Considerations on separation of hash

Tadahiko Ito (SECOM CO., LTD.)

Overview of my presentation : How to support diverse sizes of data

- Current PQCs seems OK with authentication and small data.
- Current PQCs seems OK with signing for large data, if it were with pre-hash.

- However, it seems We need
  - to differentiate two usecases ? OR
    - Use different PQC protocol (including minor change)　AND
      - e.g. hash(0|tr|m)for without pre-hash, hash(1|tr|m) for with pre-hash.
    - introduce salted external pre-hash for large file
  - to use same PQC protocol ?
    - Compiler approach? A compiler, which take (NIST)pqc signature algorithm as input, and output general PQC protocol
- Issues
  - Can we divide “small data” and “large data”?
    - authentication and signing? can we draw hard line between them?
    - With some size(i.e. 100k)? It can effect interoperability a lot.
  - Who and where should design compiler?
    - Anyone interested?
- Appendix: PDF signing data structure

Usecases for signing

If lifetime of data is longer, We should prepare more in advanced

Purpose	Usecase	Lifetime of Data	Data Sizes
Authentication	Authentication	days	several Kilobytes
Sign	Legally binding agreement		hundreds Kilobytes
	Medical data	can be decades	can be several Gigabytes
	CAD data		
Time stamp	Time stamp	around one decade	several Kilobytes

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

Signed data sometime have much larger data

We use HSMs for Signing purpose,
- raw data for Multi-slice CT can be 2GB for a single inspection (2MB/slice, 1000 slices).
- PDF file for CAD data can be several GBs .

It is very resource consuming to calculate entire hash outside of HSM (I would say it is not realistic)

Not realistic to calculate several Gigabytes inside HSM.

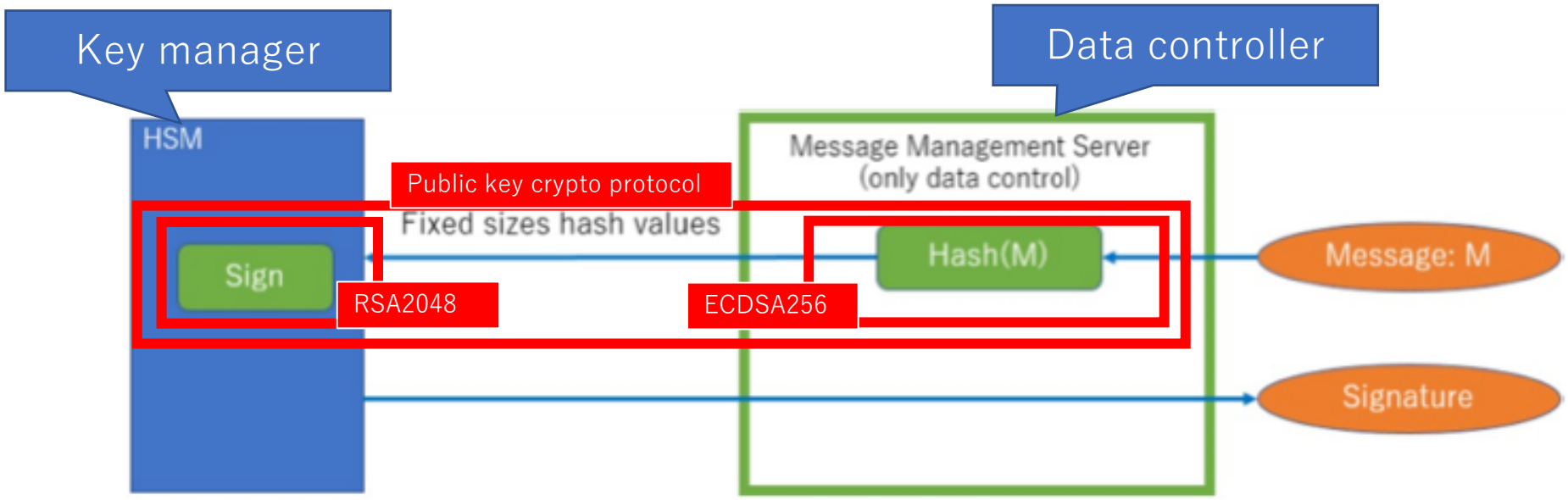
(it would be faster with HW accelerator, but still, it is not for HSMs.)

Scheme	Cryptography Boundary Type	Time (millisecond)				
		1k	10k	100k	1M	10M
FALCON	A	33.36	38.08	142.26	1240.59	11667.19
	B	0				
DILITHIUM	A	34.67	45.79	156.19	1351.4	12727.83
qTESLA	A (before Ver. 2.8)	34.78	44.78	138.05	1196.26	11810.52
	B (from Ver. 2.8)	30.84	38.25	38.63	38.63	31.42

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

Current Practice (with RSA or ECDSA):

divide Public key crypto into asymmetric operation(in HSM) and hashing(in data control server)

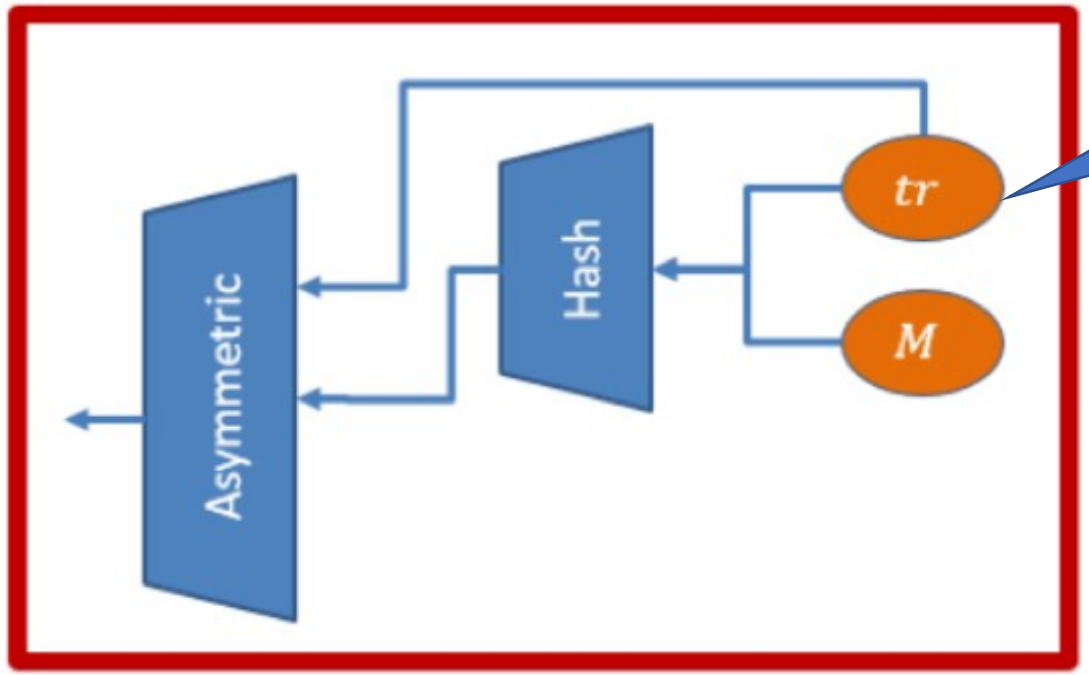


https://eprint.iacr.org/2020/990

Does it work with PQCs?

It seems we may not divide in some situation.

CRYSTALS- DILITHIUM



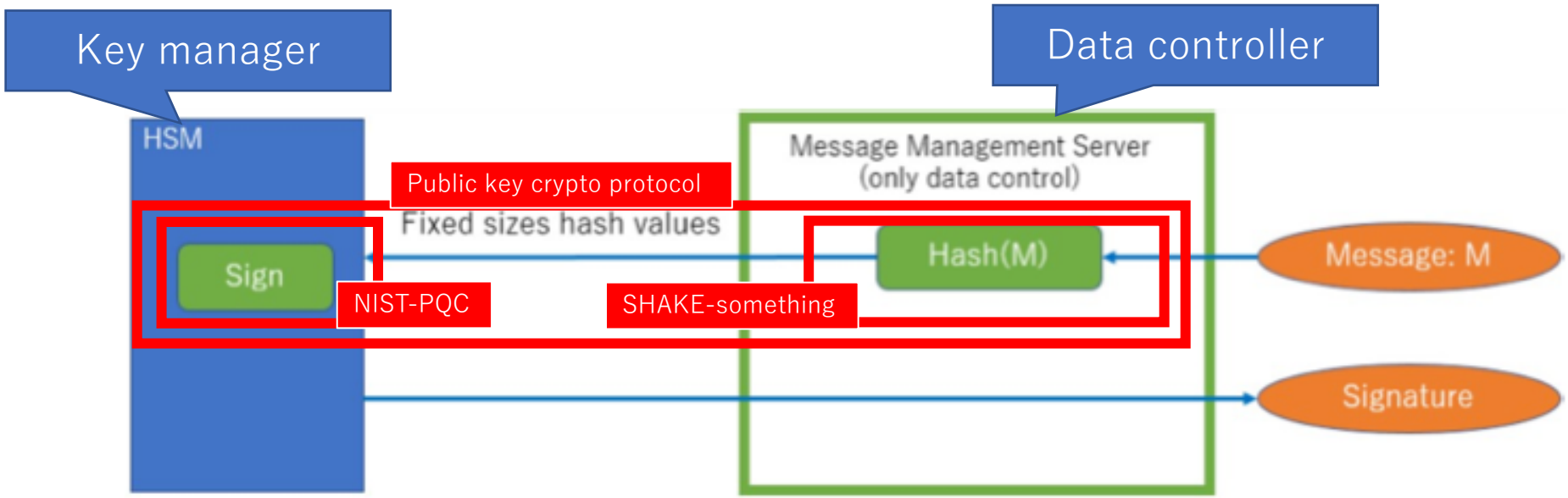
Should be secret
I am not sure how strict it should be

Tr and asymmetric operation should be inside of HSM.
Then, hash should be inside of HSM also.

data control only able to pypass?

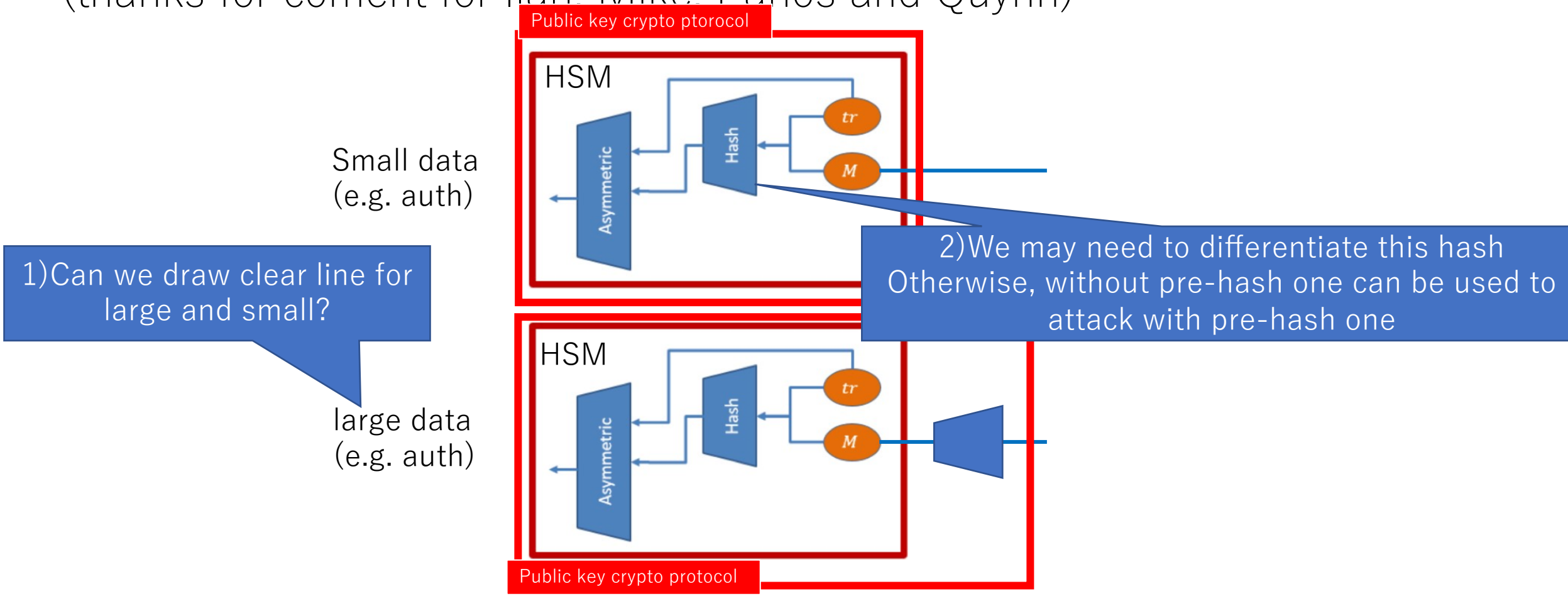
https://eprint.iacr.org/2020/990

We may be able to use pre-hash approach

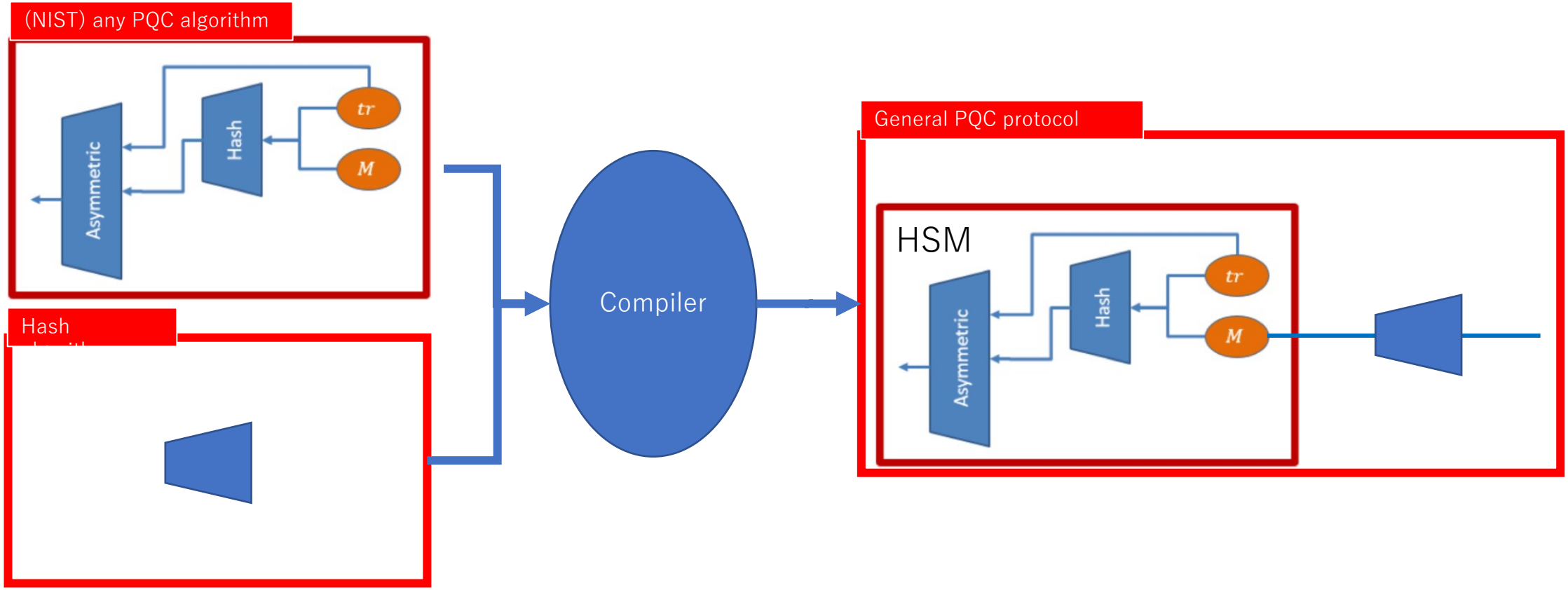


Issue with pre-hash?

(thanks for coment for Ilari, Mike, Panos and Quynh)



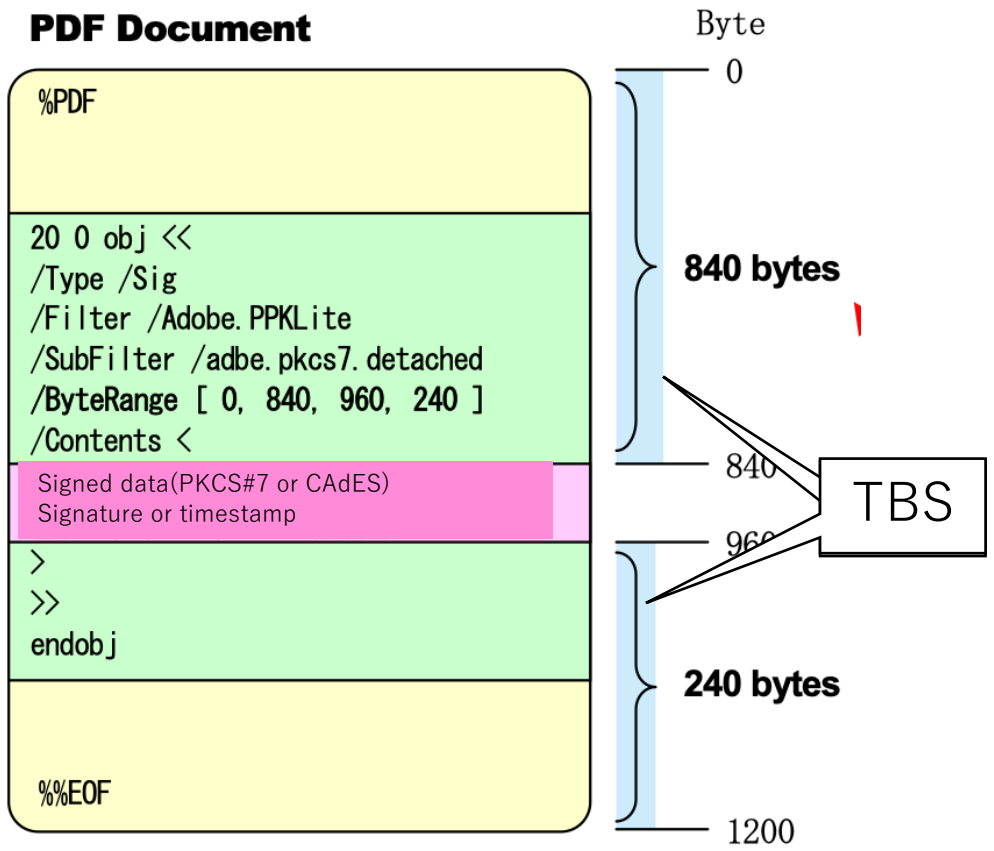
Another choice :
make compiler for PQC signature protocol?

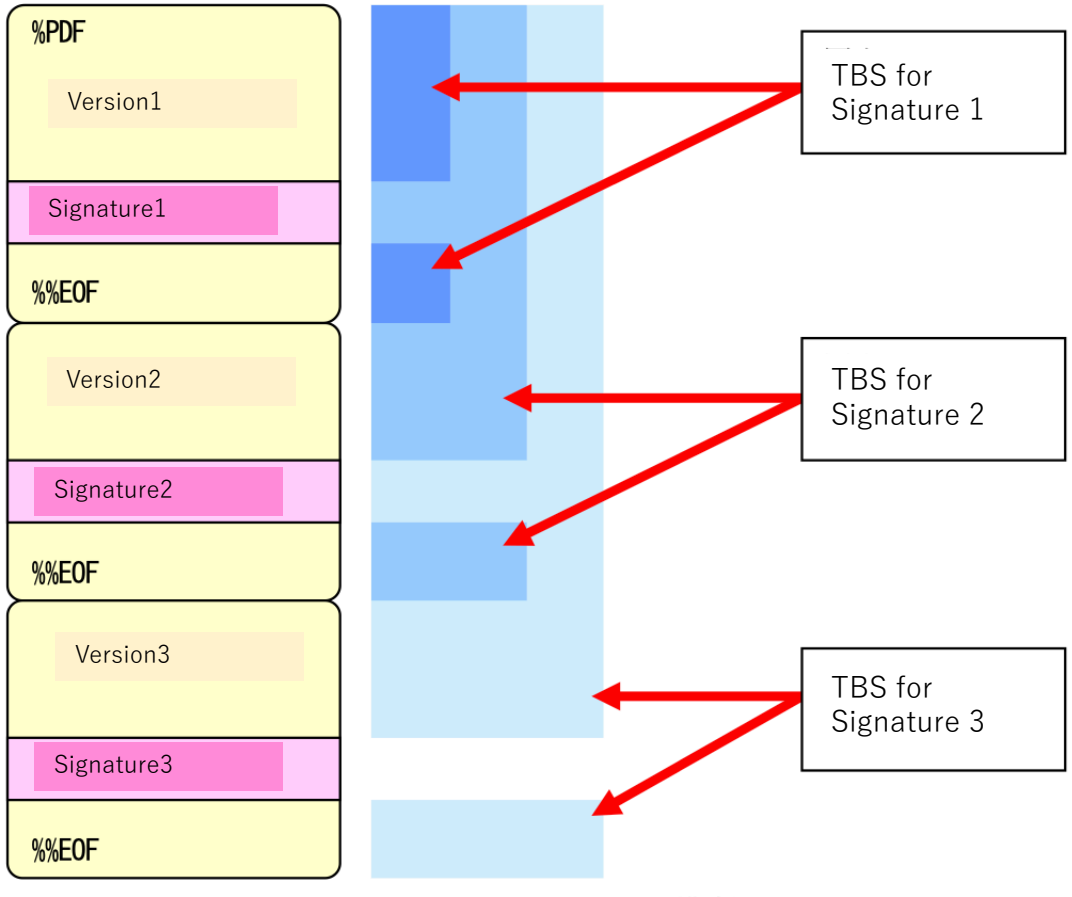


Compiler(PQCalgo, some pre-hash, (+ with some data format on input?)) => general PQC Signature

Pro: designing compiler can be done parallel to NIST PQC Project
Con: I am not sure where to discuss that.

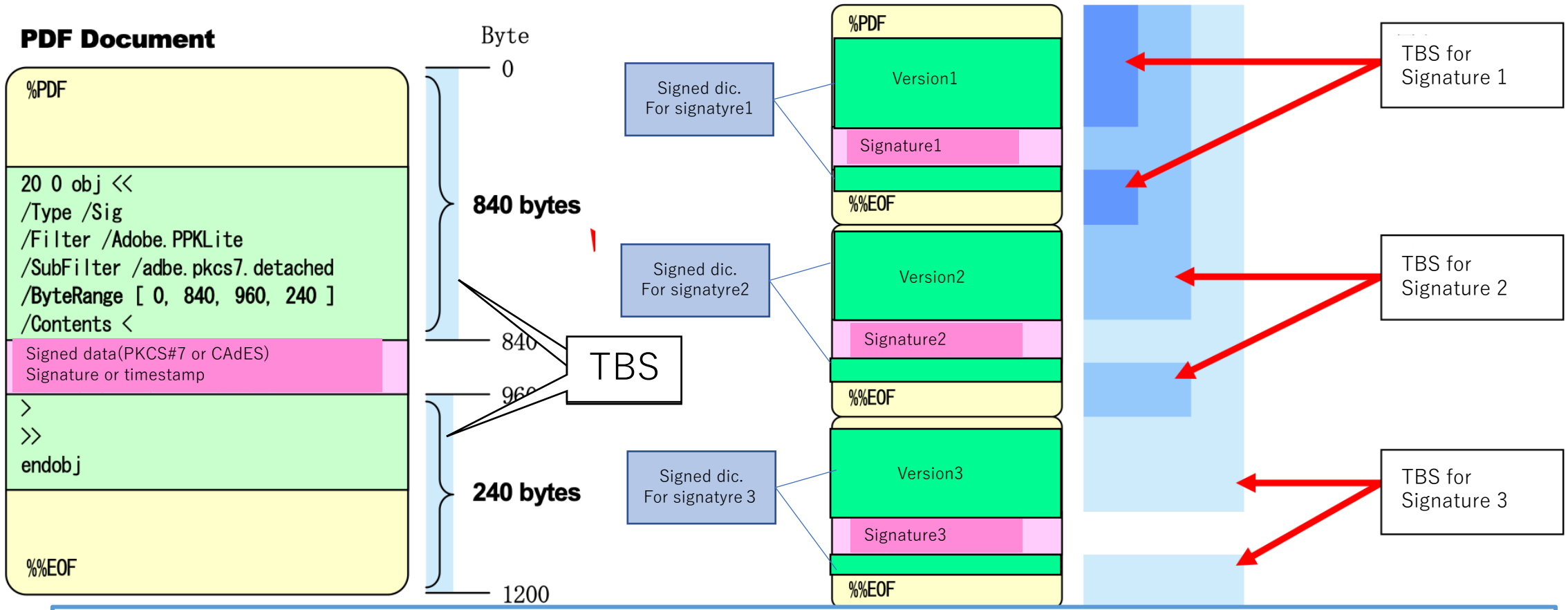
Appendix: An example of PDF serial signature





https://www.jipdec.or.jp/archives/publications/J0004290.pdf?fbclid=IwAR2VShfmSTDVEvOxKbKBIayQHUGHy8kQx0wnsmemXifassmZeLf7JmhpkVw

Appendix: An example of PDF serial signature

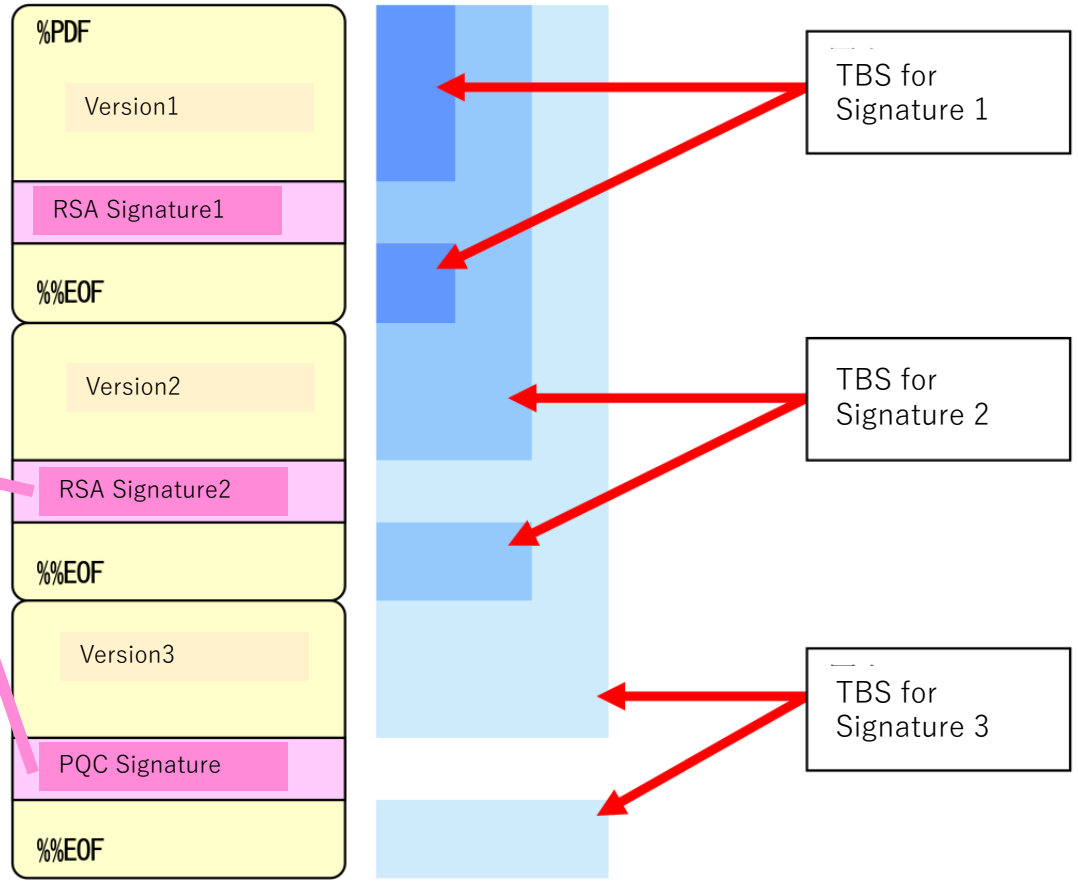


Note: Single Signature Dictionary(green area) with single CMS Signed Data with Single singer Info

https://www.jipdec.or.jp/archives/publications/J0004290.pdf?fbclid=IwAR2VShfmSTDVEvOxKbKBIayQHUGHy8kQx0wnsmemXifassmZeLf7JmhpkVw

Appendix: An example of PDF serial signature

Separate certificates and separate signatures seem to work fine with current architecture.



https://www.jipdec.or.jp/archives/publications/J0004290.pdf?fbclid=IwAR2VShfmSTDVEvOxKbKBIayQHUGHy8kQx0wnsmemXifassmZeLf7JmhpkVw