

Post Quantum Digital Signature ?

Are we going calculate hash function inside of the HSM?

Tadahiko Ito (SECOM CO., LTD.)

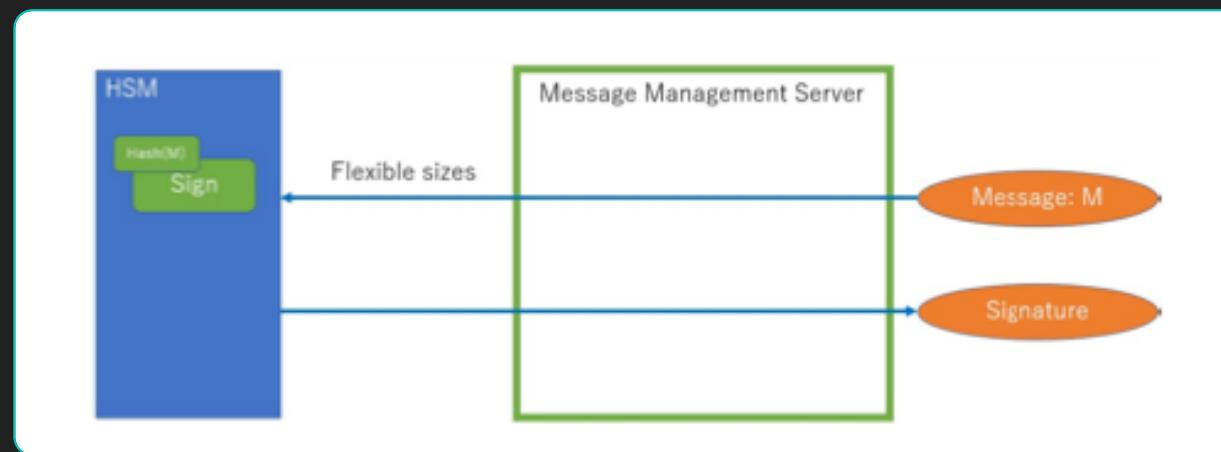
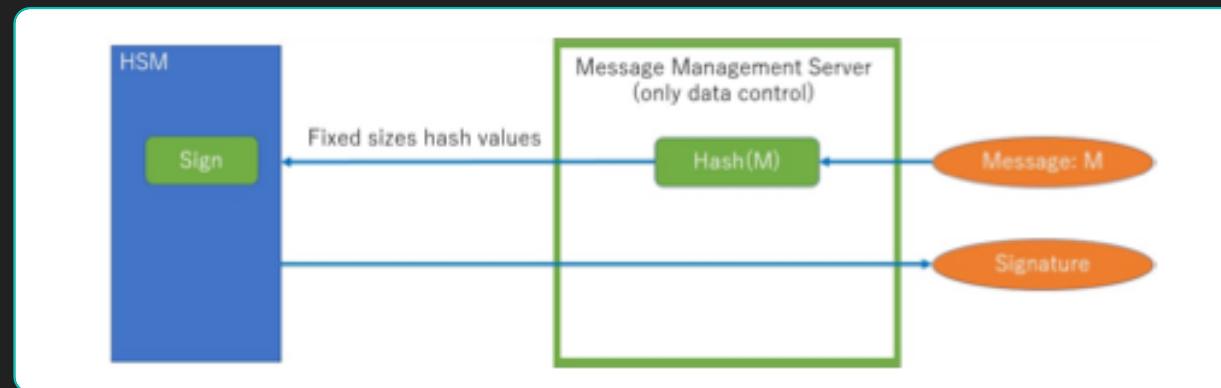
<https://eprint.iacr.org/2020/990>

- I publish a paper about separation of hash and asymmetric operations
- On RSA or ECDSA, you can separate hash operations and asymmetric operations
 - Typical case
 - Data controllers calculate hash
 - Send that hash into HSM
 - HSM (or key manager) calculate asymmetric operation, and return
 - It was effective, because
 - We could minimize resource requirement of HSM
 - Separation of control over data and key
- On PQC, We may not able to separate hash and asymmetric operation... What happen?

Our Scope(1)

- NOT “Key Encryption”
 - Actually, It can be most difficult topic, but many people are caring that.
- NOT “Authentication”
 - Lattice base PQC is most “promising”.
- But “Content commitment”
 - extendable signature, like PAdES, CAdES, XAdES, JAdES, may just solve that.
However, not all mechanism support that.
 - Not much people are caring that.

How PQC signature may work



Our paper

- Implement lattice base PQC signature with HSM
 - We **only** evaluate “**lattice base**” PQCs (which supposed to be most promising PQC algorithms)
 - Used **Programable USM** (Thales **Protect Server2** and **LunaSA-7**)
- Many lattice base ones can not separate hash and asymmetric operation.
It seems **Entire hash function** need to be calculated inside HSM in most cases
 - We have customers who need to sign document of 10GB.

Would it works??

May be not (our result)

Table 3 The time cost for hashing operations executed inside of the HSM (LunaSA-7) for three lattice-based digital signature schemes with different structures of cryptographic boundary

Scheme	Boundary Type	Time (millisecond)				
		1k	10k	100k	1M	10M
FALCON	A	4.29	9.06	52.40	501.48	4911.52
	B			0		
DILITHIUM	A	3.13	8.57	65.94	630.73	6294.30
qTESLA	A (before Ver. 2.8)	3.99	8.78	53.53	507.02	4922.03
	B (from Ver. 2.8)	2.27	3.49	3.15	3.42	2.99

Able to calculate hash outside. However, uniform random “salt” need to be provided at the place of hash function.

May need to change architecture bit? Use EaaS? Not sure about that.

Result

- Can be solved easily with one more hash function...
 - but it may separate number of hash operation for Authentication and digital signature.
 - Should we unify number of hash call?
 - some people may be happy to separate Authentication and content commitment!
- One solution might be separation of cryptographic algorithms between authentication(use lattice base PQC) and signing (use hash base PQC).
 - some people may be happy to separate Authentication and content commitment!
- qTESLA seems fine for us, but they are not 3rd round candidates.

Next Step?

- I am trying to raise issue and find best practice.
- ONIST? IETF?
- Anyone who is interested, please contact me.

Our Scope(2)

- How about Code signing, or Document signing?
 - Long Protection Period.
 - May need to care about Raise of Quantum Computer
 - It is much closer to CAs' Scope
- Well, problem might be less if we can use extendable signature, like PAdES, CAdES, XAdES, JAdES, etc. but not all mechanism support that.
- People seems to be not caring as much as encryption

Breaking down Problems(1)

- How about TLS encryption
 - Authenticating (like server-auth)
 - My Prime Scope
 - Relatively much **shorter protection period**
 - We may able to wait implementation for a while, but API should be considered in advance
 - **Key exchange or key encryption** (ie: edh for forward security)
 - It seems current **Primary target** for many people
 - It is problem, because exchanged key need to have same protection period as protected data, which can be very long.
 - Protection period can be very long
 - It might be scope for browser? (**may not be Cas scope**)
- Symmetric encryption
 - May be not our scope

Background

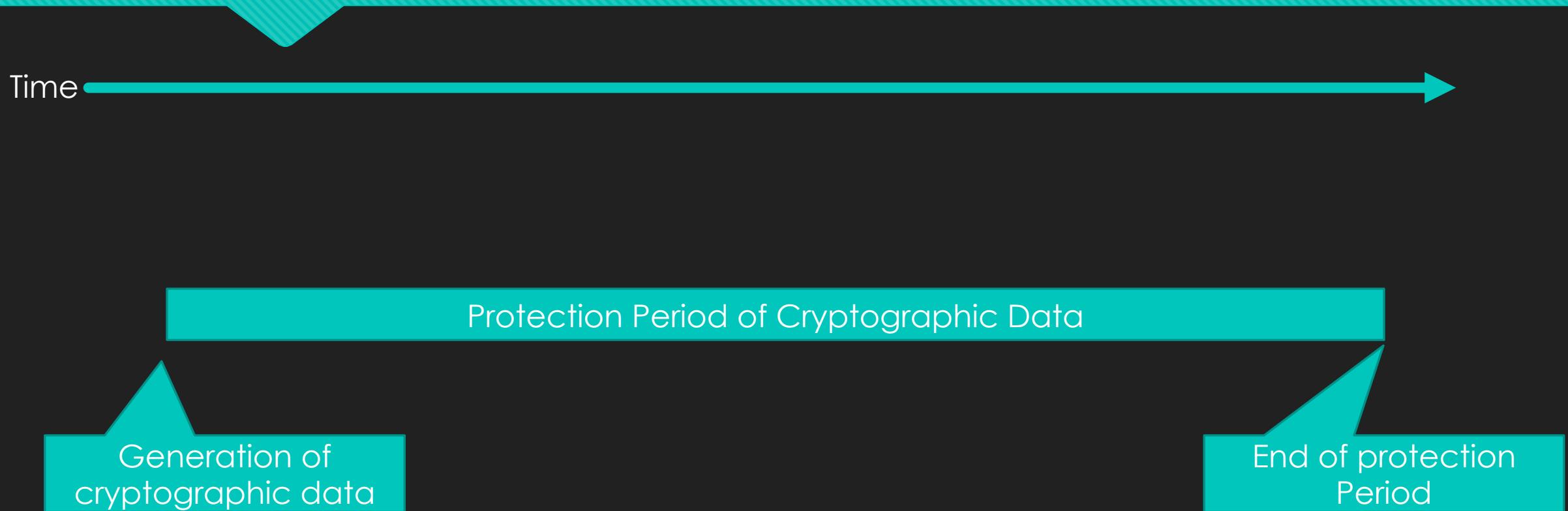
- A Quantum Computer which is able to Solve Crypto Algorithm?
 - When it will come to real world?
 - It seems, it would not be possible within 10-15 years [1]
 - We really do not know, if it will come, or even if it is impossible for human.
- Quantum Safe
 - Symmetric encryption
 - AES 256 I fine enough
 - Hash function
 - Longer output size of SHA2 or SHA3 may be needed
 - Public Key Crypto
 - No Longer Secure

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	-----	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

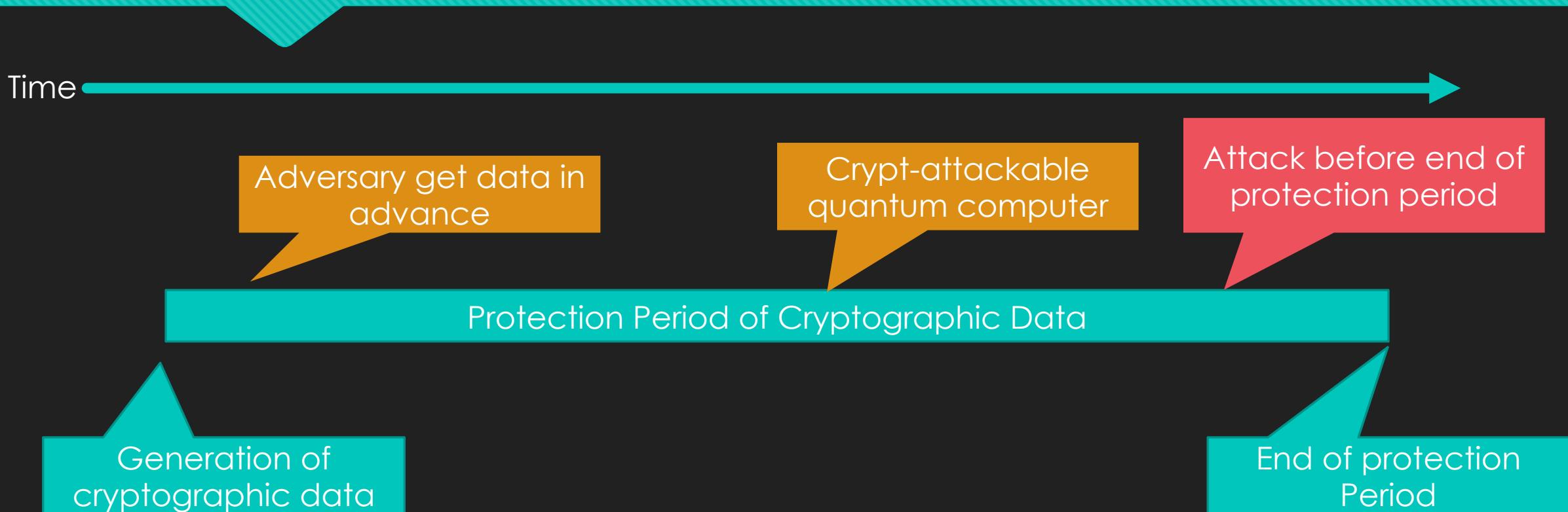
[1] Virtual Workshop on Considerations in Migrating to Post-Quantum Cryptographic Algorithms

[2]<https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>

Major Risk For Public Key Crypto



Story to attack Public Key Crypto



**The risk is very depending on the lifetime of data.
(longer is risky)**

14