ETSI The Standards People

IQC Institute for Quantum Computing

ETSI/IQC Quantum Safe Cryptography Conference 2026

Trilithium: two-party ML-DSA signing for future digital society

Presented by: Peeter Laud

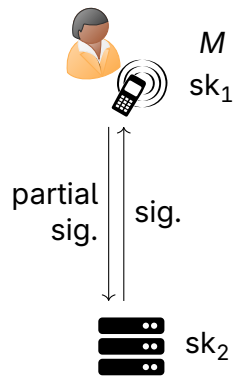
CYBERNETICA

Trilithium: two-party ML-DSA signing for future digital society

Peeter Laud

Smart-ID service and SplitKey technology

- Two-party **RSA** signing
  - "server-supported": phone initiates the signing
- Results in standard PKCS #1 signatures
  - Indistinguishable from non-threshold signatures
    - (except for double length)
- QSCD
- Used in EE, LV, LT since 2016. 3.7M users out of 6.1M population
  - later also in IS, BE



Why threshold cryptography?

- SplitKey development was instigated by the expected obsolesence of Mobile-ID
  - Mobile-ID: store private key in SIM card. Obtain HW protection for it
  - Mid-2010s: eSIMs were going to take over the world, very soon
- Now: a powerful tool for **digital sovereignty**
  - Who does your digital infrastructure depend on? Can use “OR” in the answer

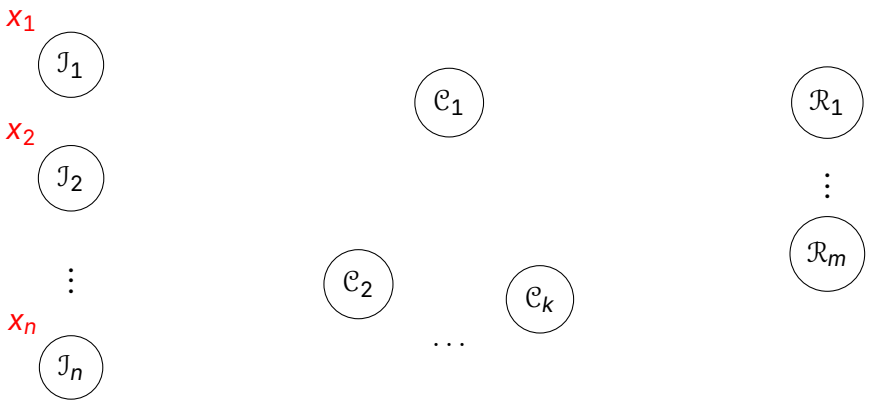
Desiderata for quantum-secure replacement

- Two-party protocol
- “SplitKey-like properties”
  - Not in today’s talk. Should not be too difficult
- Standardized
  - At least *interchangable* with a standardized scheme
- Certifiable
  - Can be meaningfully made to pass the tests created for single-party key generation / signing

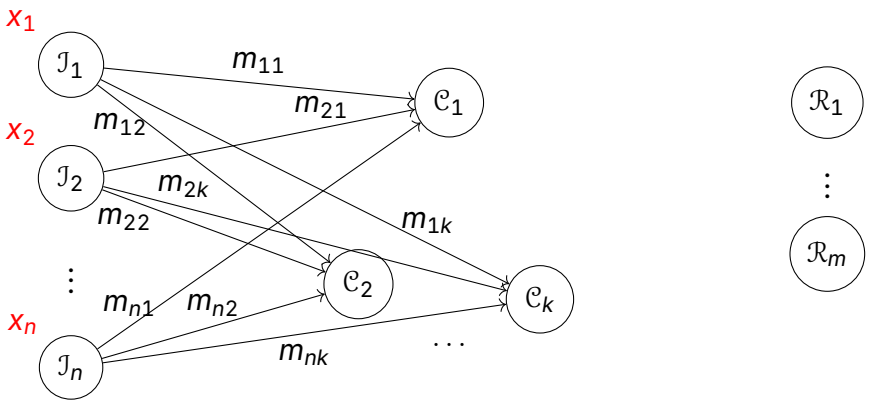
ML-DSA

- Lattice-based identification protocol + Fiat-Shamir-with-Aborts
  - (ECDSA, Schnorr: EC-based identification protocol + Fiat-Shamir)
- Some steps are difficult to thresholdize:
  - Rounding, rejection check
- Multiparty signing:
  - easy for RSA, Schnorr
  - more difficult for ECDSA
  - even worse for ML-DSA

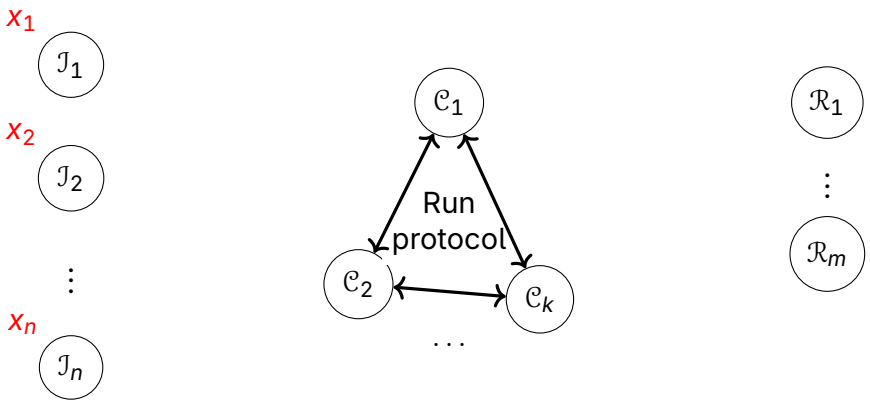
Secure multiparty computation (MPC)



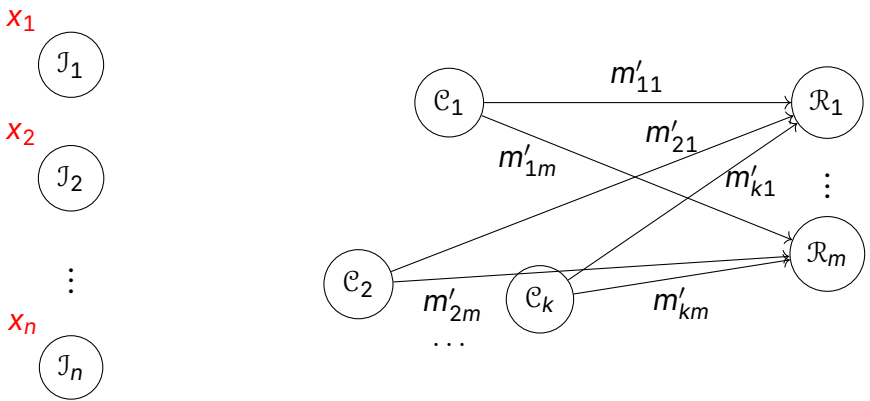
Secure multiparty computation (MPC)



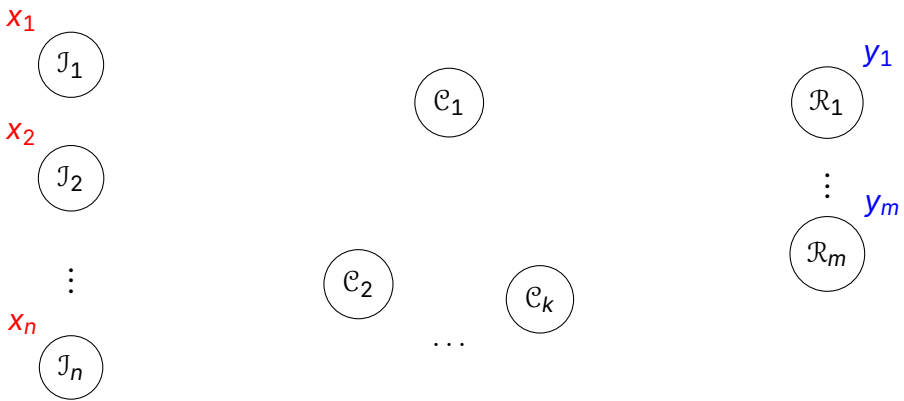
Secure multiparty computation (MPC)



Secure multiparty computation (MPC)



Secure multiparty computation (MPC)



Linear operations and linearization

(Linear) secret-sharing based MPC

- Private values are secret-shared among computing parties
- Operation on private values — proto.: shares of inputs $\rightarrow$ shares of outputs

- Addition protocol: each computing party adds its shares
- Protocols for non-linear operations: need additional *correlated randomness (CR)*
  - Together with CR, the operation is linear again

Linear and non-linear parts of ML-DSA signing

Signing

- $\mathbf{y} \leftarrow\$ S^{\ell}_{\gamma_1 - 1}$
- $\mathbf{w} \leftarrow \mathbf{A} \cdot \mathbf{y}$
- $c \leftarrow H(M, \mathbf{w}^H) \in B_\tau$
- $\mathbf{z} \leftarrow \mathbf{y} + c \cdot \mathbf{s}_1$
- $\mathbf{x}' \leftarrow \mathbf{w}^L - c \cdot \mathbf{s}_2$
- if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$, restart
- if $\|\mathbf{x}'\|_\infty \geq \gamma_2 - \beta$, restart
- return $(\mathbf{z}, c)$

Set-up

- Additive secret sharing modulo q
- $\mathbf{w}^H$ is declassified. c computed in clear

How?

- Green — linear computations
- Blue — we know how to do
  - Elements of $\mathbf{y}$ are between $-2^{\gamma_1 - 1}$ and $2^{\gamma_1 - 1} - 1$
    - Can generate random shared bits modulo q
- Red — complicated

(2+1) parties

- Correlated Randomness (CR) for non-linear computations has to come from somewhere

Parties themselves run a protocol for this

- 👍 Can be done ahead of time
- 👍 Does not involve an extra party
- 👎 Efficient generation known only for some operations
  - 👎 Even then, not cheap

Introduce an extra Correlated Randomness Provider (CRP)

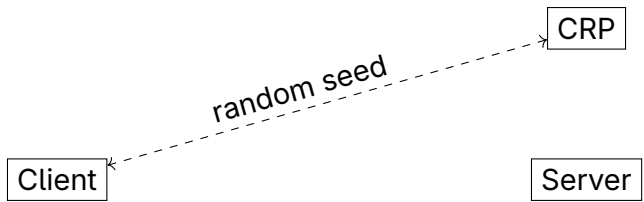
- 👎 An **extra party** to be managed
  - 👍 Communication patterns can be restricted
- 👍 More complicated operations can be supported

Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness

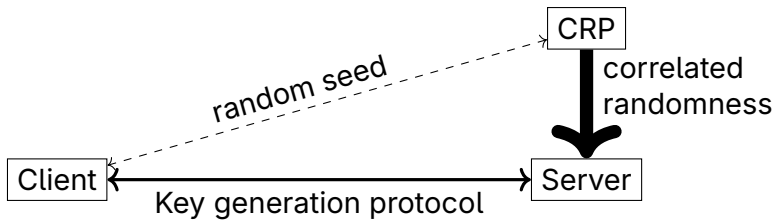
Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



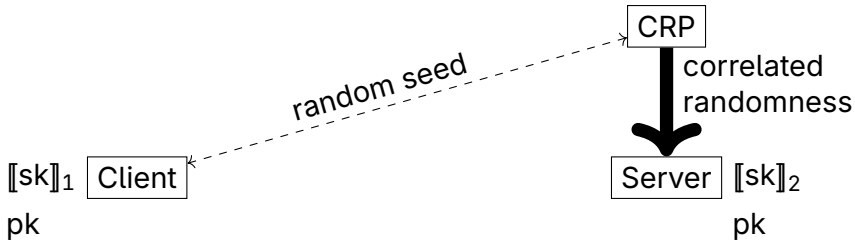
Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



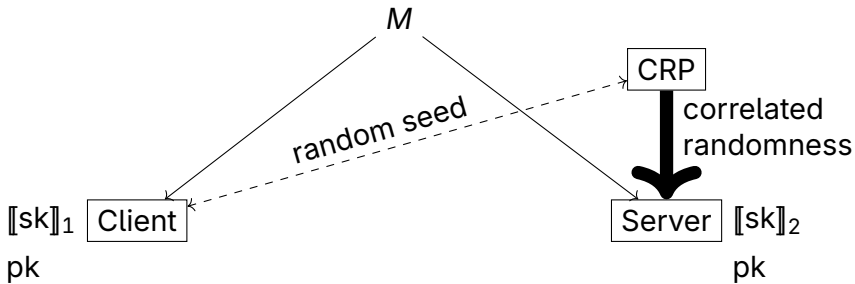
Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



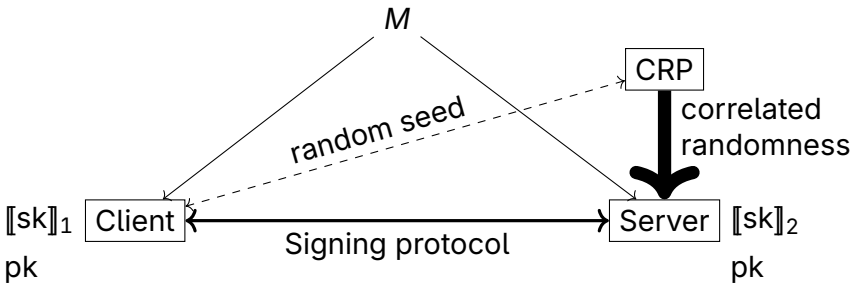
Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



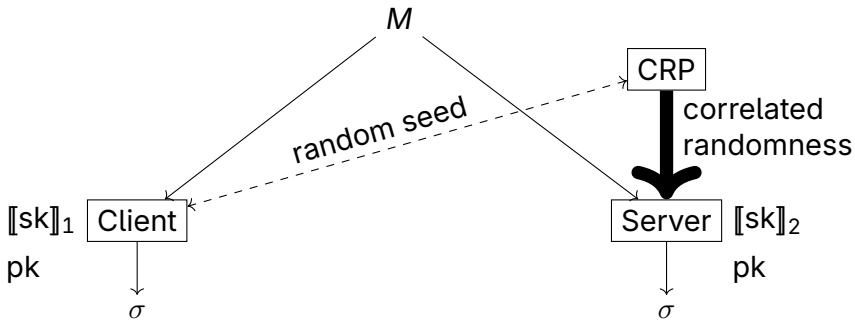
Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



Trilithium

- Based on techniques of Secure Multiparty Computation (MPC)
- Two parties. And a third party (CRP) for providing correlated randomness



Network traffic

Key generation: traffic volume

ML-DSA-	P↔S	CRP→S
-44	920.52 KiB	1.42 MiB
-65	1760.22 KiB	2.66 MiB
-87	3704.24 KiB	4.44 MiB

Signing attempt: traffic volume

ML-DSA-	P↔S	CRP→S
-44	186.72 KiB	55.05 MiB
-65	244.86 KiB	71.23 MiB
-87	328.50 KiB	95.95 MiB

Network traffic

Key generation: traffic volume

ML-DSA-	P↔S	CRP→S
-44	920.52 KiB	1.42 MiB
-65	1760.22 KiB	2.66 MiB
-87	3704.24 KiB	4.44 MiB

Signing attempt: traffic volume

ML-DSA-	P↔S	CRP→S
-44	186.72 KiB	55.05 MiB
-65	244.86 KiB	71.23 MiB
-87	328.50 KiB	95.95 MiB

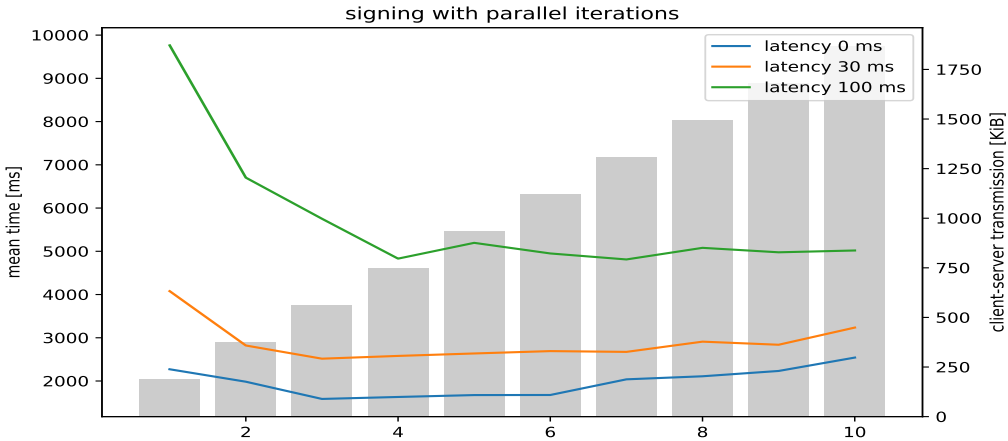
Key generation: num. of rounds

3

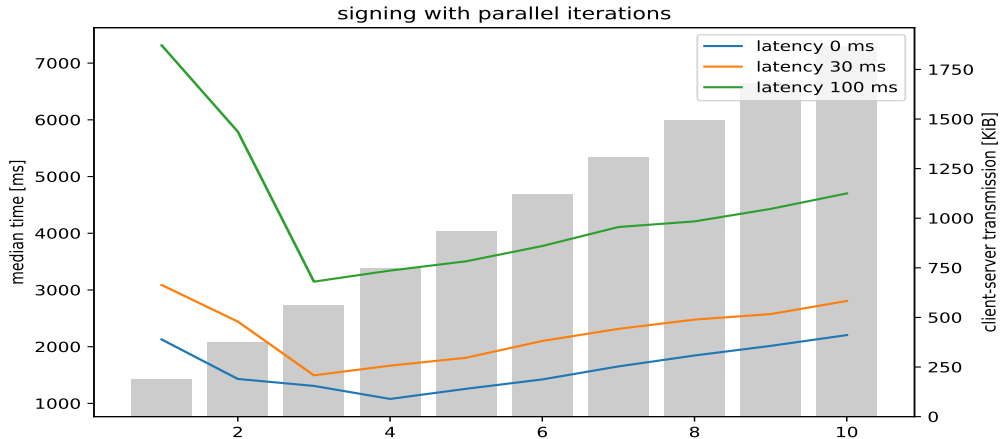
Signing attempt: num. of rounds

14

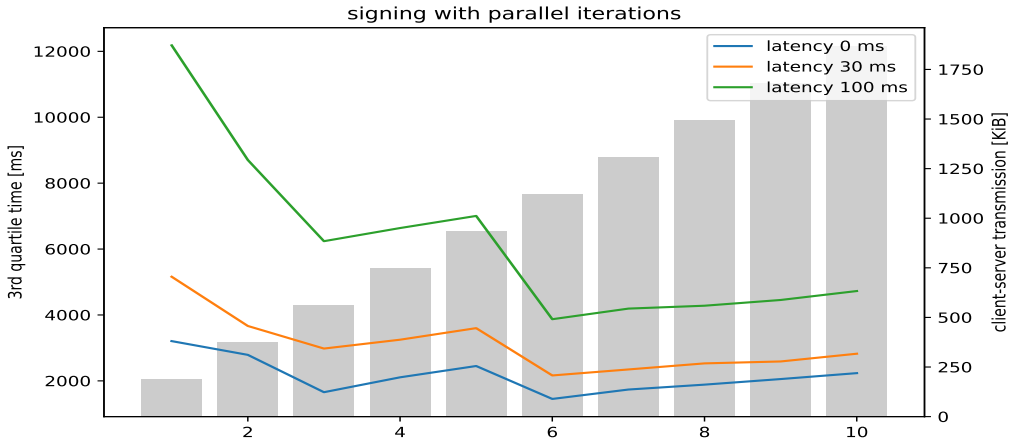
Time to create a signature (mean)



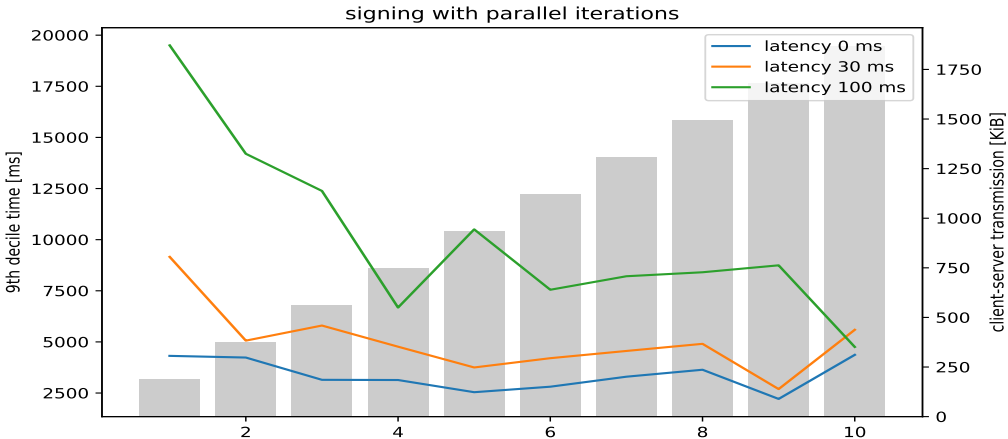
Time to create a signature (median)



Time to create a signature (3rd quartile)



Time to create a signature (9th decile)



Security model

As two-party protocol

- (output to CRP: nothing)
- Secure against one malicious party
- (additionally, a trusted CRP)

As three-party protocol

- (output to CRP: signature or rejection)
- Secure against one malicious party
- A *selective failure attack* is available for the CRP

Deployment discussion

- 👎 Separation between Server and CRP is a must.
- 👍 Time to sign is very much acceptable
- 👍 Client $\leftrightarrow$ Server traffic amount is very good
- 👎 CRP $\rightarrow$ Server traffic amount is concerning
  - 👍 *Pseudorandom Correlation Generators* can bring this down, at the cost of increasing Server $\rightarrow$ Client traffic
- 👍 The signatures cannot be distinguished from FIPS-204 signatures
  - 👎 But FIPS-204 requires **pseudo**random generation of the ephemeral secret

Thank you!

- Questions?

cybernetica
Cybernetica
cybernetica_ee
Cybernetica