University of Applied Sciences and Arts Northwestern Switzerland
School of Engineering and Environment

END-TO-END QUANTUM SAFE SECURITY FOR SATELLITE DATA LINKS

Christoph Wildfeuer,
Timeo Jauslin, **Alain Lavoyer**,
Milenko Starcik, Afonso Serra,
Laszlo Etesi, Valentina Tamburello,
Bruno Huttner

VISIONSPACE

esa

IDQ

ATELERIS

Fachhochschule Nordwestschweiz
Hochschule für Technik

Overview

- Introduction
- PQC Algorithms
- Certificates
- CCSDS Protocol Stack
- System Architecture
- Protocol
- Formal Verification
- Python Simulator

Introduction

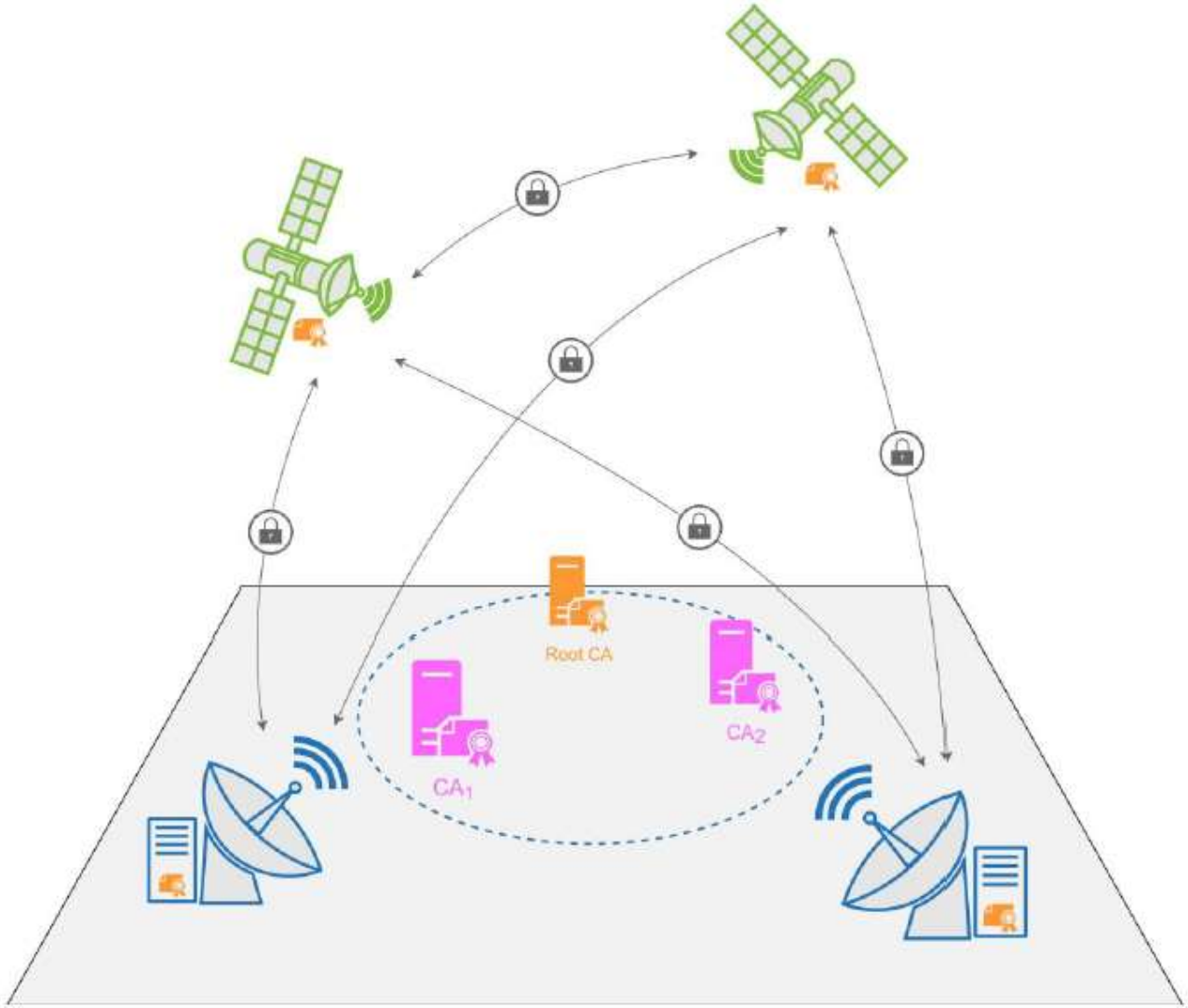
Asymmetric Key Exchange

↓

Symmetric Key

↓

Symmetric Bulk Encryption



Motivation

– Symmetric cryptography requires every user to share a symmetric key with every other user

$$\frac{n\,(n-1)}{2} = O(n^2)$$

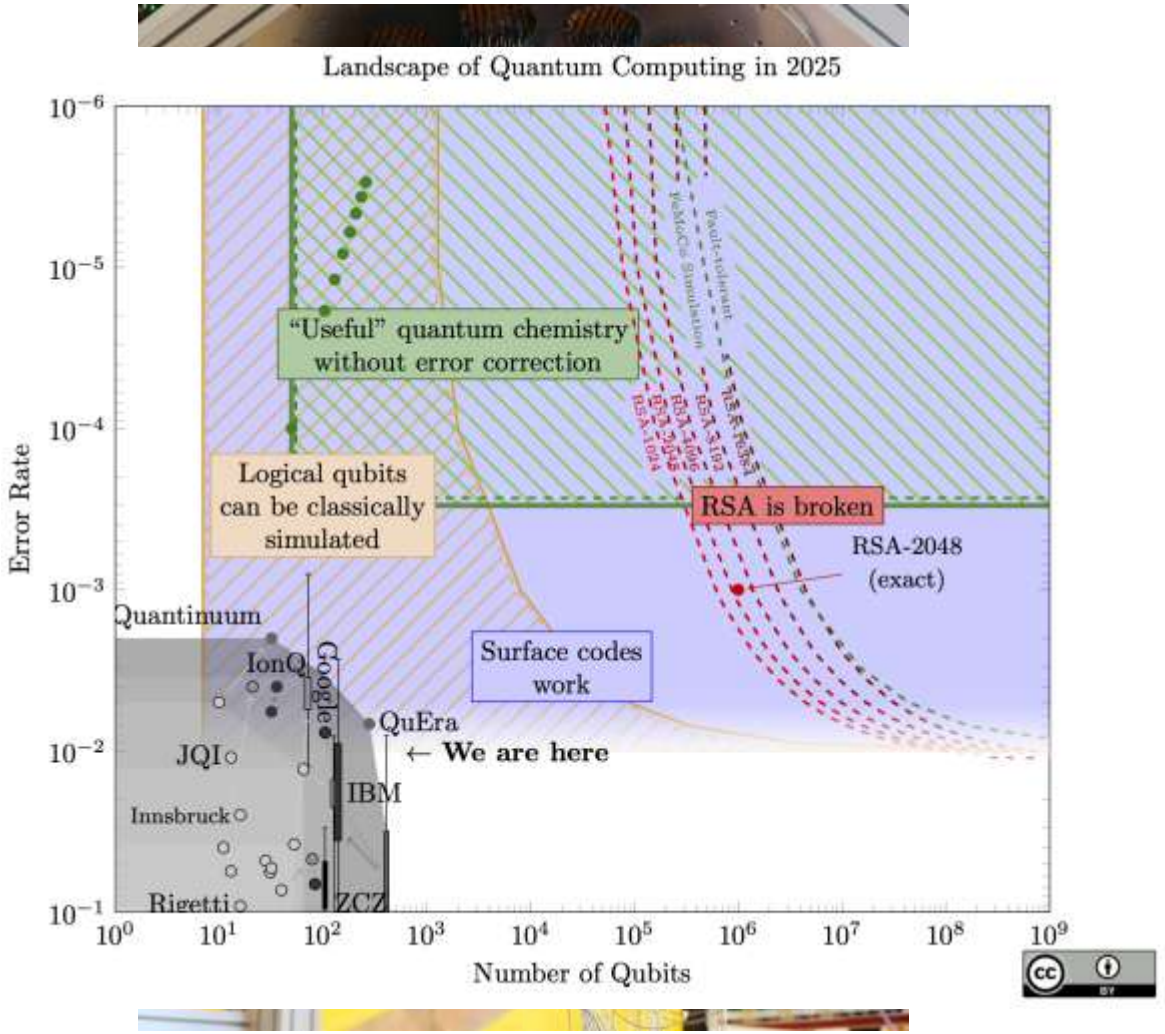
7’600 Starlink satellites → ~28’876’200 symmetric keys!

– Asymmetric cryptography requires every user to have **one** key pair

$$\mathrm{n} = O(n)$$

Motivation

- Quantum threat
- RSA and ECDH will be broken
- AES requires doubled key sizes
- “harvest now – decrypt later”
- Use Post-Quantum Cryptography algorithms



https://quantumzeitgeist.com/quantum-computing-landscape

ML-KEM

FIPS 203

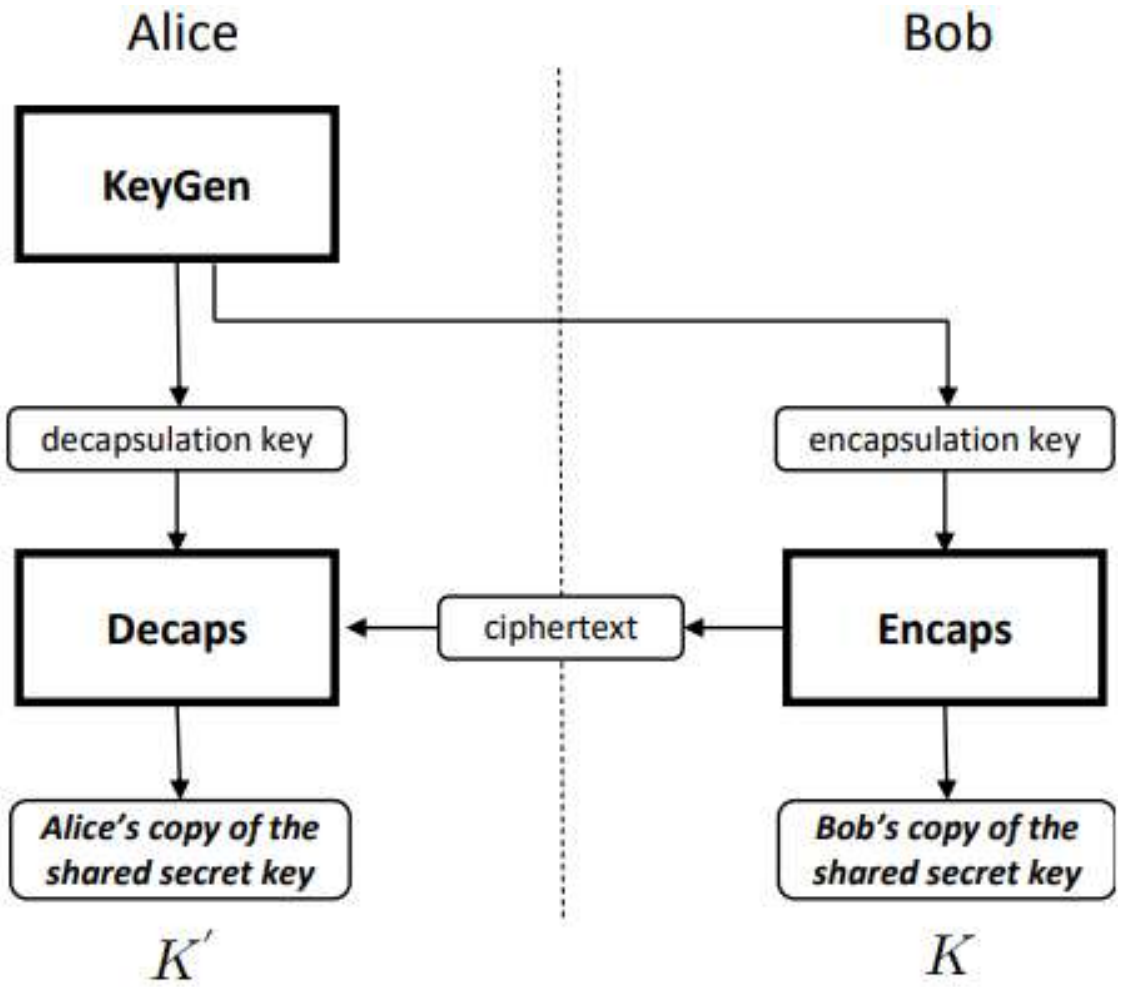
Federal Information Processing Standards Publication

Module-Lattice-Based Key-Encapsulation Mechanism Standard

3 Functions:
- (*ek, dk*) ← ML-KEM.KeyGen()
- (*c, K*) ← ML-KEM.Encaps(*ek*)
- K' ← ML-KEM.Decaps(*dk, c*)

3 Security Categories:
- ML-KEM-512 (cat. 1 ≙ AES-128)
- ML-KEM-768 (cat. 3 ≙ AES-192)
- ML-KEM-1024 (cat. 5 ≙ AES-256)

security strength categories



FIPS 203

ML-KEM Hardware Accelerator

Implementation	#LUT / #DSP / #BRAM	NAC	#kCycle (KeyGen/Encaps/Decaps)	Time [μs] (KeyGen/Encaps/Decaps)
Our ML-KEM	7356/4/6.5	8'536	26/24/31	130/121/155
Area Efficient [1]	7412/2/3	7'972	6.3/7.9/10.2	39.2/47.6/61.3
Direct Impl. [2]	97k/36/200	124k	-/77/102	-/500/659
High Perf. [3]	10.4k/6/8.5	12'020	2.7/3.9/5.0	12.3/17.7/22.9
More generic [4]	14k/11/14	16'780	112/177/191	4461/7102/7623
Co-Processor [5]	25k/0/2	25'200	5.5/66/8.0	36.4/44.1/53.6
RISC-V [6]	24k/18/32	29'640	273/325/340	-
HLS [7]	1978k/-/-	1978k	-	-

Core / Submodule	#LUTs	#DSPs	#BRAMs	NAC
NTT	484	2	0	684
Poly Arithmetic	443	1	0	543
Keccak and Sampling	4743	0	2	4983
Non Polynomial	1222	1	2	1562
L2 Memory and DMA	150	0	1.5	330
Control Logic	314	0	1	434
Total	7356	4	6.5	8536
[%]	14%	1.8%	4.6%	-

ML-DSA

FIPS 204

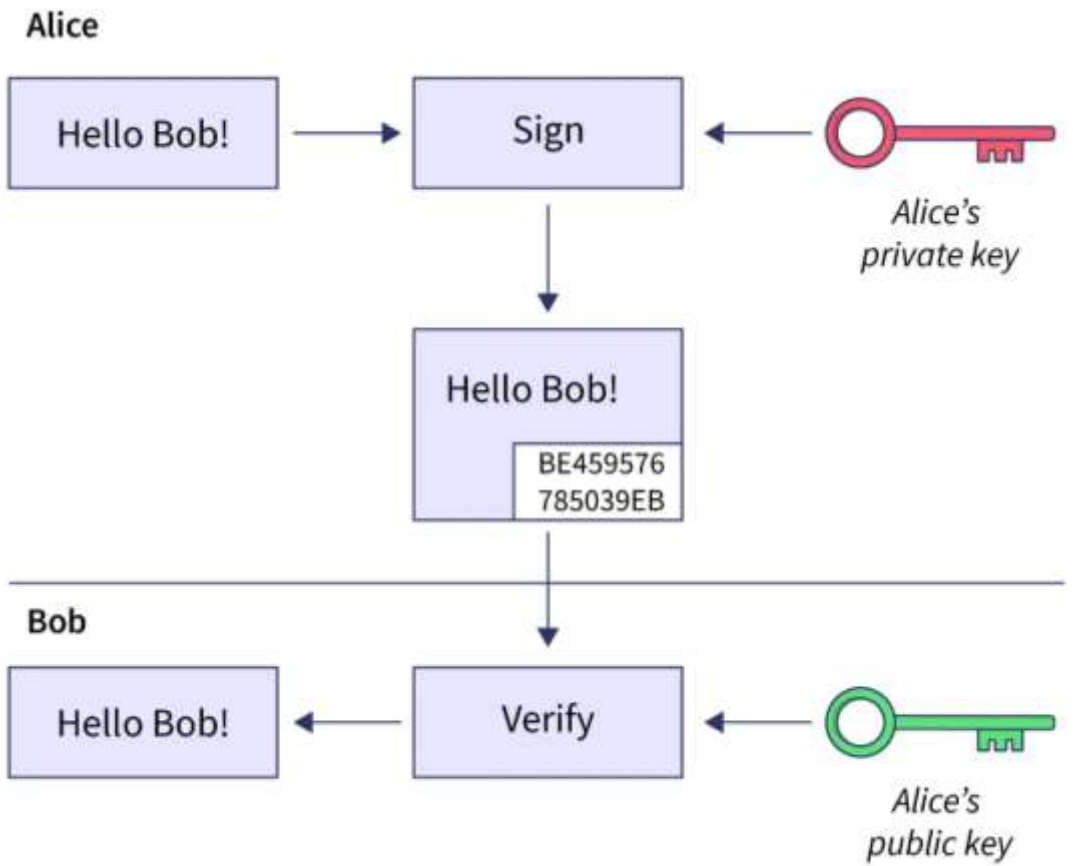
Federal Information Processing Standards Publication

Module-Lattice-Based Digital Signature Standard

3 Functions:
- (*pk, sk*) ← ML-DSA.KeyGen()
- (*σ*) ← ML-DSA.Sign(*sk, M, ctx*)
- True | False ← ML-DSA.Verify(*pk, M, σ, ctx*)

3 Security Categories:
- ML-DSA-44 (cat. 2 ≙ SHA3-256)
- ML-DSA-65 (cat. 3 ≙ AES-192)
- ML-DSA-87 (cat. 5 ≙ AES-256)

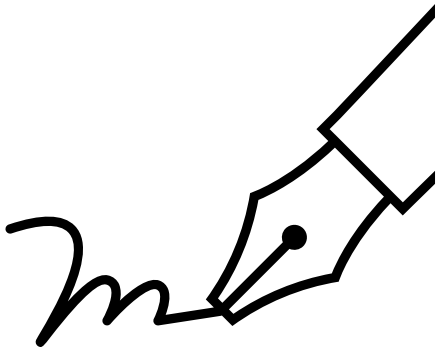
security strength categories



https://www.scaler.in/digital-signature-in-computer-network/

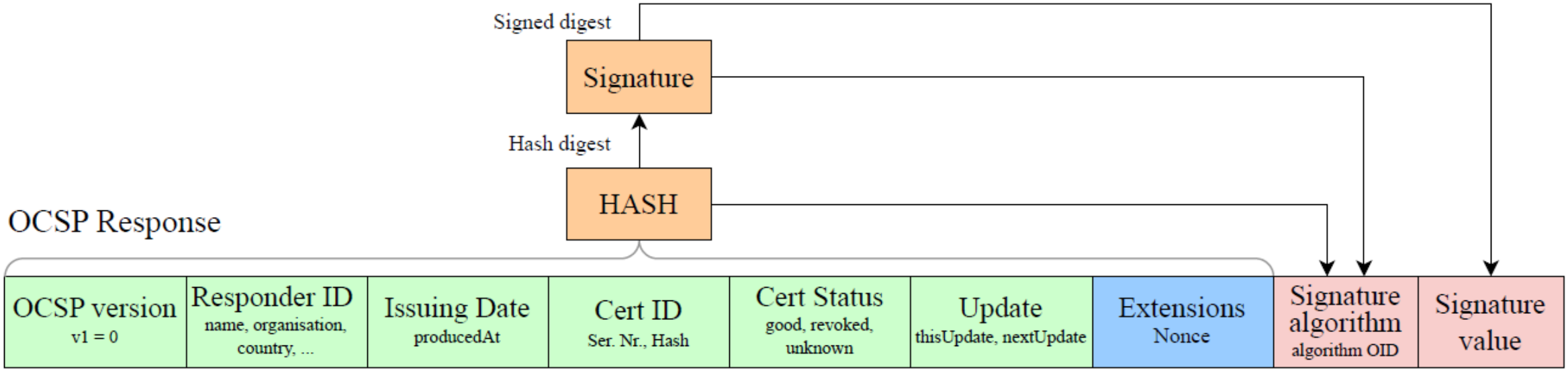
X.509 Certificate

Quantum-safe X.509 certificate → sign with ML-DSA

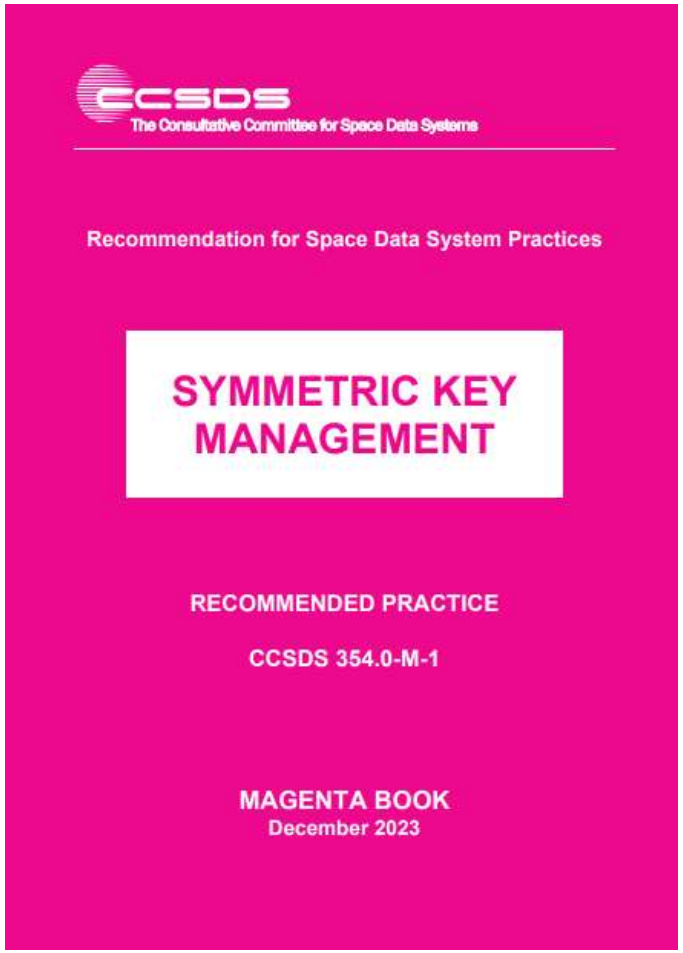


Certificate Revocation - OCSP

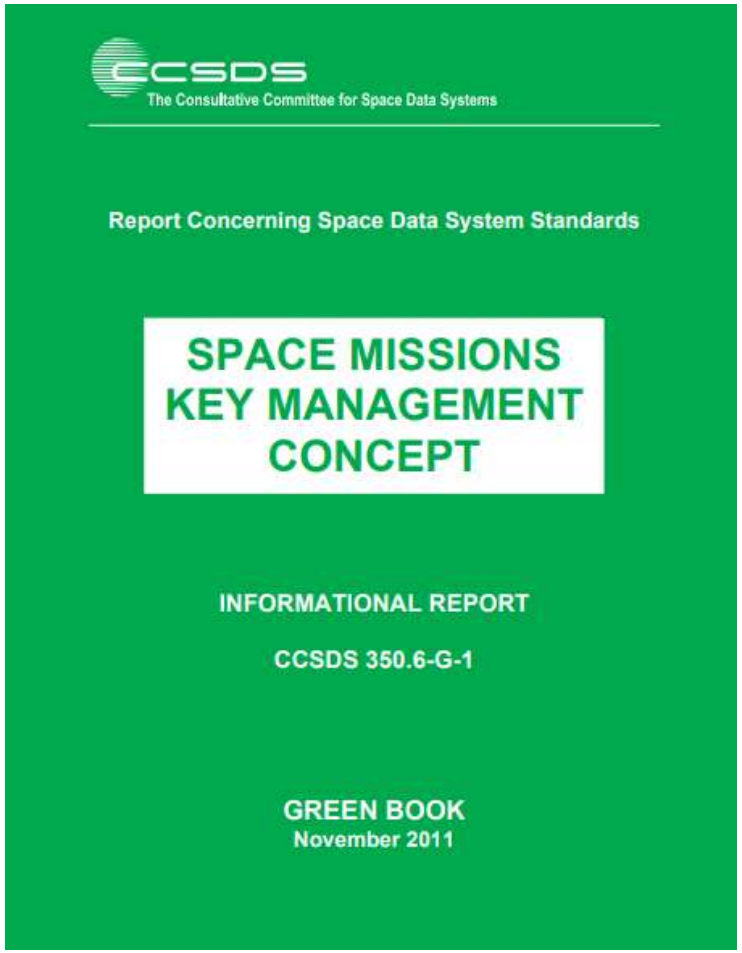
- **O**nline **C**ertificate **S**tatus **P**rotocol
- Well-established method for certificate revocation
- OCSP-Stapling leads to improvement
- Independent of revocation mechanism. We just need one!



CCSDS Standards

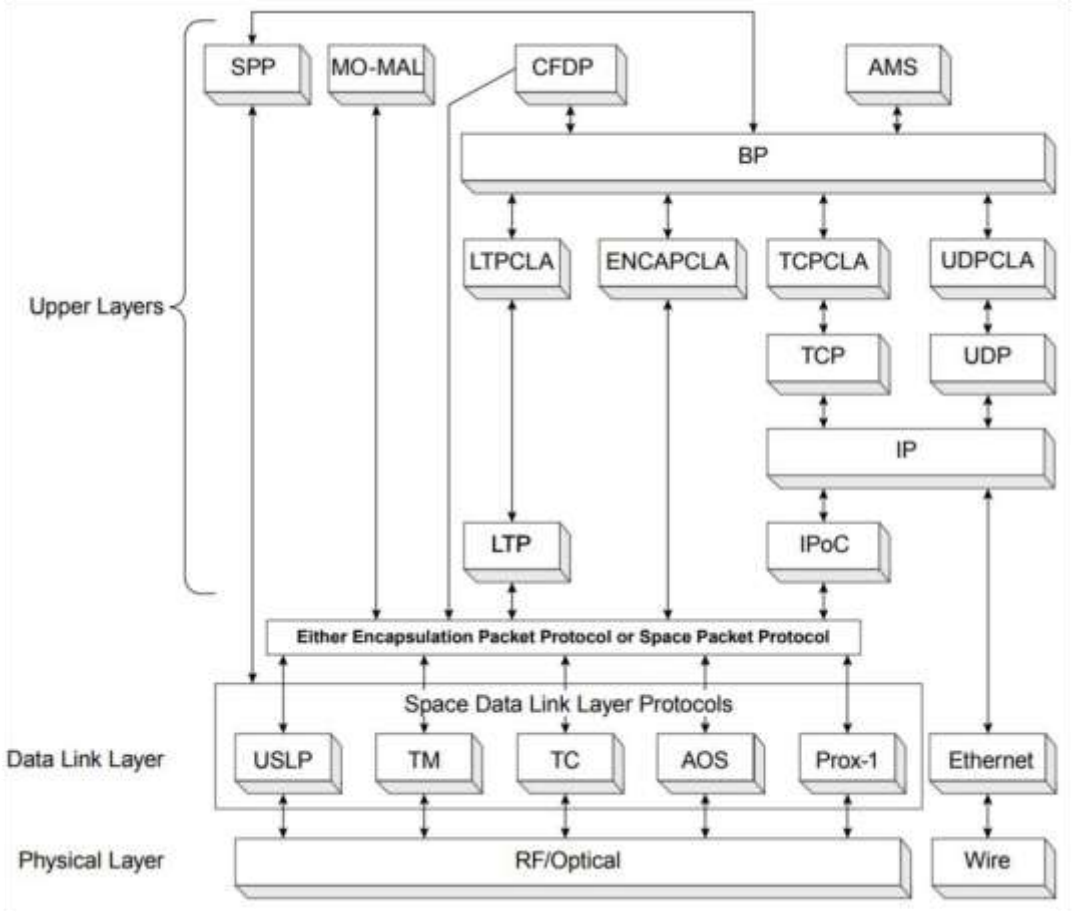


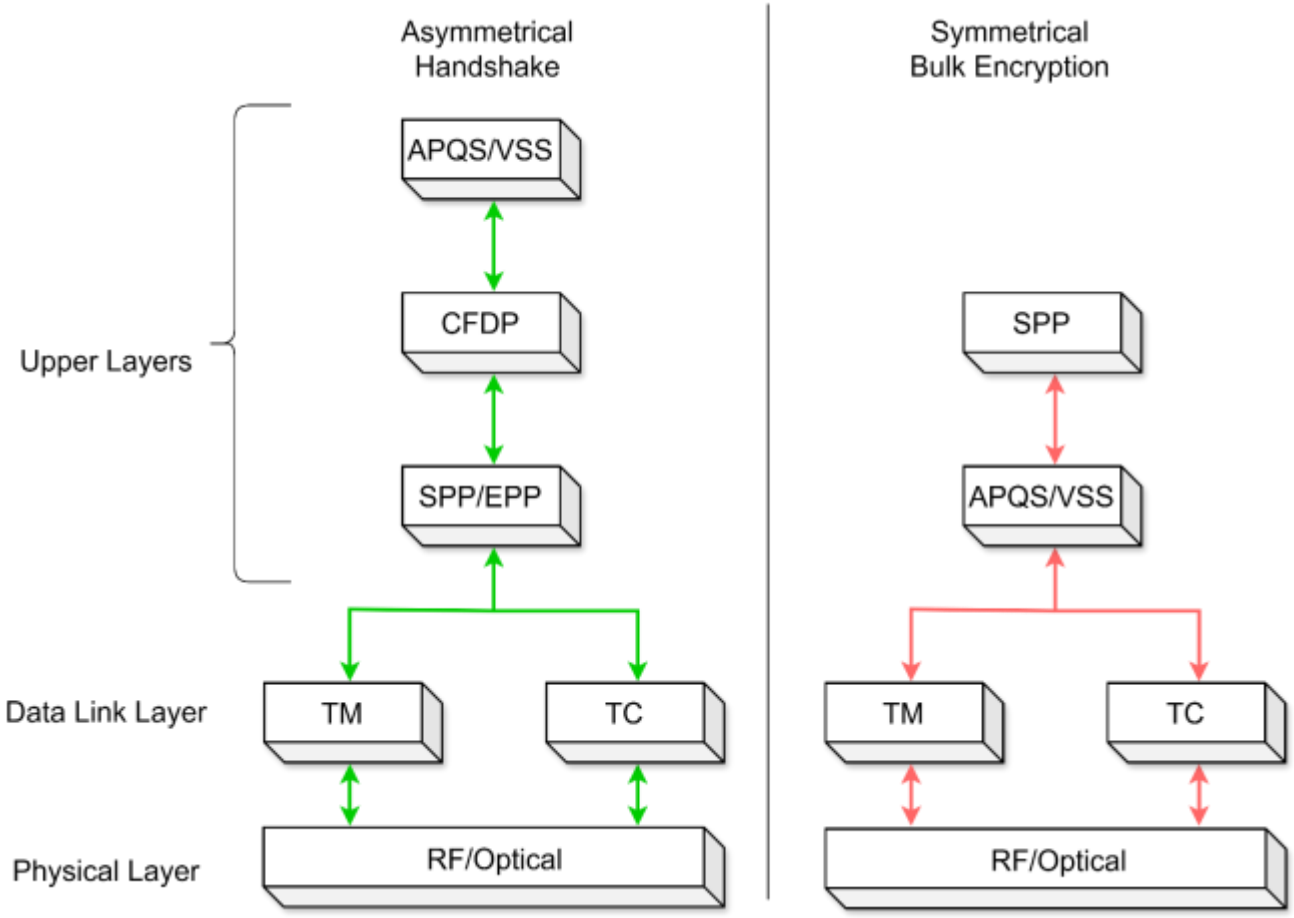
CCSDS 354.0-M-1



CCSDS 350.6-G-1

CCSDS Protocol Stack

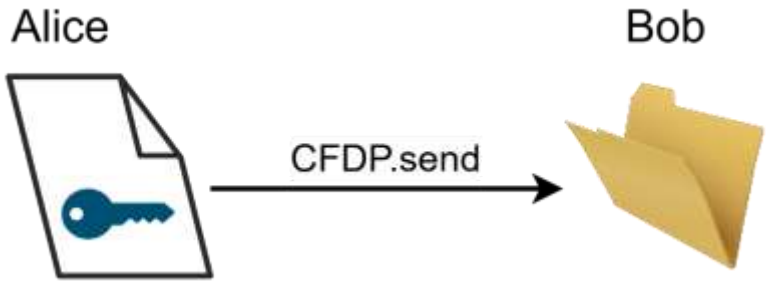


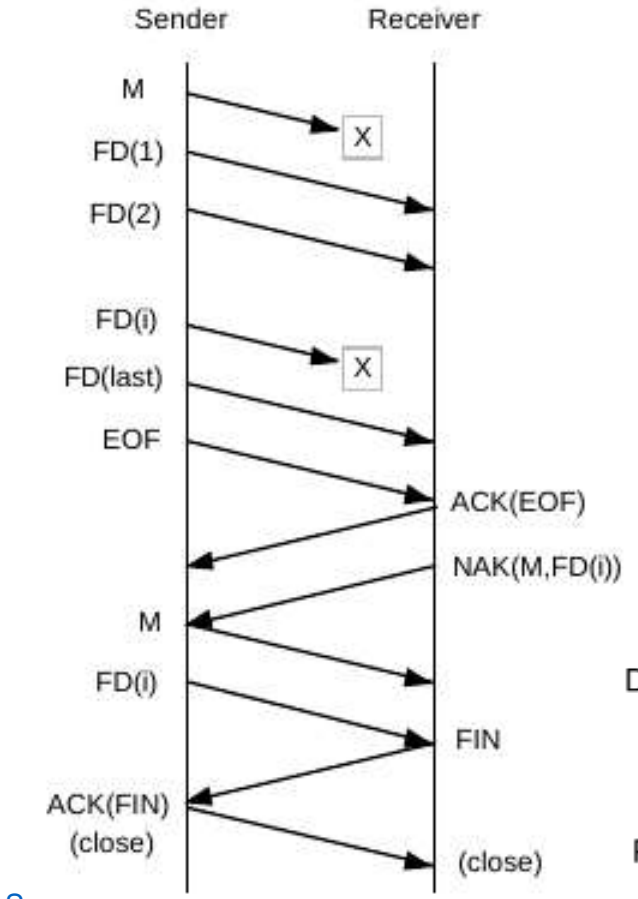


CCSDS 130.0-G-4

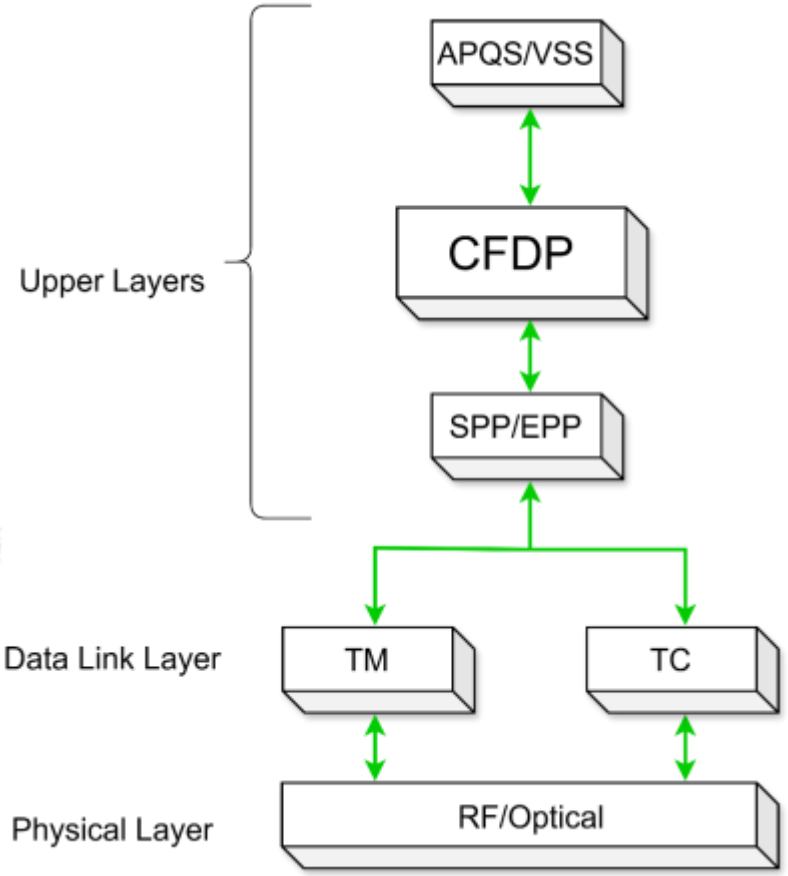
CFDP

- **C**CSDS **F**ile **D**elivery **P**rotocol
  - Complete protocol
  - Reliable transmission (NAK, EOF)
  - Independent of lower-level protocol



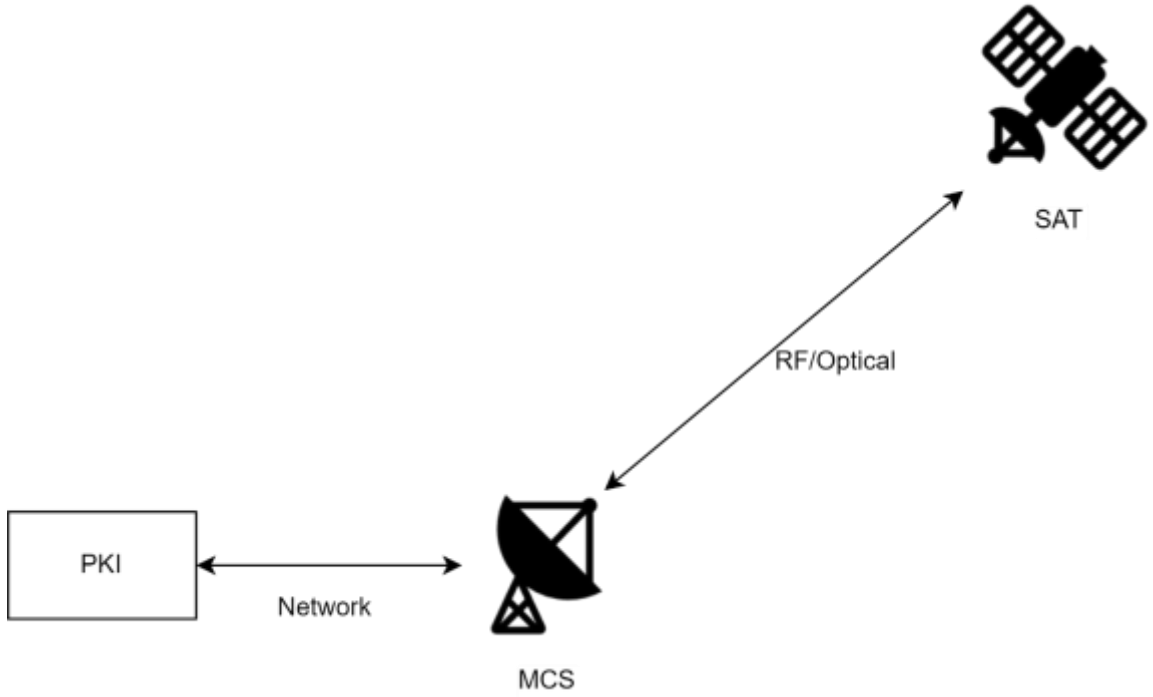


Source

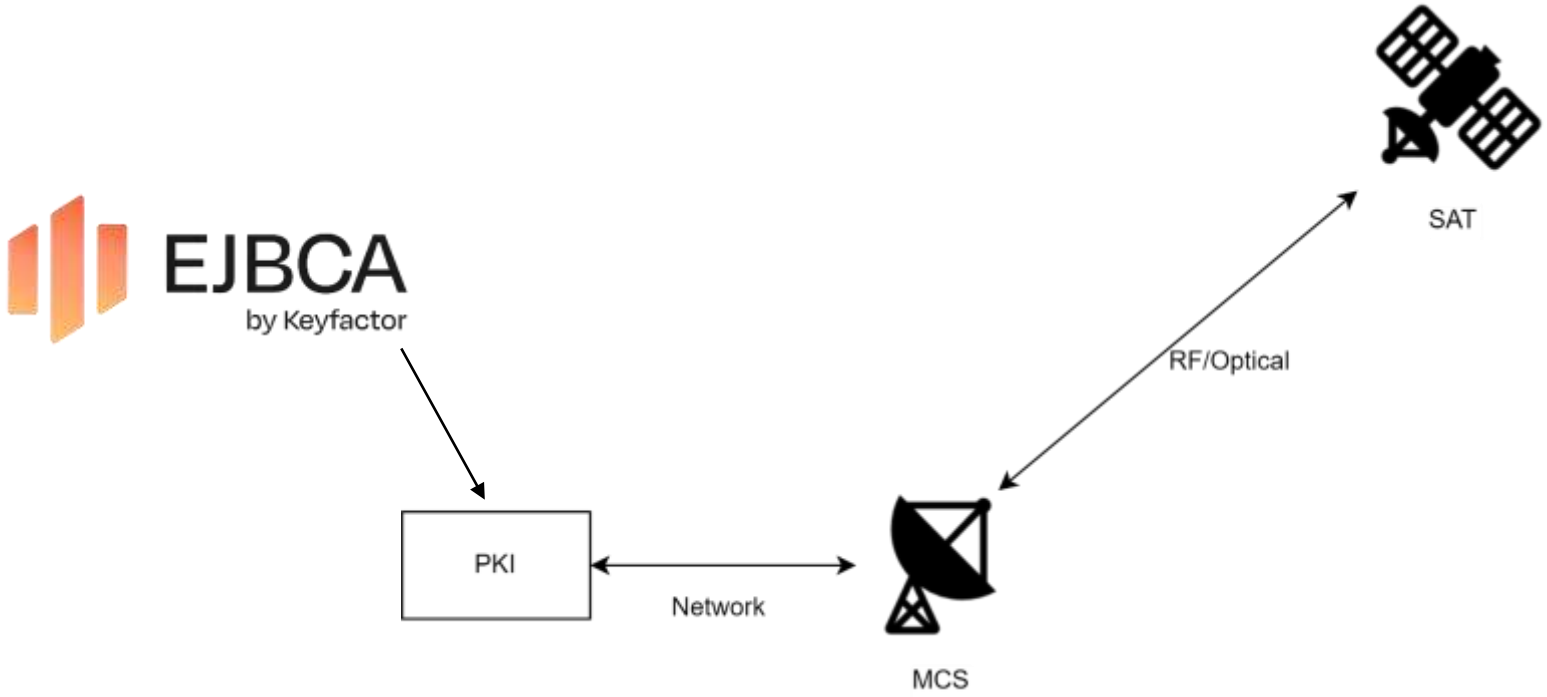


https://ccsds.org/Pubs/727x0b5e1.pdf

E2EQSS Architecture

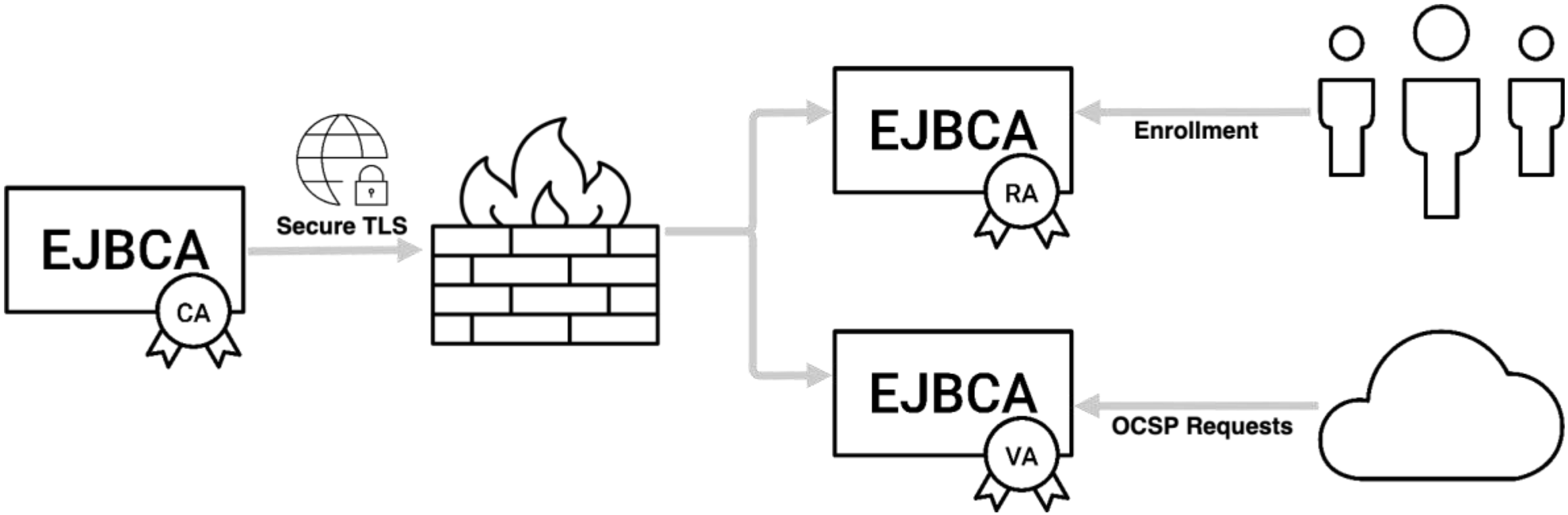


E2EQSS Architecture



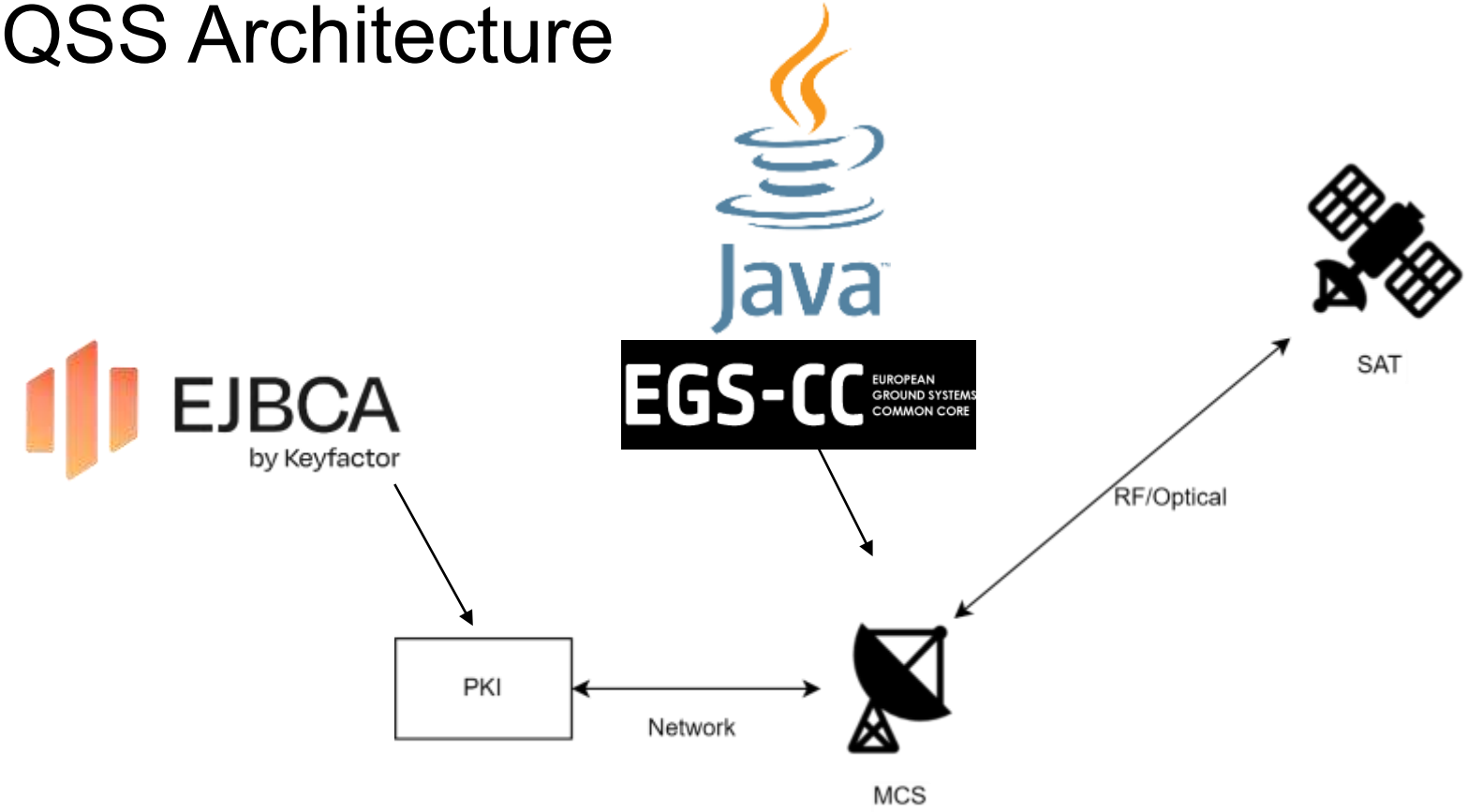
ejbca.org/

EJBCA



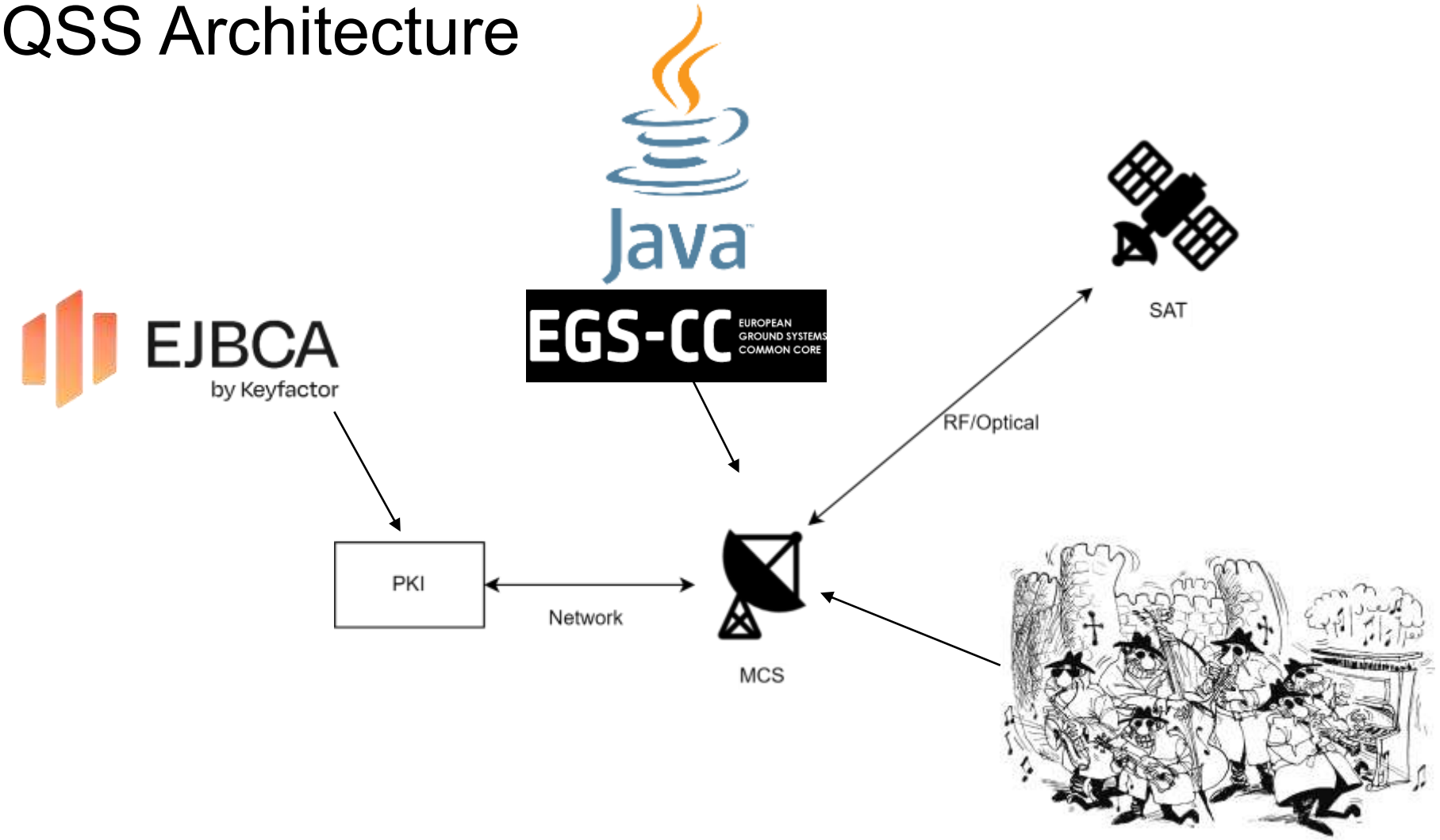
docs.keyfactor.com/ejbca/latest/ejbca-concepts

E2EQSS Architecture



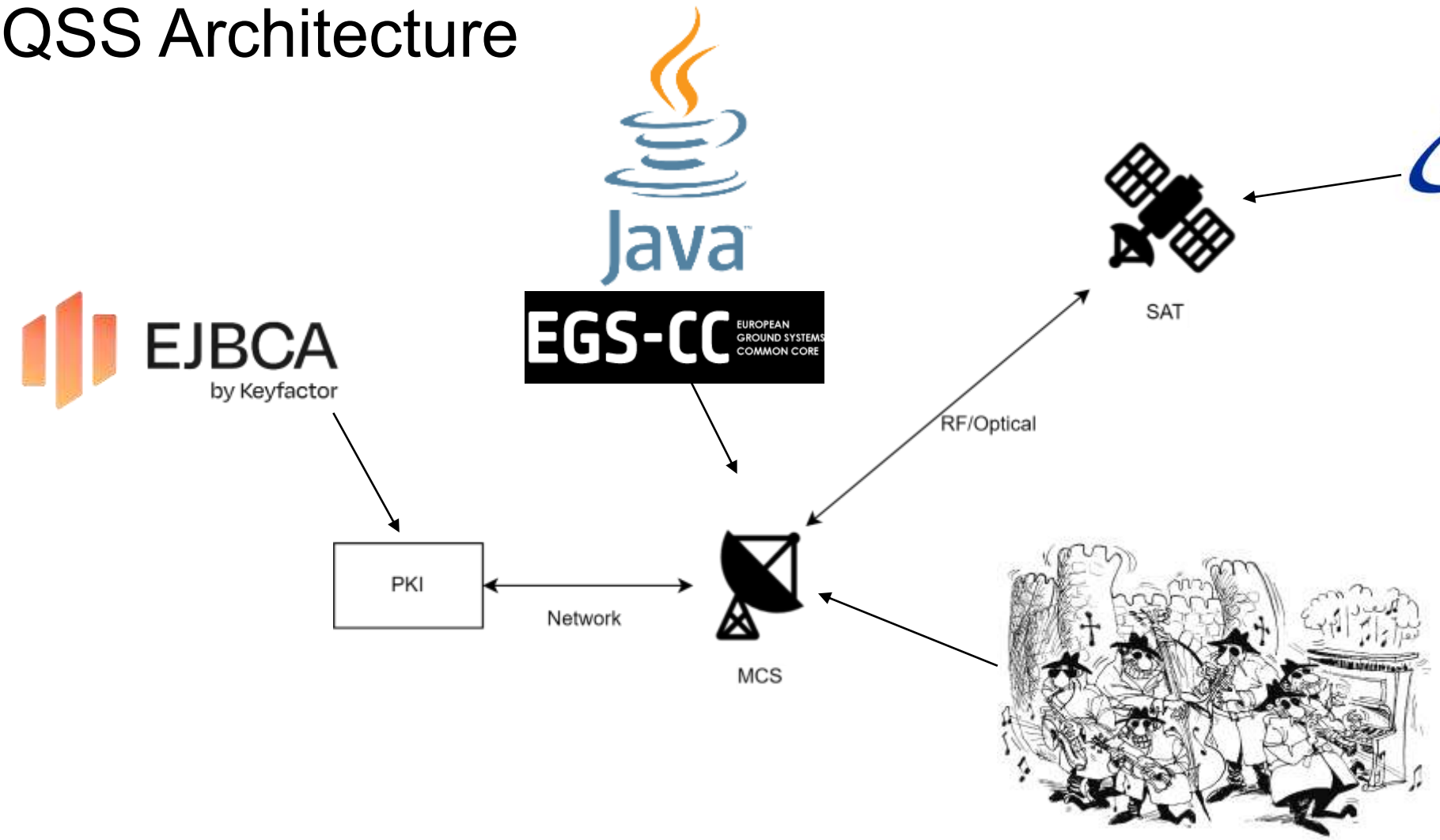
egscc.esa.int/

E2EQSS Architecture



bouncycastle.org/

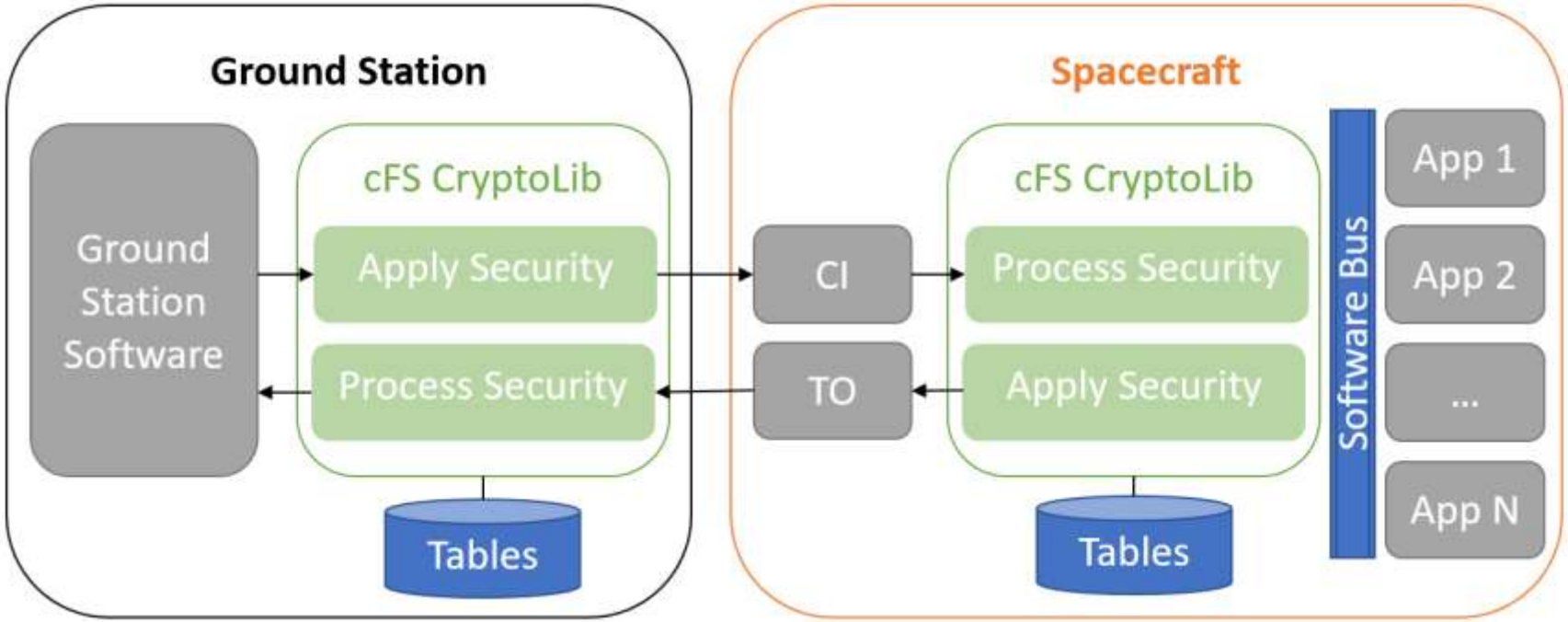
E2EQSS Architecture



etd.gsfc.nasa.gov/capabilities/core-flight-system/

Nasa cFS

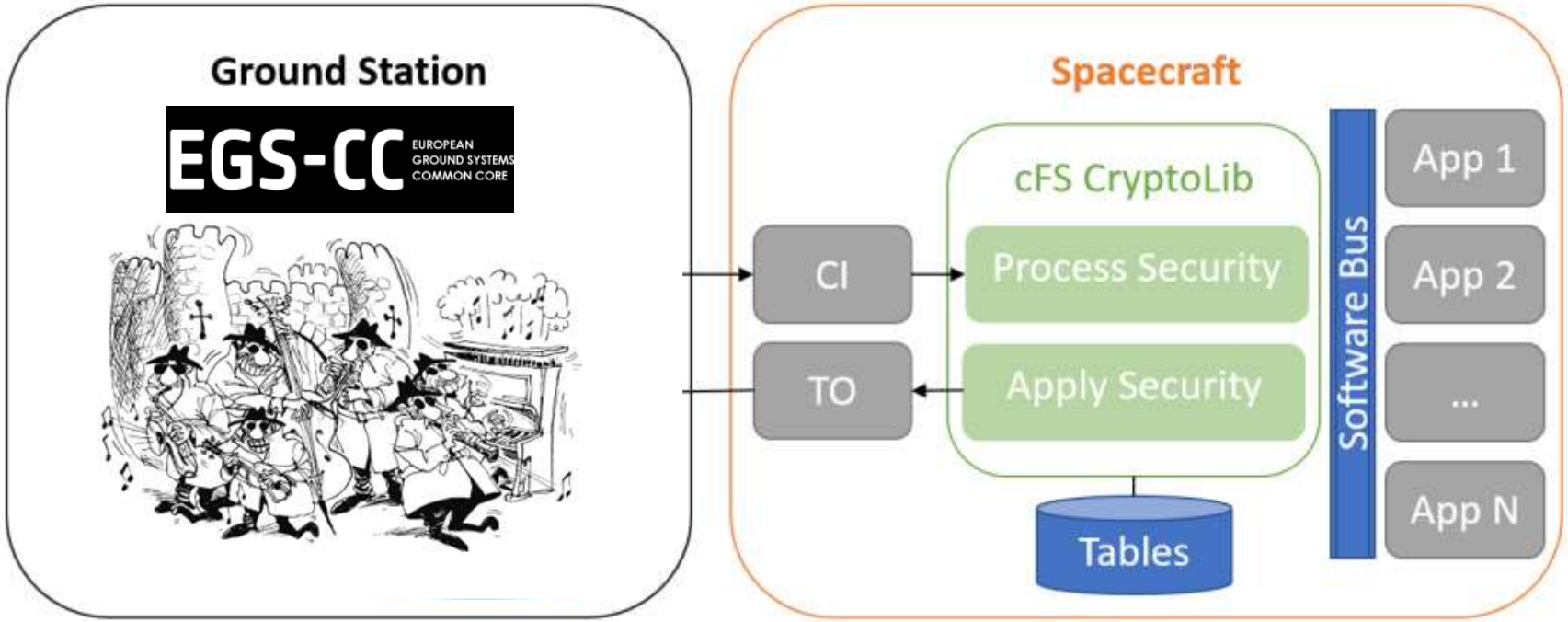
– CryptoLib



SSC23 NOS3 Design Reference Mission

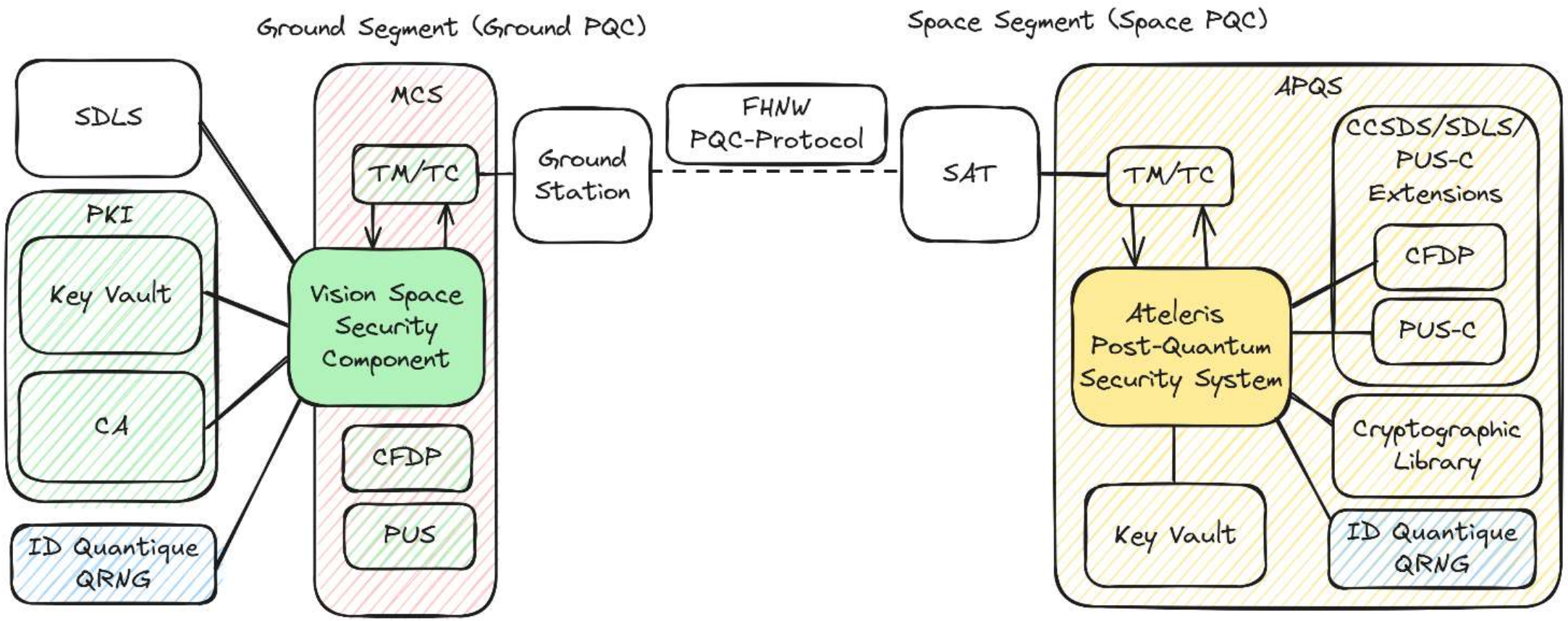
Nasa cFS

– CryptoLib



egscc.esa.int/
bouncycastle.org/
SSC23 NOS3 Design Reference Mission

E2EQSS Architecture



Quantum Random Number Generator

- ML-DSA and ML-KEM need randomness
- NIST certified entropy source
- The QRNG does **not** make ML-KEM and ML-DSA quantum-safe!

Algorithm 19 ML-KEM.KeyGen()

Generates an encapsulation key and a corresponding decapsulation key.

Output: encapsulation key ek $\in \mathbb{B}^{384k+32}$.
Output: decapsulation key dk $\in \mathbb{B}^{768k+96}$.

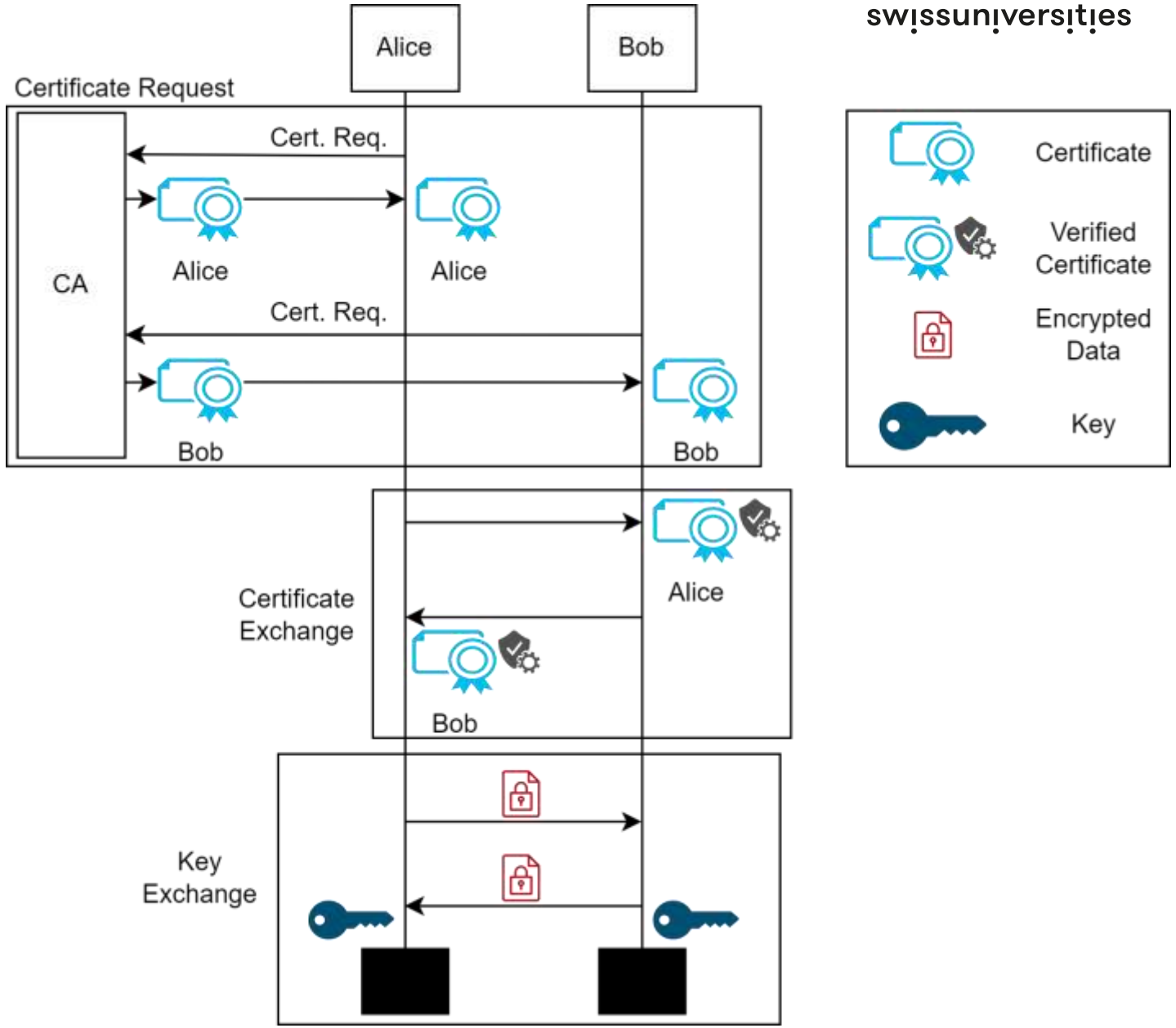
1: $d \xleftarrow{\$} \mathbb{B}^{32}$ ▷ d is 32 random bytes (see Section 3.3)
2: $z \xleftarrow{\$} \mathbb{B}^{32}$ ▷ z is 32 random bytes (see Section 3.3)
3: **if** $d ==$ NULL **or** $z ==$ NULL **then**
4: **return** $\perp$ ▷ return an error indication if random bit generation failed
5: **end if**
6: $(\text{ek}, \text{dk}) \leftarrow$ ML-KEM.KeyGen_internal(d, z) ▷ run internal key generation algorithm
7: **return** (ek, dk)

csrc.nist.gov/projects/cryptographic-module-validation-program/entropy-validations/certificate/63

Protocol

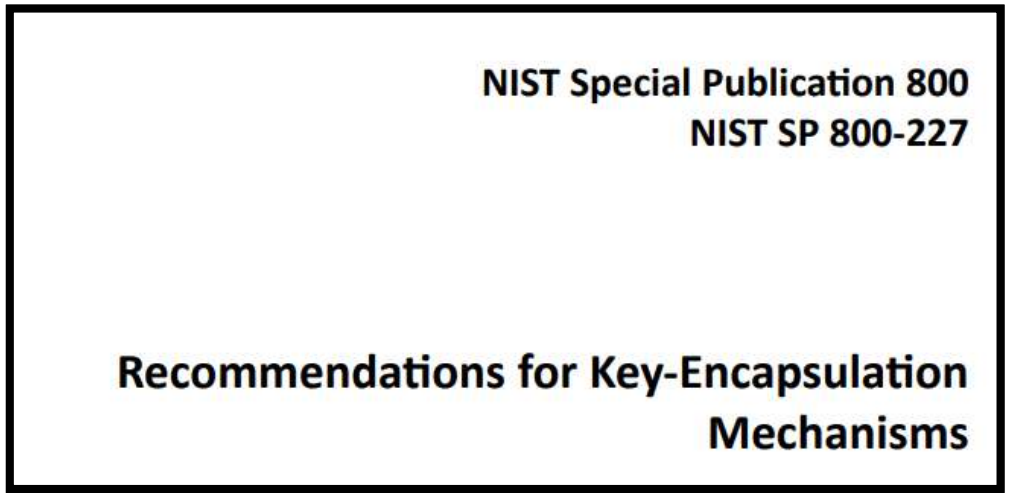
Quantum-Safe protocol with:

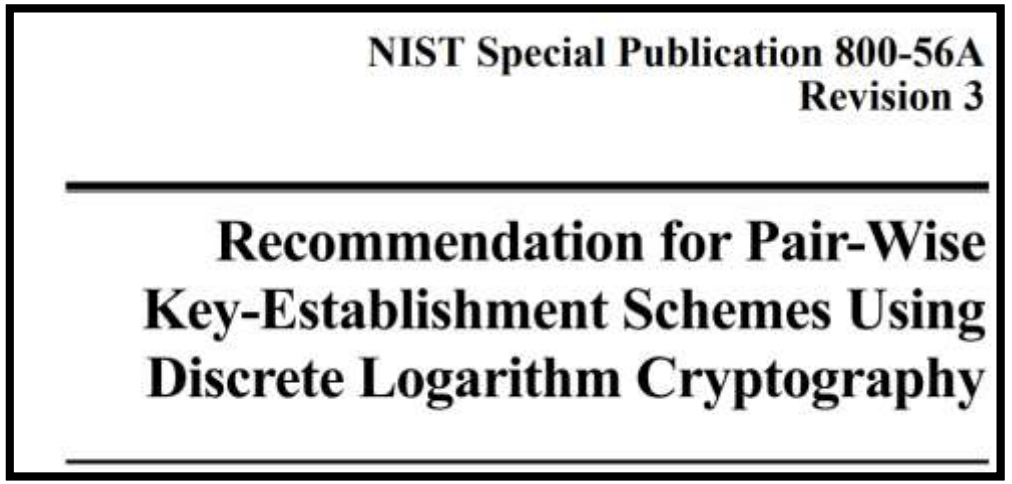
- Authentication
- Hybrid cryptography
- Forward secrecy
- Crypto-agility



Hybrid Key Exchange

- 4 Keys:
  - 2 long-term, static ML-KEM keys
  - 1 short-term, ephemeral ML-KEM key
  - 1 short-term, ephemeral ECDH key





Key Derivation

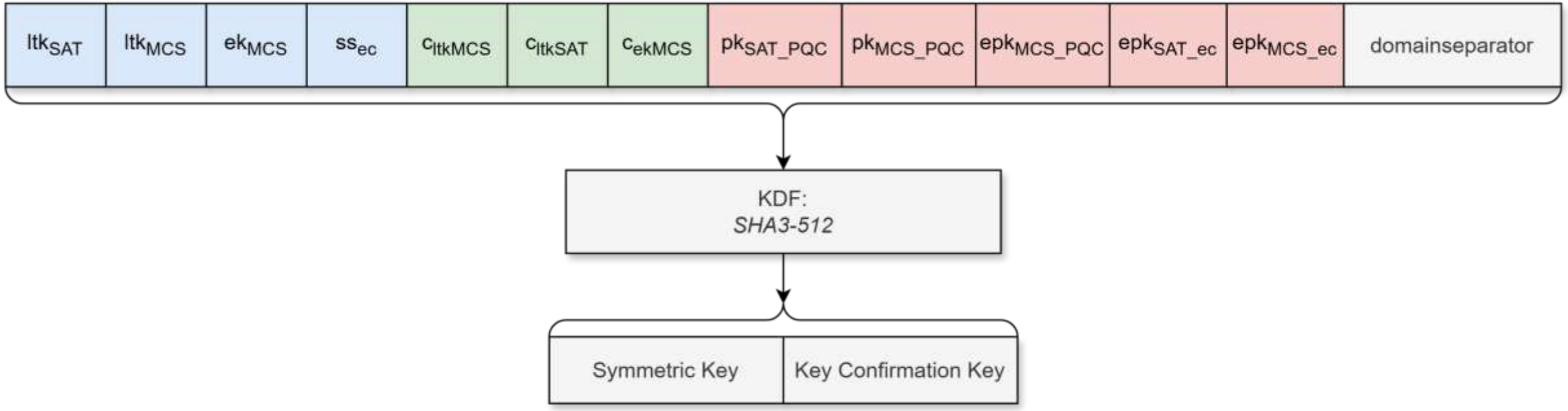
NIST Special Publication 800-56C
Revision 2

Recommendation for Key-Derivation Methods in Key-Establishment Schemes

NIST Special Publication 800
NIST SP 800-227

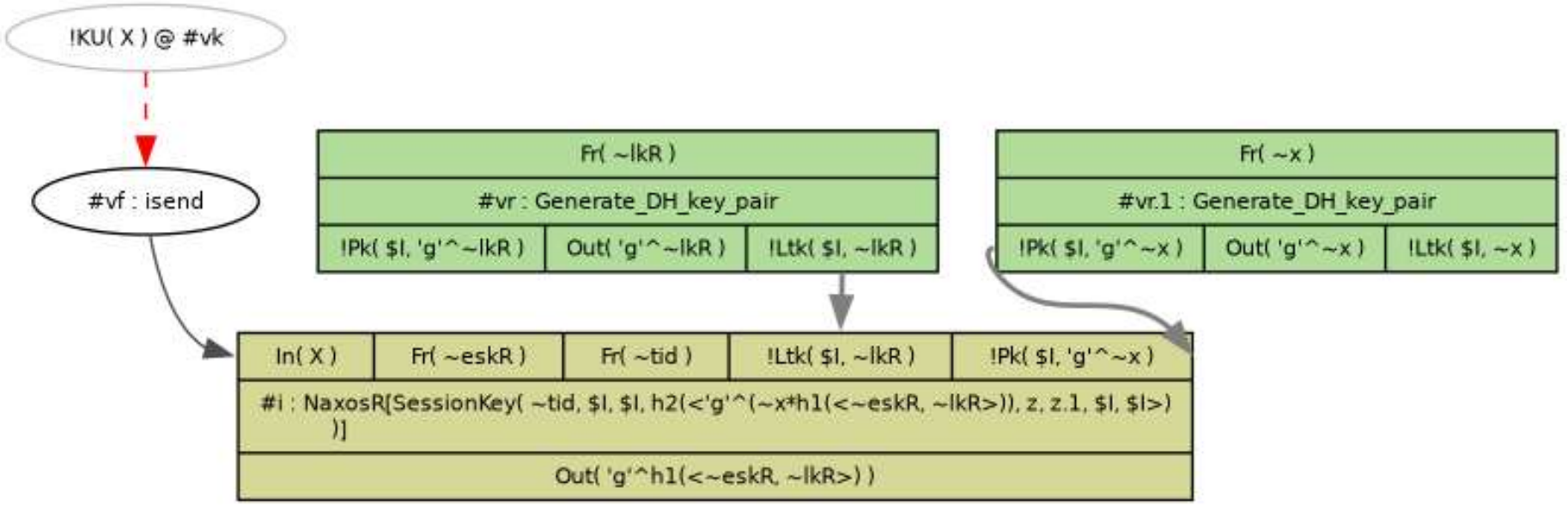
Recommendations for Key-Encapsulation Mechanisms

- Key Derivation Function: SHA3-512
- Key confirmation step



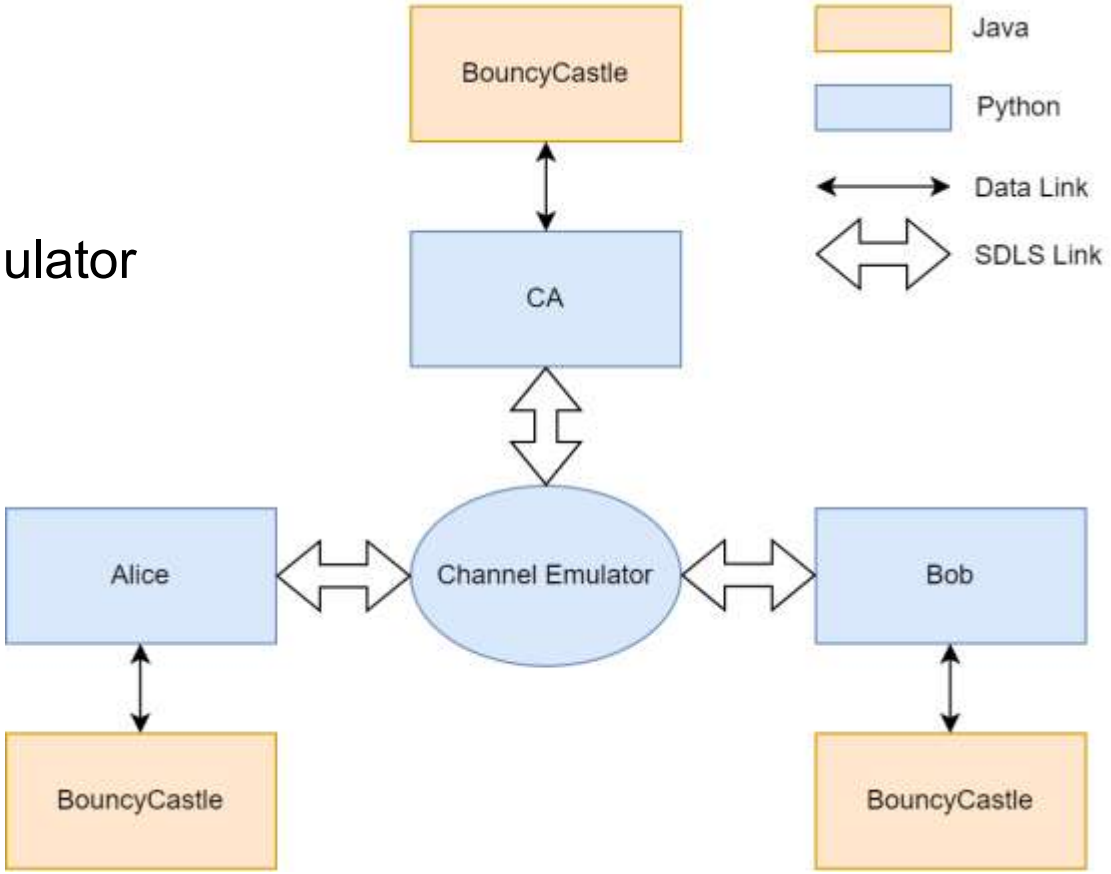
Tamarin Prover

- Symbolic protocol analyzer
- Simulates protocol-level attacks



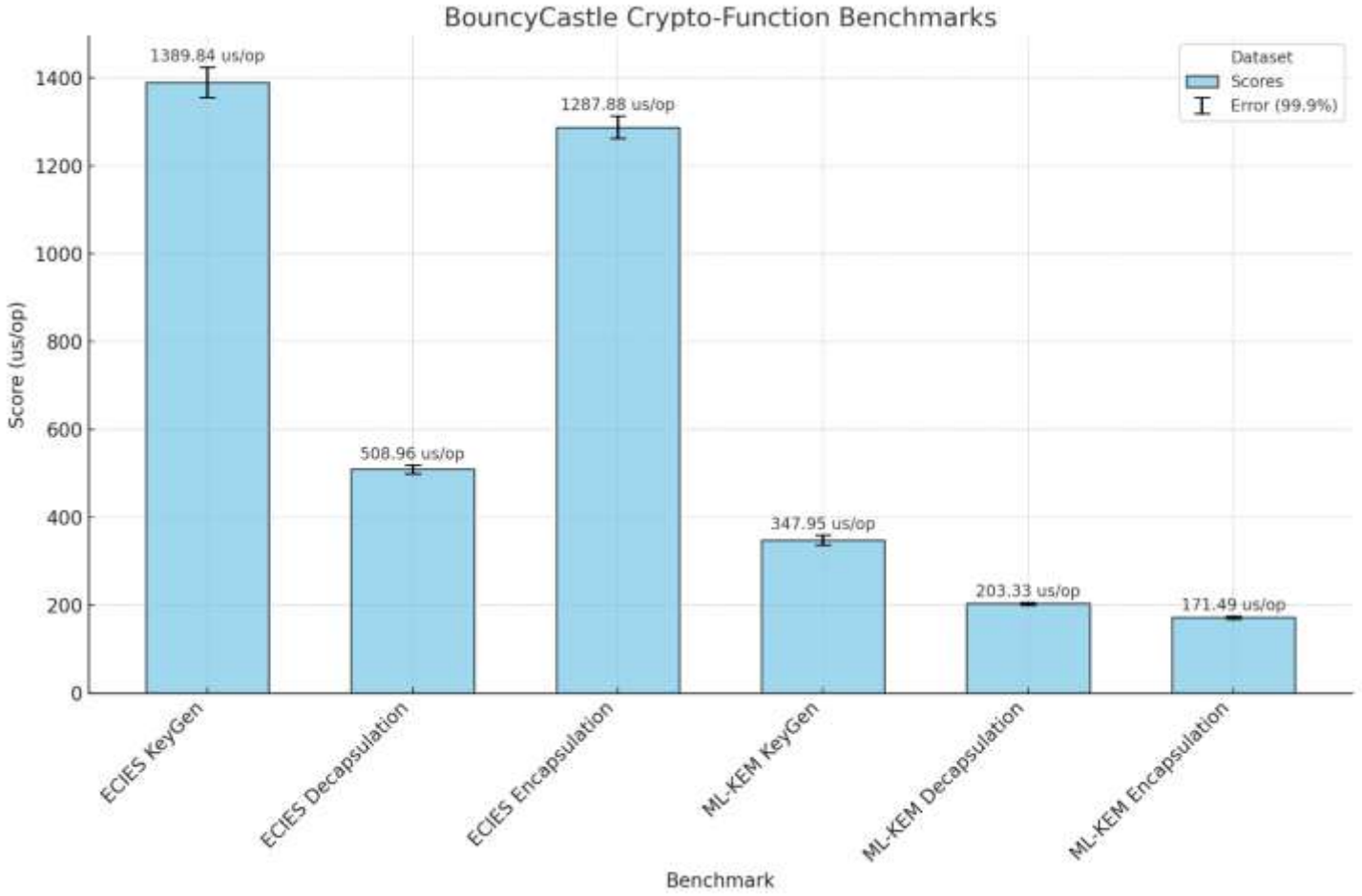
Python Simulator

- Simulates the protocol
- Proof of concept
- Includes a channel emulator

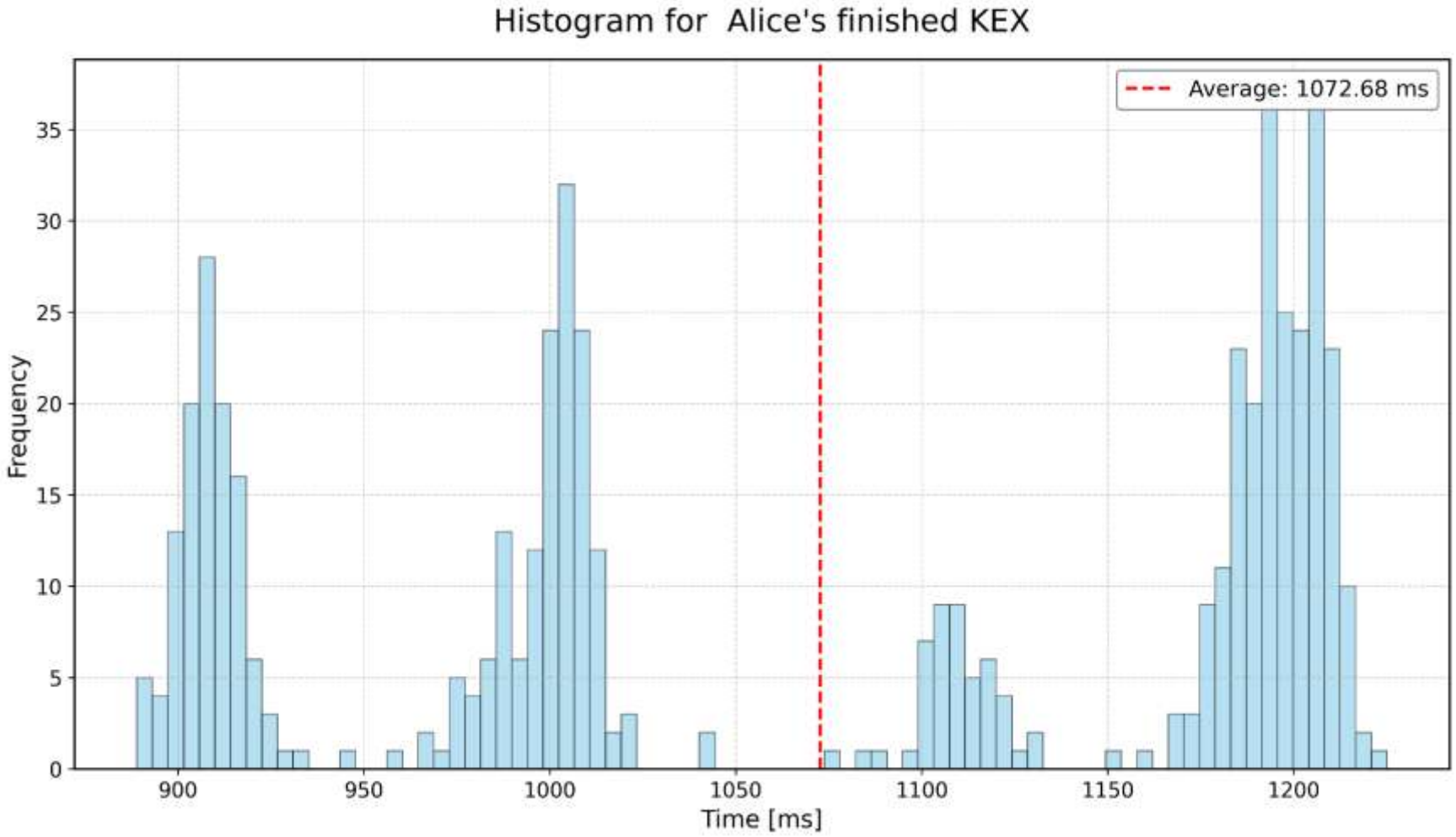


Python Simulator

- **Python Benchmarks (ms/op):**
  - KeyGen: 3.31
  - Encap: 4.48
  - Decap: 6.14



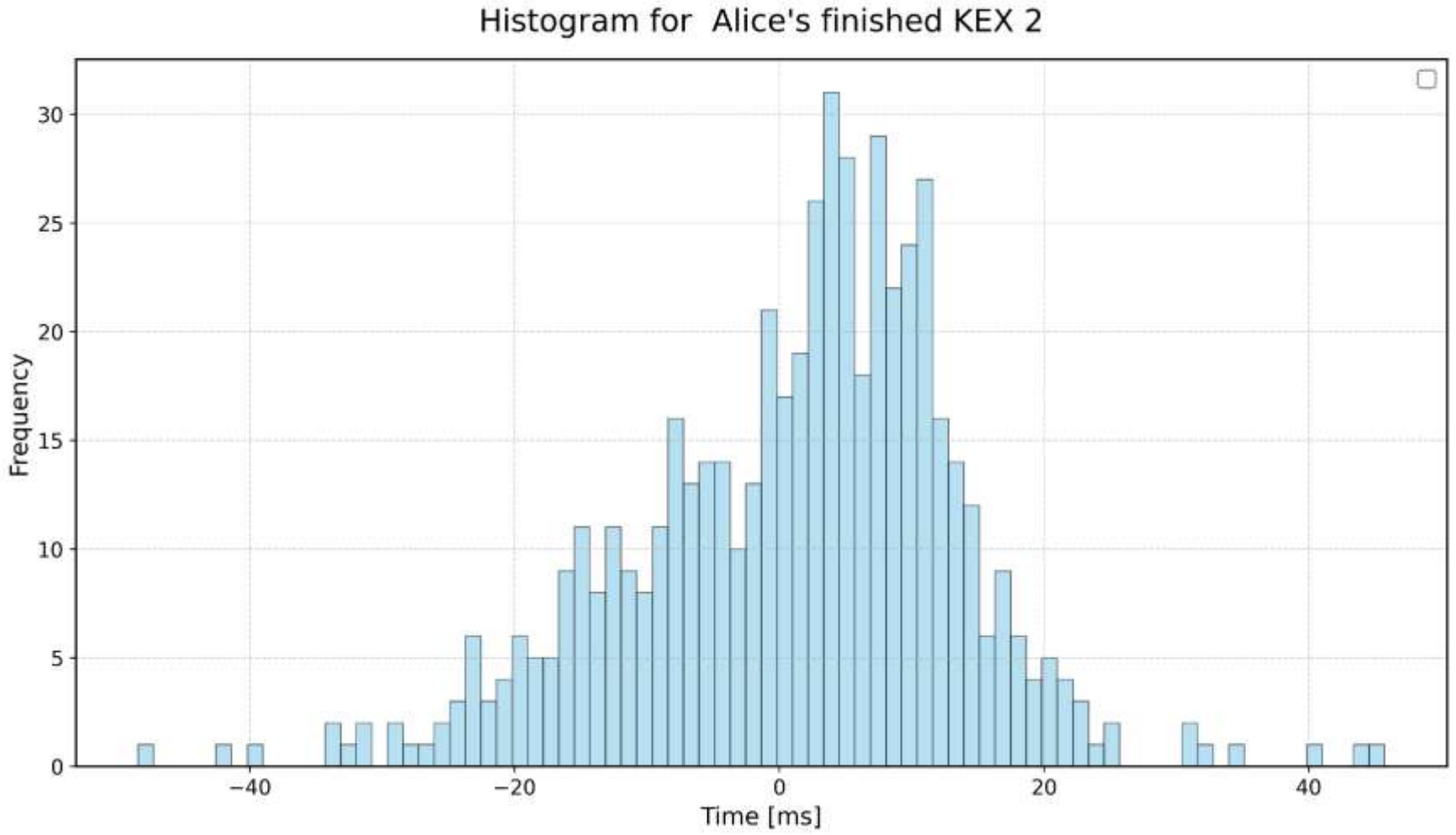
Python Simulator



Questions:

Contact: christoph.wildfeuer@fhnw.ch

Python Simulator



ML-KEM Hardware Accelerator

Implementation	#LUT / #DSP / #BRAM	NAC	#kCycle (KeyGen/Encaps/Decaps)	Time [μs] (KeyGen/Encaps/Decaps)
Our ML-KEM	7356/4/6.5	8'536	26/24/31	130/121/155
Area Efficient [1]	7412/2/3	7'972	6.3/7.9/10.2	39.2/47.6/61.3
Direct Impl. [2]	97k/36/200	124k	-/77/102	-/500/659
High Perf. [3]	10.4k/6/8.5	12'020	2.7/3.9/5.0	12.3/17.7/22.9
More generic [4]	14k/11/14	16'780	112/177/191	4461/7102/7623
Co-Processor [5]	25k/0/2	25'200	5.5/66/8.0	36.4/44.1/53.6
RISC-V [6]	24k/18/32	29'640	273/325/340	-
HLS [7]	1978k/-/-	1978k	-	-

Sources:

[1] Y. Xing and S. Li, „A compact hardware implementation of cca-secure key exchange mechanism crystals-kyber on fpga“, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 328–356, 2021

[2] Y. Huang, M. Huang, Z. Lei, and J. Wu, „A pure hardware implementation of crystalskyber pqc algorithm through resource reuse“, *IEICE Electronics Express*, vol. 17, no. 17, pp. 20 200 234–20 200 234, 2020

[3] V. B. Dang, K. Mohajerani, and K. Gaj, „High-speed hardware architectures and fpga benchmarking of crystals-kyber, ntru, and saber“, IEEE Transactions on Computers, vol. 72, no. 2, pp. 306–320, 2022

[4] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan, „Sapphire: A configurable cryptoprocessor for post-quantum lattice-based protocols“, arXiv preprint arXiv:1910.07557, 2019

[5] S. S. Roy and A. Basso, „High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: Saber in hardware“, IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 443–466, 2020

[6] T. Fritzmann, G. Sigl, and J. Sepúlveda, „Risq-v: Tightly coupled risc-v accelerators for post-quantum cryptography“, IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 239–280, 2020.

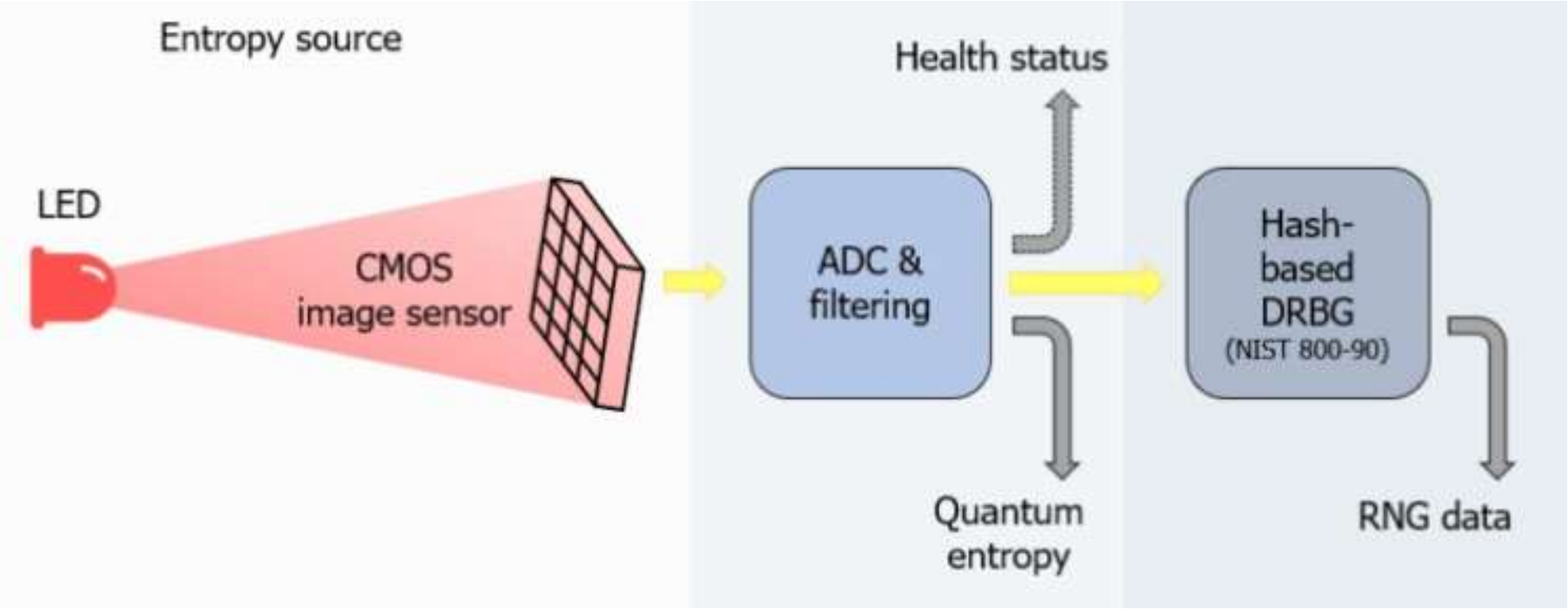
[7] K. Basu, D. Soni, M. Nabeel, and R. Karri, „Nist post-quantum cryptography-a hardware evaluation study“, *Cryptology ePrint Archive*, 2019

Normalized Area Cost

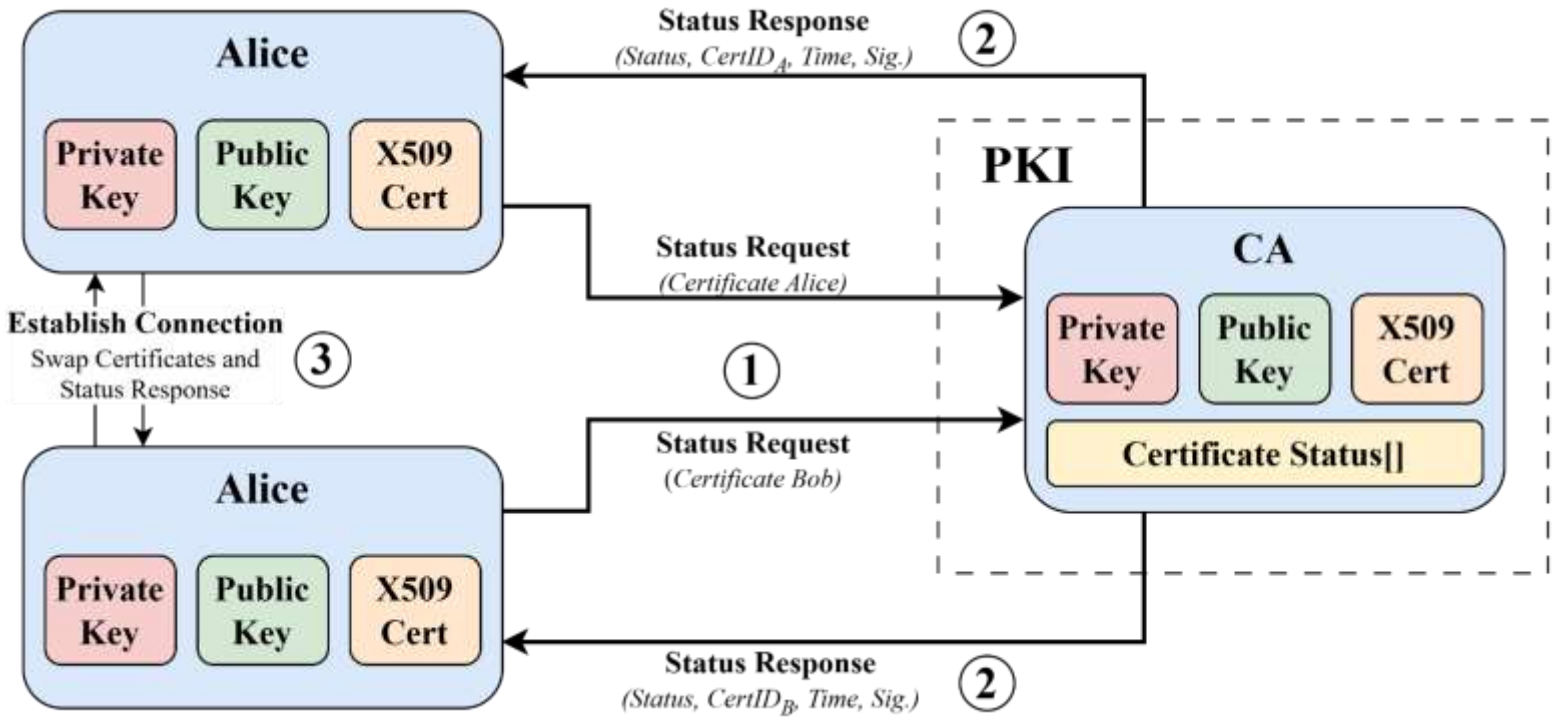
$$\mathrm{NAC} = \#\mathrm{LUT} + 100 \cdot \#\mathrm{DSP} + 120 \cdot \#\mathrm{BRAM}$$

Core / Submodule	#LUTs	#DSPs	#BRAMs	NAC
NTT	484	2	0	684
Poly Arithmetic	443	1	0	543
Keccak and Sampling	4743	0	2	4983
Non Polynomial	1222	1	2	1562
L2 Memory and DMA	150	0	1.5	330
Control Logic	314	0	1	434
Total	7356	4	6.5	8536
[%]	14%	1.8%	4.6%	-

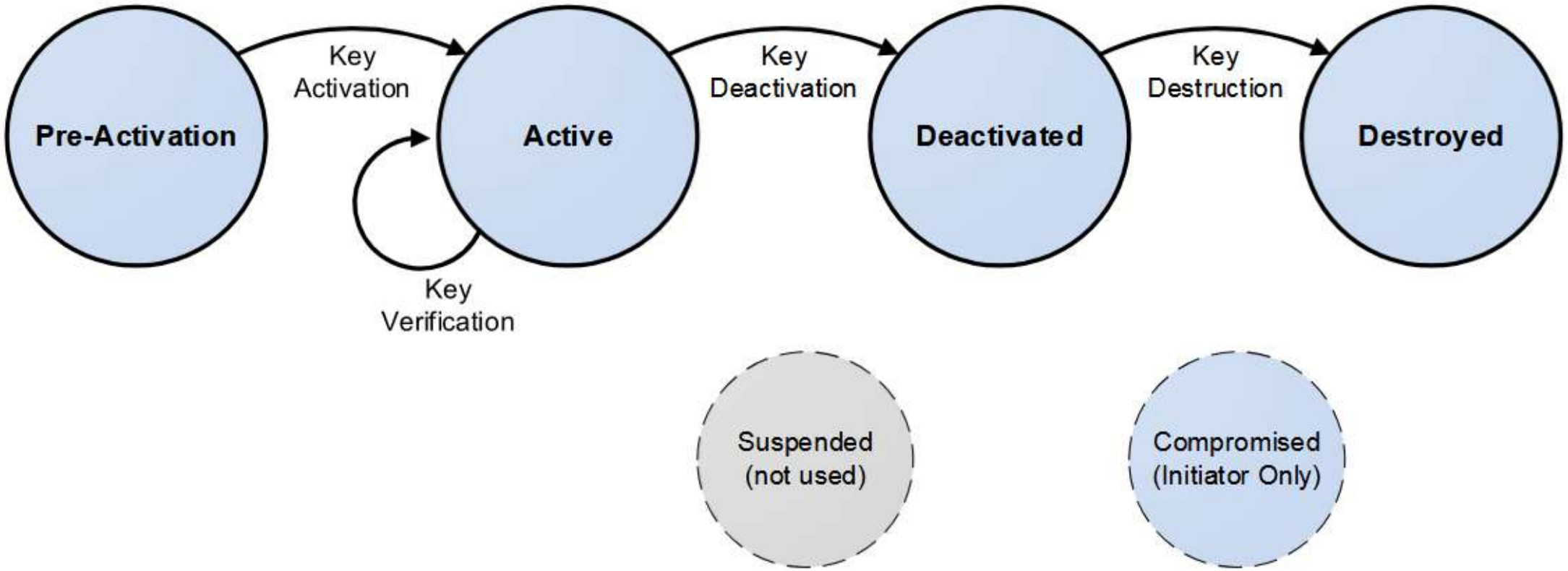
QRNG Architecture



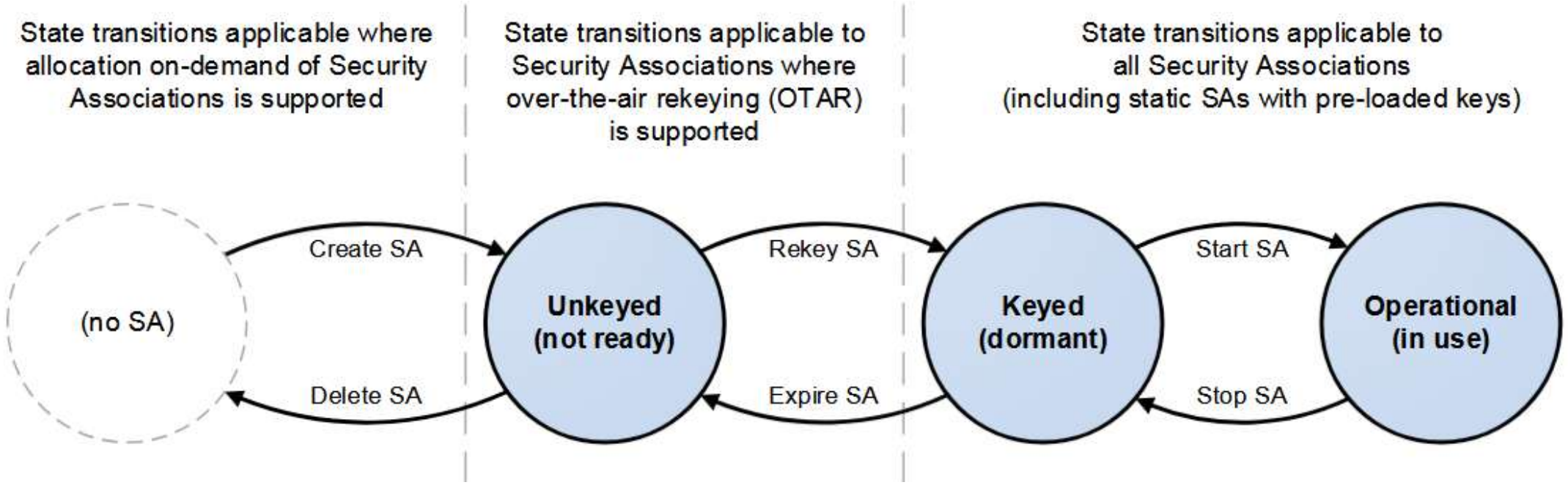
OCSP-Stapling



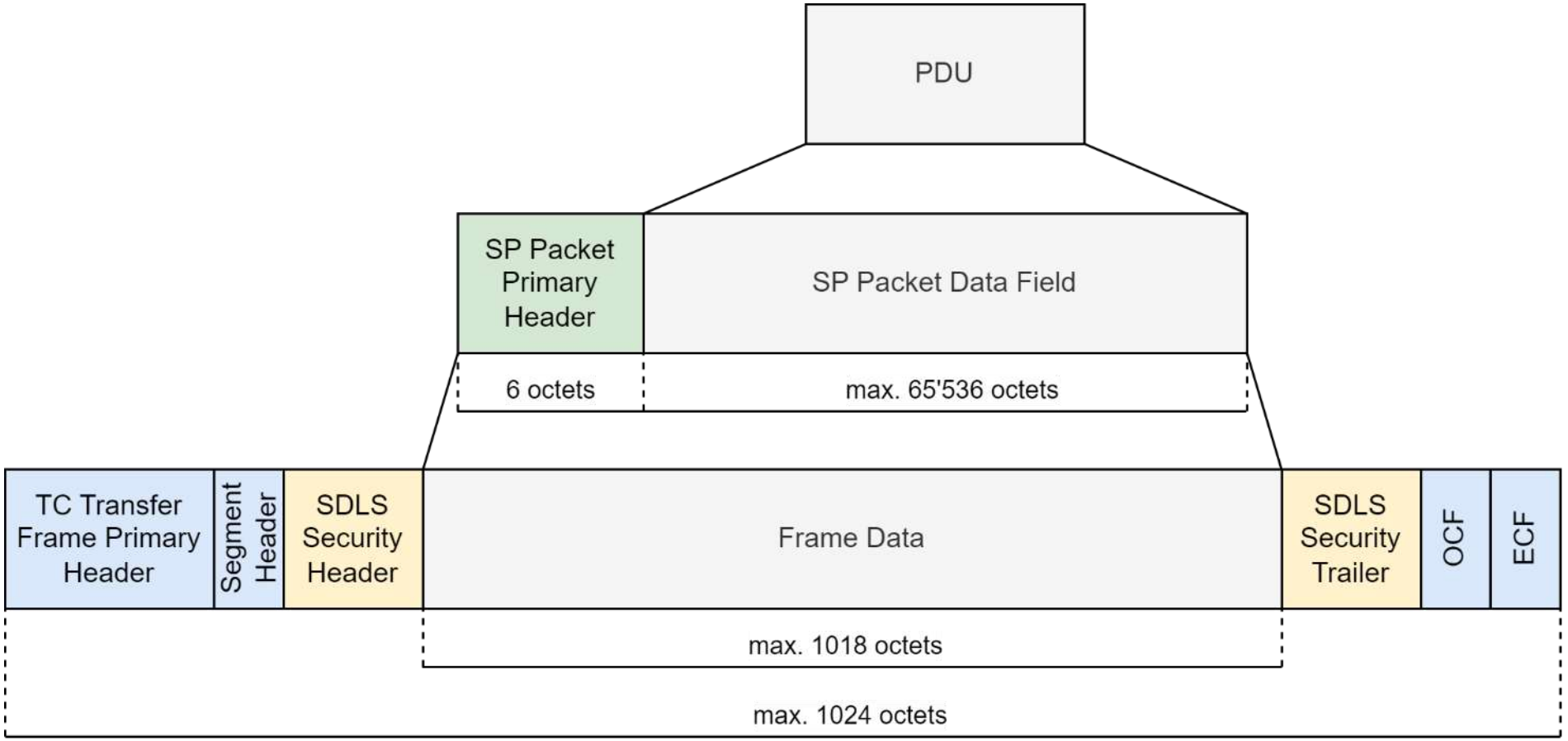
SDLS Key States



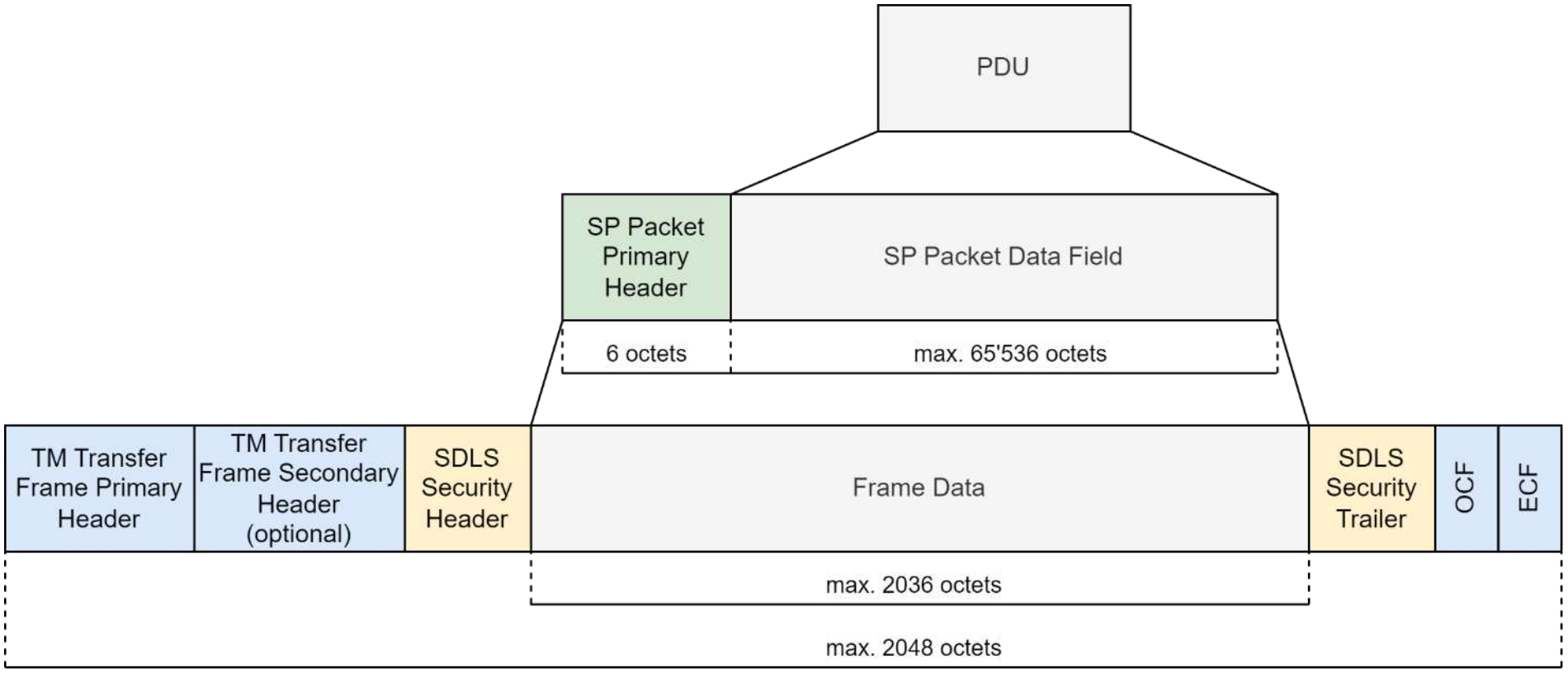
SDLS SA States



SDLS TC Frame



SDLS TM Frame



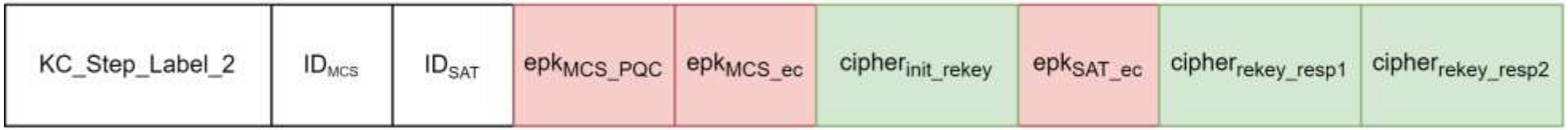
Key Confirmation

- MAC generation: SHA3-384
- Key confirmation key = ½ Symmetrical Key

NIST Special Publication 800
NIST SP 800-227

Recommendations for Key-Encapsulation Mechanisms

SAT KC MAC Data



GS KC MAC Data