Post-Quantum

Cryptography Conference

Stateful Hash-Based Signature Schemes

Volker Krummel

Chapter Lead PQC at Utimaco

Stateful Hash-Based Signature Schemes

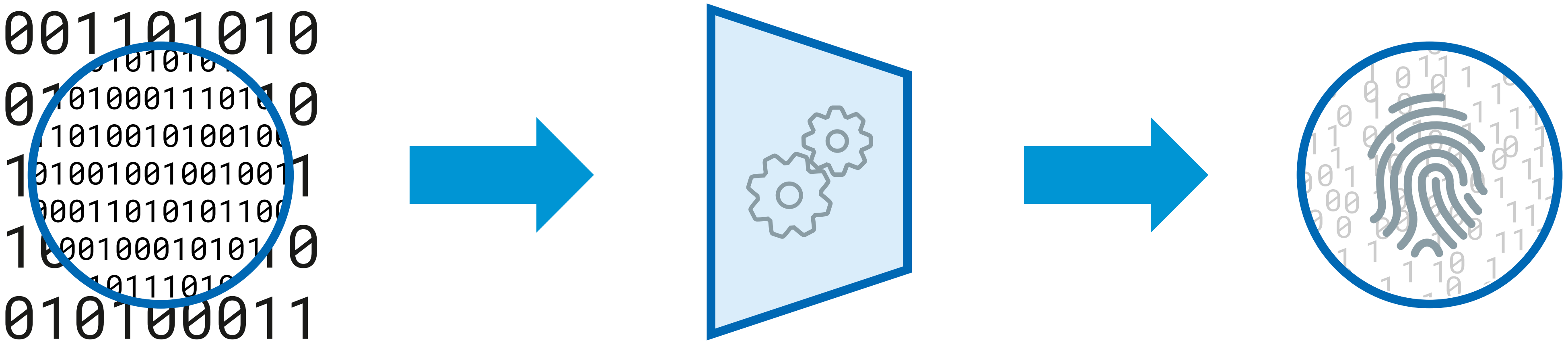
Dr. Volker Krummel, Chapter Lead PQC

07.11.2023

Agenda

1. **Stateful Hash Based Signatures (s-HBS)**
2. **Limitations of s-HBS**
3. **Proper State Handling Approach**

Cryptographic Hash Function H



Hash function means…

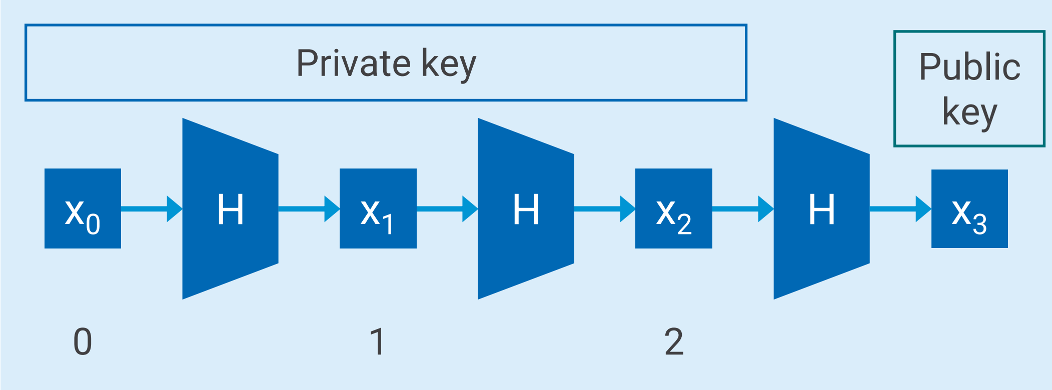
- $H : \{0,1\}^* \rightarrow \{0,1\}^{256}$
- a method of compressing strings
- input is called “message”, output is called “digest”

Cryptographic means… (in this context)

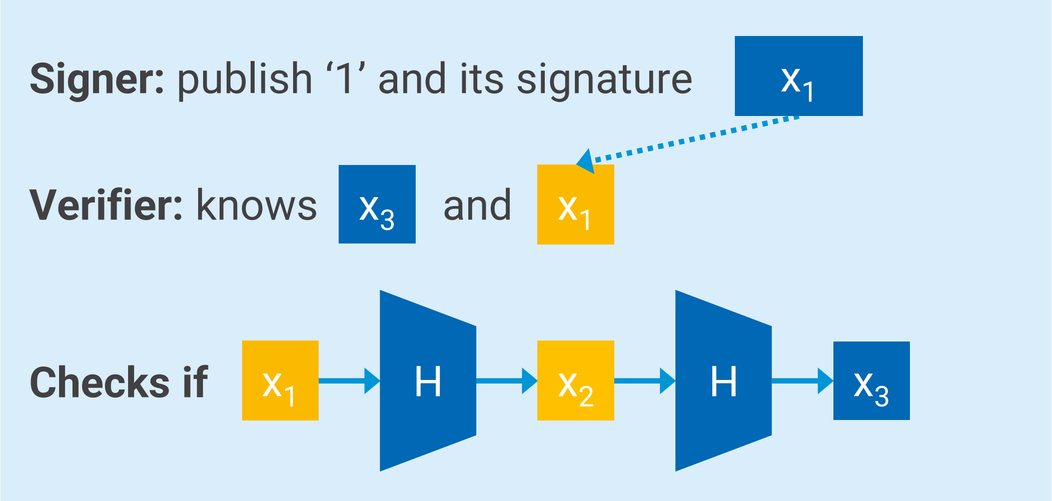
- **One-way:** Given D, hard to find M such that H(M)=D
- **Collision resistance:** Hard to find M <> M’ for which H(M)=H(M’)
- **Unpredictability:** M→H(R,M) unpredictable when R is secret
- **Extraction:** if M has high entropy then H(M) is ~ uniform

Stateful Hash based Signatures

One Time Signature – Basic Idea

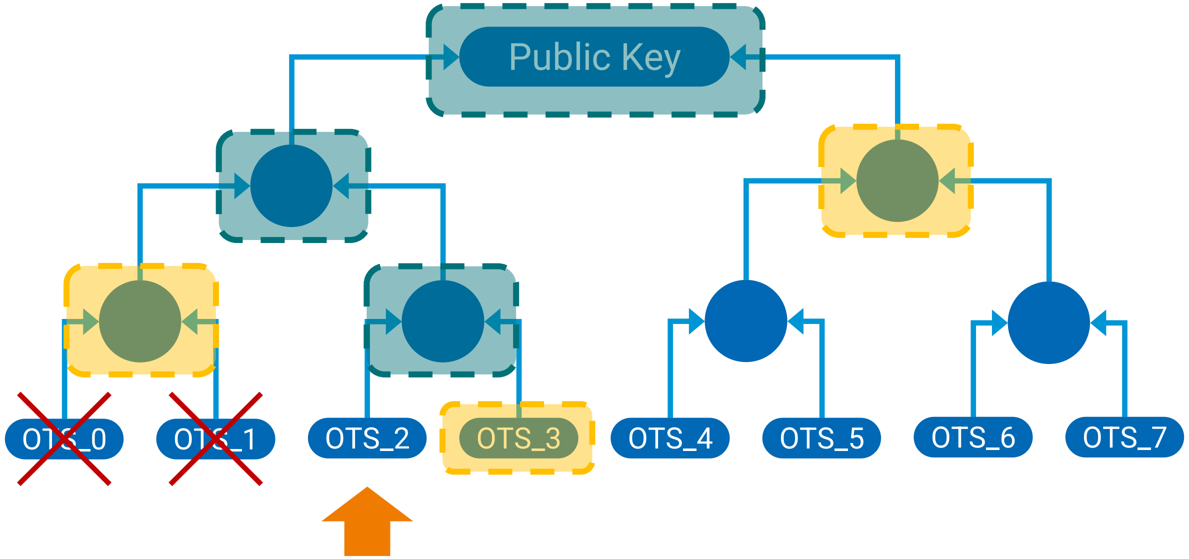


Example: signing and verifying message “1”



Combining many OTS Schemes

- OTS would require too many public keys
- Idea: build up a tree structure → single public key



Challenges remain

- Keep track about which OTS private key was already used→ State handling
- Limited number of signatures

Pros and Cons of s-HBS

Pros of s-HBS

1. best ratio of pub key size and signature
2. well understood security guarantees
3. simple & mature design
4. already standardized
5. recommended as 1:1 substitution
   -> may skip hybrid-approach

Cons of s-HBS

1. Limited number of OTS-keys
   -> limited number of signing operations
2. Stateful

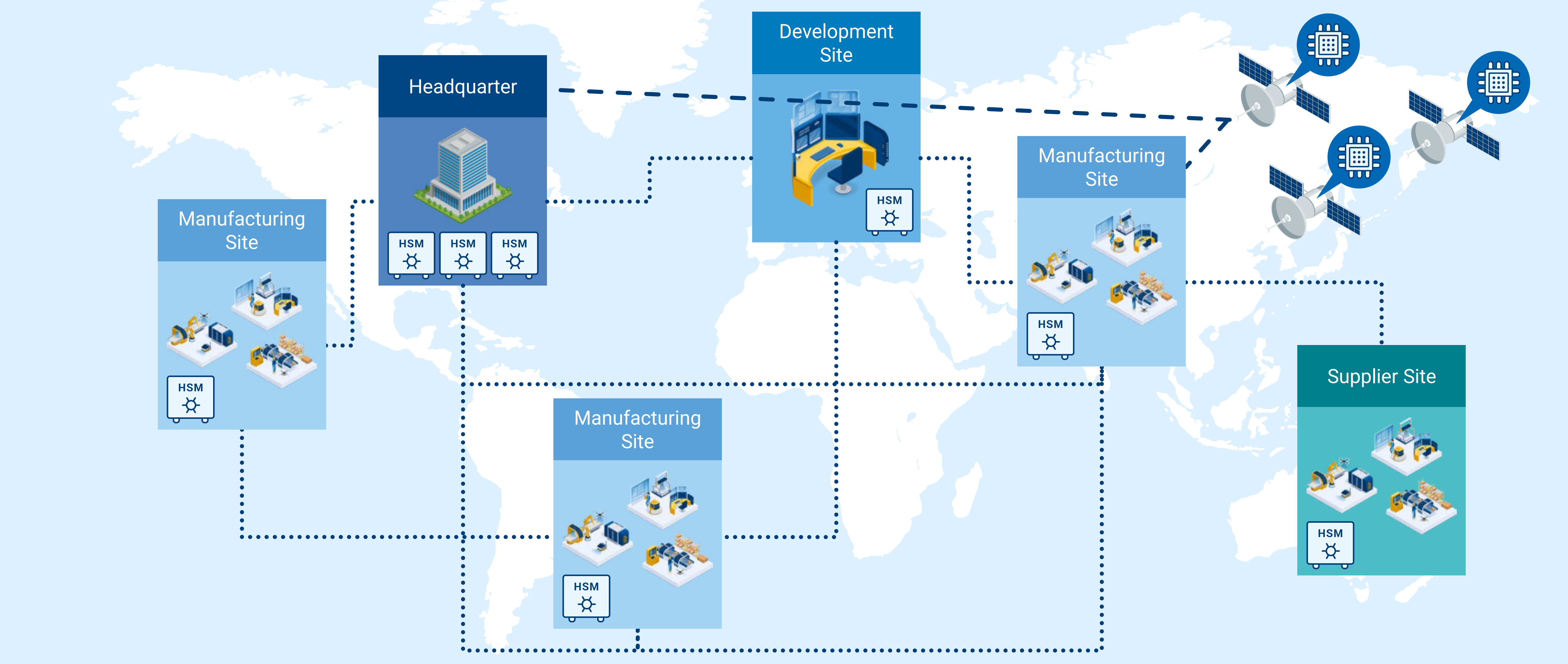
Holy Grail of PQC-Signatures!?

The Great Seal!?

"You chose wisely. But the Grail cannot pass beyond the great seal. That is the boundary, and the price of immortality."

Case Study Chip Manufacturer

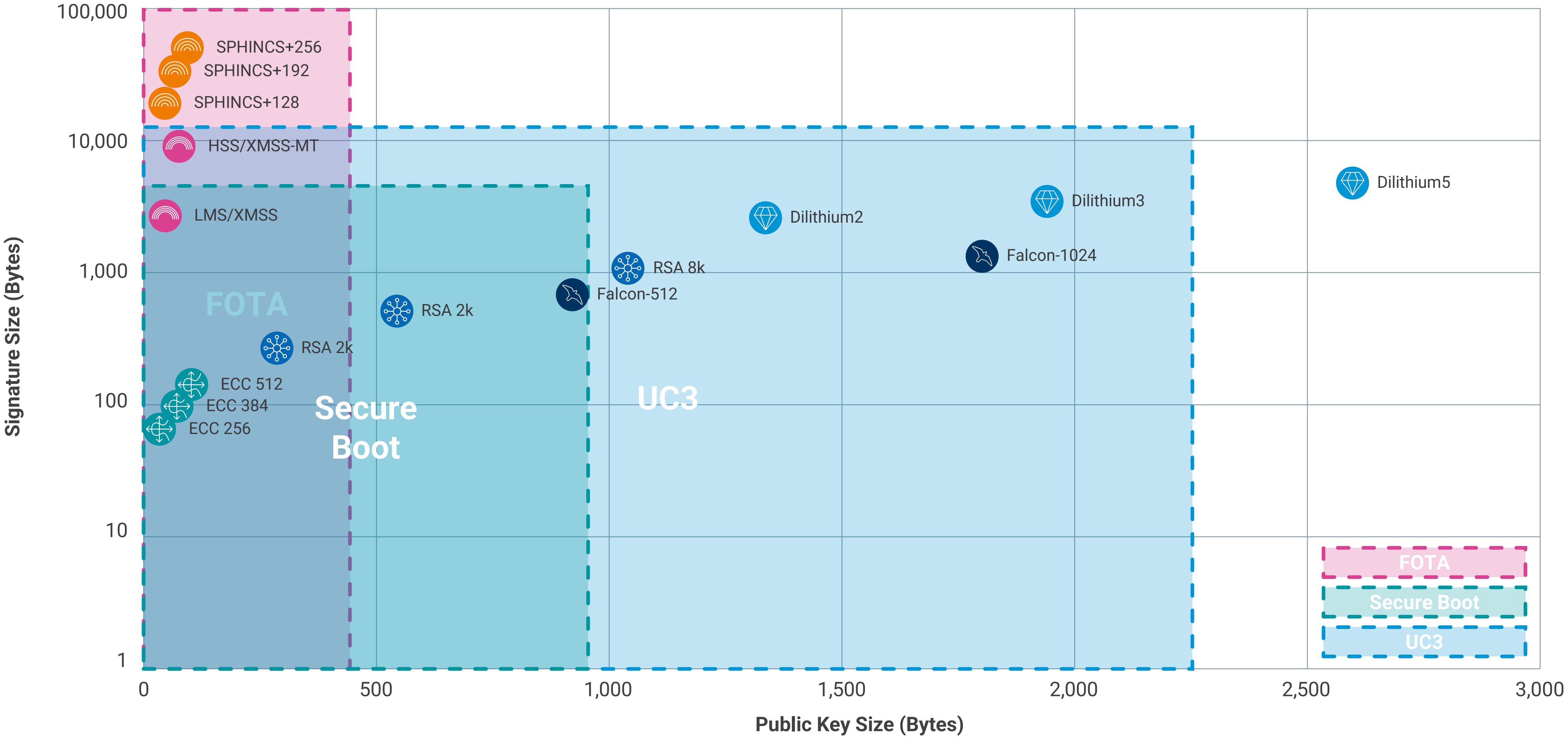
Challenges of Distributed Sites



Selection of PQC Algorithms by parameter (example)

utimaco®

PQC signature algorithms compared to ECC / RSA



Thoughts on Limitation #1 – limited number of keys

utimaco®

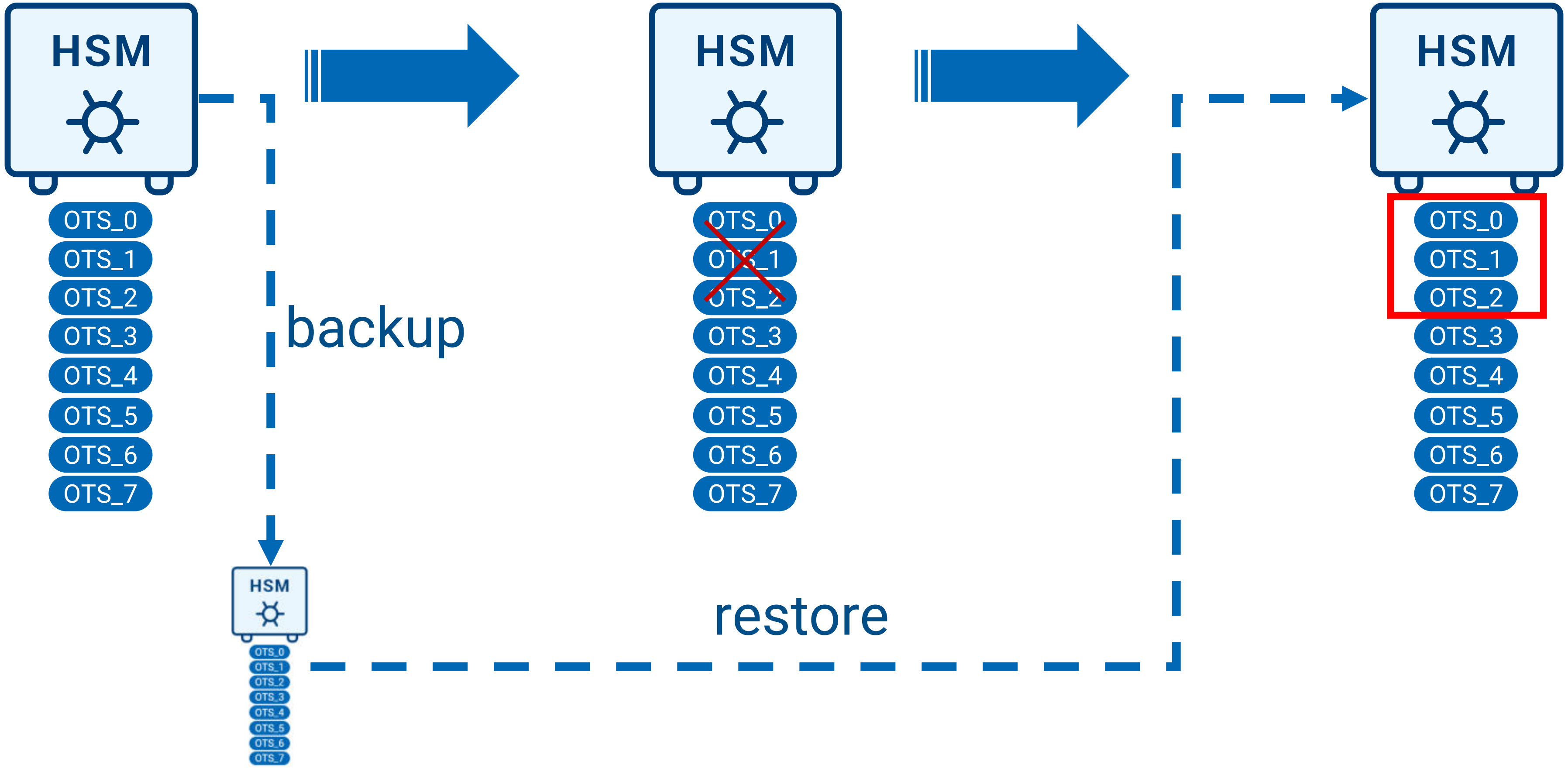
- Restricts application to use cases with reliable estimation of number of signatures
  -> adds a further risk of running out of keys
- be as close as possible to real number of signatures
  -> keeps size of signatures low
- Works well for long term „static“ security use cases
  - Root-CA
  - Firmware Signing
- **Option1:** enable multi-tree variant
- **Option2:** establish procedure for key substitution (good practice!)

Thoughts on Limitation #2 – state handling

utimaco

Backup & Restore

- Classical Backup & Restore procedures restore an old state -> violate the security requirement!



Thoughts on Limitation #2 – state handling

- simple Backup & Restore procedures restore an old state
  -> violate the security requirement!

- **Option 1:** adapt backup & restore procedure to support disaster recovery
  1. „know what you signed“
  2. Add-on: if double usage is detected -> **revoke the key**

- **Option 2:** establish a proper state handling mechanism

State Handling

OTS Keys must be used maximal once!

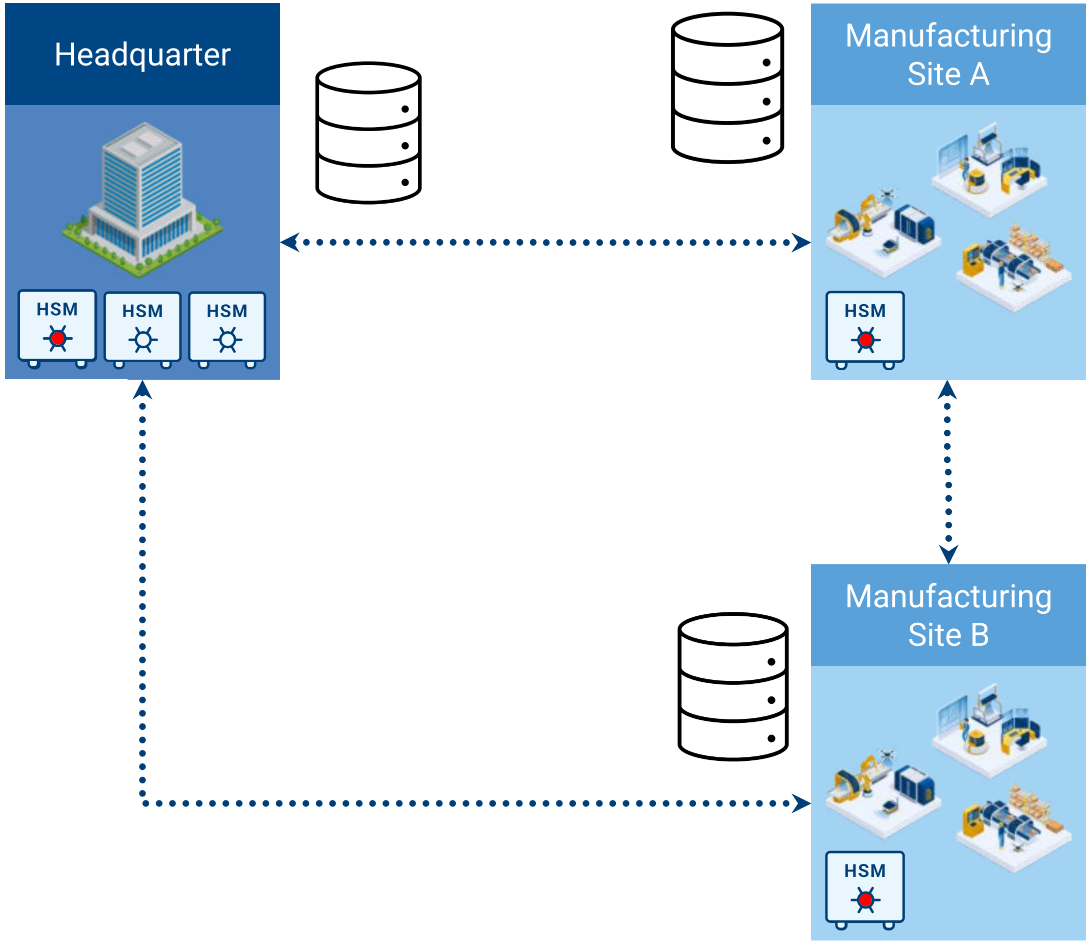
Simple bookkeeping becomes complex
- Usage of multiple HSM instances
- Usage of Backup & Restore

State handling must be
- Secure (must have)
- Flexible (disaster recovery / performance)

Design Properties of a Secure State Handling Architecture

1. **Authentic and confidential end-to-end export and import of key / state information**
   1. Do not use asymmetric PQC algorithms – not an adequate level of maturity
   2. Use symmetric cryptography (maximum entropy)
2. **Establish a reliable trust relationship between the HSM instances** (in secure environment)
   1. Allows a highly flexible transfer even during operating in the field
3. **Prevent replays** – protect the freshness
4. **Prepare for offline data** – allow for external key storage
5. **Separate keys and state information** (least to know principle)
6. **Asynchronous** → no need for direct (real time) communication between HSMs

Proper State Handling Approach – Security is Paramount



1. Setup phase (set up trust relationship)
2. Generate key in HQ
3. Distribute subsets to destinations
4. Operate …
    1. If risk of key exhaustion at one site - Securely transfer further keys from any other site(s)
    2. If site will be shut down - Securely transfer remaining keys to other site(s)
    3. Attacks – e.g., if A replays key transfer -> blocked
    4. Risk of faulty app exhausting all keys - only import small portions of the key; keep rest offline
    5. If HSM is destroyed -> loss is limited to a well defined subset of the key

······· Logical connection (network, portable storage, …)

External key storage (optional)

State handling like this is not an option for your use case?

Then …

Thoughts on Limitation #2 – state handling

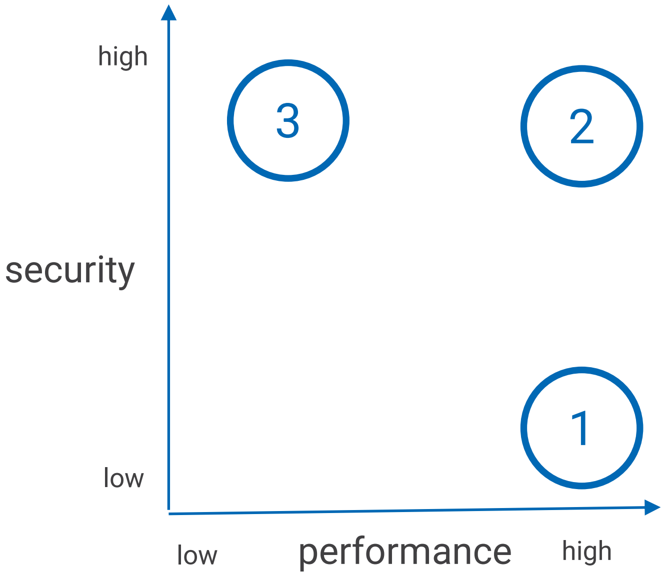
Source: https://medium.com/asecuritysite-when-bob-met-alice/a-lifetime-dedicated-to-citizens-rights-to-privacy-daniel-j-bernstein-ab5ab2bf0dc6

Thoughts state handling - Overview

- simple Backup & Restore procedures / key distribution mechanisms restore an old state -> violate the security requirement!

Options

1. adapt backup & restore procedure to support disaster recovery (know what you signed)
2. establish a proper state handling mechanism
3. go for a stateless signature algorithm



Summary and Q&A

utimaco®

PQC migrations need sophisticated planning

→ Long term security and availability

→ **Per proper(!)** use case definition

Use Cases for s-HBS exist

→ Analyze thoroughly!

SHBS provide best ratio of

- Public key size
- Signature size
- Performance
- very high level of maturity

Proper state handling in HSM

→ Limited number of signatures!

→ Adapted Backup & Restore as an independent means

Demo on request

Any more questions?

Volker.Krummel@utimaco.com

Thank you
for your attention!

Utimaco IS GmbH
Germanusstraße 4
52080 Aachen
Germany

Phone +49 241 1696-0
Web utimaco.com
E-Mail hsm@utimaco.com

utimaco®

Post-Quantum Cryptography Conference

PKI Consortium