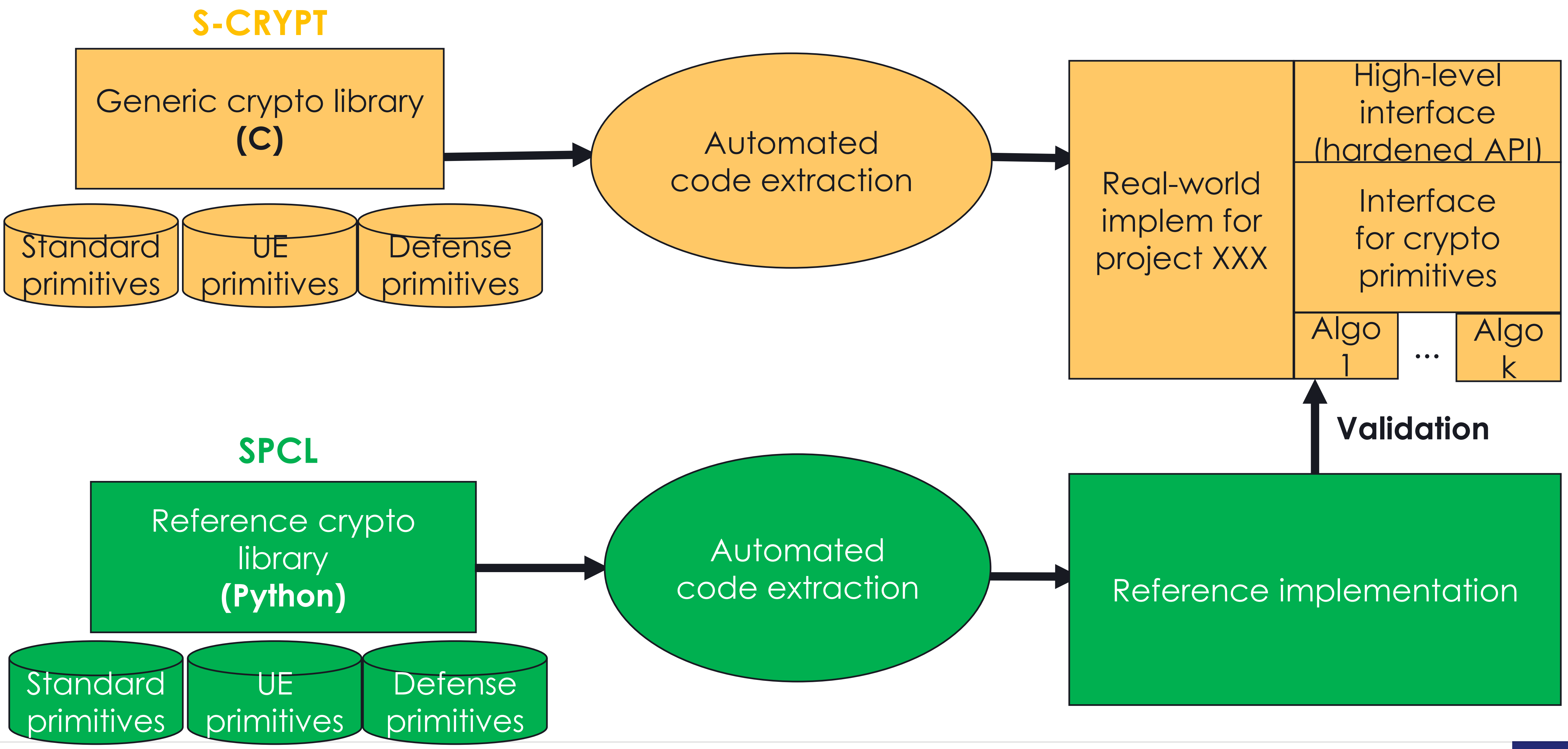
✓ Common **validation** process→ **CAVP** (NIST Cryptographic Algorithm Validation Program) and **tests vectors** specified by standards and norms

Facilitates migration toward post-quantum algorithms, but <u>not limited to PQ</u>

✓ Generic interfaces with **interchangeable components** (algorithms, parameters, etc.)

✓ Crypto is constantly evolving → need **flexibility** to **anticipate** future changes

S-CRYPT (real-world implementation) / SPCL (reference implementation)



Integration of many primitives, included PQ algorithms

> SPCL (Python reference library)

XMSS (RFC 8391), SPHINCS+(SLH-DSA), Dilithium (ML-DSA), Falcon (FN-DSA), FrodoKEM, etc.

> S-CRYPT (C real-world implementation)

✓ XMSS_2/5/10/14_SHAKE256_256, XMSS_2/16_SHAKE256_512

✓ FrodoKEM_640/976/1344_AES/SHAKE

✓ Wrapping of *libecc* (Elliptic Curve primitives) and *liboqs* (many PQ algorithms) for **software risk assessment**

✓ **Hardened interface**: tests for vulnerabilities in memory allocation, incorrect parameters, code coverage, fuzzing, etc.

S-CRYPT: optimisation and hardening

XMSS

- ✓ **parallelisation** in Merkle trees and authentication path construction
- ✓ better **time-memory tradeoffs**
  - **BDS algorithm**, **full tree in memory** during KeyGen to minimize signature time (30min ↘ few min)

Hardened interface

- ✓ tests for vulnerabilities in **memory allocation**
- ✓ incorrect **parameters**
- ✓ **code coverage**, **fuzzing**, etc. → **fuzzing on XMSS_BDS resulted in several bugs identified**

Simple crypto library encompassing only the strict necessity

Hybrid hardware-software architectures for PQ algorithms

> Step 1: software implementation

Which SW implementation?

> Step 2: benchmarking

Execution time, number of operations, most used functions…

> Step 3: identification of bottlenecks

Which **operations** or **subparts** of the algorithm would benefit from HW acceleration?

> Step 4: partial HW implementation VS full HW implementation

Full HW implementation is complex, costly, not flexible (crypto-agility) and not always necessary → best to **accelerate specific operations**

Dedicated HW accelerators for PQ building blocks

PQ algorithm	HW acceleration
BIKE	✓ MUL for large & sparse polynomials over binary rings ✓ Decoder
FrodoKEM	✓ SHAKE (eXpendable Output Function) ✓ Sampling ✓ Matrix multiplication
Dilithium (ML-DSA)	✓ Number Theoretic Transform (NTT) ✓ Fast Fourier Transform (FFT) ✓ Sampling
Kyber (ML-KEM)	SHAKE
Falcon (FN-DSA)	✓ NTT, FFT ✓ Floating point operations ✓ Gaussian sampling
XMSS (RFC 8391)	Full HW implementation

LOTUS : HW acceleration for BIKE and FrodoKEM (2021-2023)

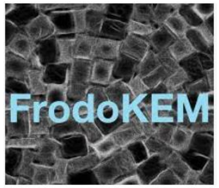
1

Study of 6 KEMs

6 post-quantum KEMs studied
3 Lattice based
CRYSTALS KYBER, FrodoKEM, NTRU
3 Code based
ROLLO-I, BIKE, Classic McEliece
Bibliographical study
Implementation technics
Side-channel attacks
& Counter-measures
Implementation of a reference cryptographic library

2

Selection of 2 KEMs



BIKE

Embedded SW implementation with highest security parameters (level 5)
Side channel analysis and counter-measure implementation
Partial HW acceleration

3

Results

Two attack paths over the most critical operations were partially or totally exploitable

Implemented countermeasures against all these attacks

HW acceleration x20 (level 5)

THALES   RÉPUBLIQUE FRANÇAISE   Inria   IRISA

Hybrid hardware-software architecture in RESQUE

- ✓ Analysis of Dilithium (ML-DSA) and Kyber (ML-KEM) schemes to determine **which internal building blocks** should be **HW-accelerated by the HSM Luna (Thales DIS)**:

  eXpendable Output Functions **SHAKE**

  Arithmetic over $\mathrm{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ with flexibility over the choice of q

  → necessary for crypto-agility and protection against SCA

- ✓ **State-of-the-art of existing HW implementations** for ML-DSA and ML-KEM

  → necessary to choose the most pertinent implementation and reach the best **flexibility-efficiency-resources tradeoff**

  crypto-agility | latency and bandwith | surface

Side-channels attacks and countermeasures in RESQUE (2023-2026)

The work so far

- ✓ State-of-the-art of **side-channel attacks** against **Dilithium (ML-DSA)** and **Kyber (ML-KEM).**
- ✓ State-of-the-art of applicable **countermeasures**.
- ✓ Several **Working Groups** aiming at **solving open problems** such as

    How to mask critical variables and challenging operations?

    How to improve existing attacks and countermeasures?

    How to reduce the performance gap between non-protected implementations and protected ones (new gadgets, etc.)?

Part II: implementations – what we learned so far

> Hybridation

- Many possibilities but nothing fully **mature or compliant** with recommandations
- Lack of **standards and documentation** (ex. hybrid certificates)

> Advanced primitives

- Anonymous Credentials, Attribute-Based Encryption, etc. also **lack maturity**
- Need more investigation and **better performances**

> Secure, flexible and optimized implementations

- Every algorithm relies on specific mechanisms → **unique challenges and solutions**
- Crypto-agility requires more work but provides **flexibility to anticipate the future**

Contact

AMBLARD Zoé

Cryptology engineer (SCR, Thales)

zoe.amblard@thalesgroup.com

THALES

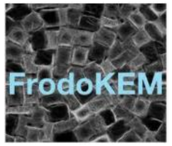
Building a future we can all trust

Merci

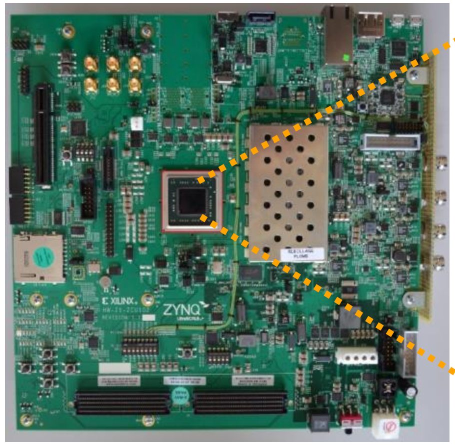
www.thalesgroup.com

Annex: LOTUS hardware platform

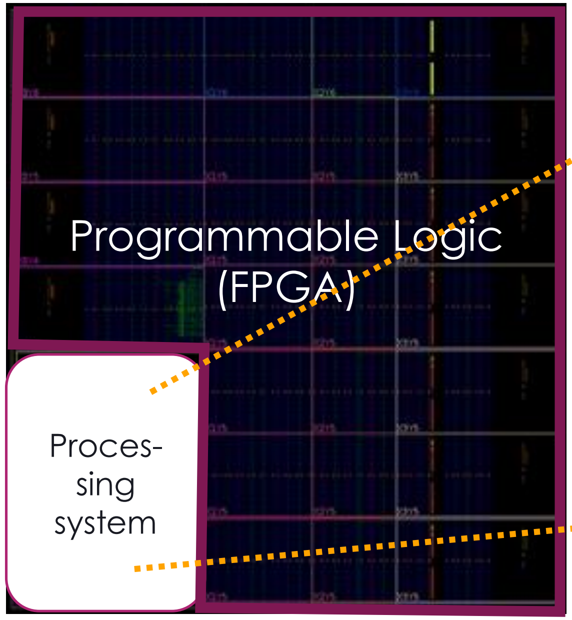
Embedded Software



BIKE



Xilinx Ultrascale+ ZCU102 board



Xilinx ZYNQ Ultrascale+ MPSOC FPGA component