Post-Quantum

Cryptography Conference

Hybrid PQC Digital Signatures and SSI

Self-sovereign identity (SSI) and digital credentials are becoming more and more practical, especially in Europe, and as such it is essential for them to be secured against the quantum threat. As we know PQC is a relatively modern area of cryptography and so we have decided to implement both a classical and a PQC digital signature scheme in a hybrid implementation within an SSI stack. Therefore, providing the classical security we know and trust while also preparing for the eventual quantum computing attack in the future. We are developing a hybrid PQC digital signature scheme in which we are implementing ML-DSA and ECDSA in parallel. The goals of our project also include the benchmarking and comparison of the hybrid vs PQC vs classical implementations.

Erik Hieta-aho

Senior Scientist at VTT, Research Center of Finland

SSL.com PQ SHIELD HID KEYFACTOR ENTRUST

January 15 and 16, 2025 - Austin, TX (US) | Online

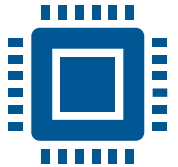
Hybrid PQC Digital Signatures and SSI

Erik Hieta-aho, PhD
Valtteri Lipiäinen

Hybrid PQC Self-Sovereign Identity Stack -HPQCSSI

Self-Sovereign Identity

Cross-border



Hybrid post quantum cryptography

Self-Sovereign Identities Stack

- Self-Sovereign Identities (SSI) allows users to control their identity
- Users have **digital credentials** (e.g. driver's license credentials, ePassports, University credentials)
- An SSI ecosystem requires a software stack
  1. Wallet
  2. Issuer
  3. Verifier
- We are supporting the open-source solution walt.id

Post-Quantum Cryptography Digital Signatures Brief Update

NIST Digital signature schemes

- Another competition for digital signature schemes that are based on hardness problems that aren't lattice-based.
  - In 2023 there were **40 signature schemes** accepted in round 1.
  - In August NIST announced that they have **accepted 14** of the 40 signatures into round two of evaluation. This second phase of evaluation and review is estimated to last 12-18 months.
  - CROSS, FAEST, HAWK, LESS, MAYO, Mirath (merger of MIRA/MiRitH), MQOM, PERK, QR-UOV, RYDE, SDitH, SNOVA, SQIsign, and UOV.
- https://csrc.nist.gov/pubs/ir/8528/final

Security of SSI

Verifiable Credentials and their security in the future

- The security within an SSI stack is focused around digital signatures.
- Currently verifiable credentials implement classical DS algorithms.
- SSI is still a developing technology without distinct standards, but with the European regulation eIDAS v.2, it is becoming more widely implemented on a global scale.
  - **eIDAS v.2 does not include the requirement of PQC.**

Hybrid Cryptograhy and PQC

Hybrid Cryptography

- Hybrid cryptography has a few different perspectives and definitions.
- In the context of our project we are considering hybrid as an implementation of two digital signature schemes in parallel.
- We thought it would be practical to implement both classical and PQC algorithms to ensure the trust of the general populace with classical ECDSA while also securing credentials for the future with ML-DSA.

Hybrid signature scheme

$$\begin{array}{c} \text{ML-DSA}(m \,||\, \sigma, k_1) \\ \uparrow \\ \text{ECDSA}(m, k_2) \end{array}$$

- Non-separability: attacker cannot claim to have a plain ML-DSA signature.
- Simplicity: lessen risk of fresh attacks due to hybrid approach.
- ML-DSA: Is a good choice for verifiable credentials due to fast verification.
- ECDSA: Used in current systems.

Implementation

- The solution being migrated
  - written in Kotlin
  - uses the Java Security API (with Bouncy Castle provider)

- Migration consists of three main tasks
  1. Implement a Java security provider for hybrid signatures
  2. Implement JOSE-specific functions
  3. Integrate both into SSI-specific logic

- Implementation done in a test environment
  - Standards are necessary for interoperability and wide adoption

Implementation challenges

- The main obstacle is the lack of support in web standards
  - SSI relies on JOSE for cryptography.
  - Standards for PQC are still at draft stage, and missing for hybrid cryptography.
  - JOSE libraries only support standard signature types.

- Crypto agility helps transition work. Using the Java Security API is useful, but attention should also be paid to application-specific code.

- PQC/hybrid signature are much larger, requiring changes in database configuration.

- Hybrid signatures are inherently harder to implement than plain PQC
  - We have both implemented.

Performance issues

- Verification speed: ML-DSA verfication is fast, so no challenges
- Key sizes and DIDs
  - ML-DSA public keys are large
  - Using did:key / did:jwk makes for large DID identifiers
- Signature sizes (in bytes)
  - ECDSA: 66
  - ML-DSA: 3309
  - Hybrid: 3393

Further plans

- Benchmark different PQC algorithms and the hybrid algorithm
- Possible development of Machine 2 Machine authentication flow
- Develop into an android application (currently a webapp)
- ZKP SSI and PQC

VTT, Finland and Ohio University, USA

Collaboration with Ohio University developing a transatlantic SSI stack

- SSI in the US and EU

- https://vcplayground.org/
  - Allows for testing implementations of a large variety of verifiable credentials
- OU partners have their own implementation of the SSI stack

“Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the granting authority can be held responsible for them. Funded within the framework of the NGI Sargasso project under grant agreement No 101092887.”

NGI SARGASSO

Co-funded by the European Union

Zero Knowledge Proofs and SSI

TANGO – Horizon Europe Project

- SSI and Zero Knowledge Proof implementation
- The implementation and testing of zero-knowledge proofs are also being developed in the SSI domain.
- Formatting and structure of verifiable credentials allows for the ability to restrict access to the information that is within a credential while also verifying necessary statements as true.
  - E.g. Verify someone is 18 years old, without revealing their birthday or age.

beyond

the obvious

Erik Hieta-aho
Erik.Hieta-aho@vtt.fi
Valtteri Lipiäinen
Valtteri.Lipiäinen@vtt.fi