Discussion of PKCS#11 v3.2 proposal

Falko Strenzke
MTG AG, Darmstadt, Germany

PKCS#11 v3.2

- v3.2 draft for public review until July 29
- PQC schemes
  - ML-KEM
  - ML-DSA, Hash-ML-DSA
  - SLH-DSA, Hash-SLH-DSA
  - XMSS
  - LMS/HSS (since v3.1)

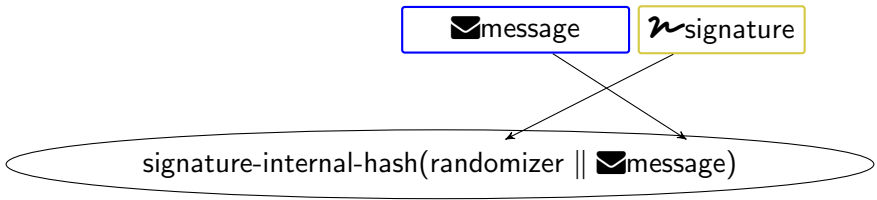
New KEM Interface

- 2 new functions for encapsulation, decapsulation

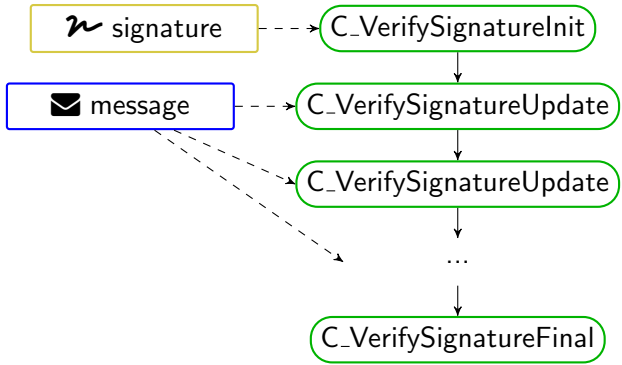
```
C_EncapsulateKey(
CK_SESSION_HANDLE hSession,
CK_MECHANISM_PTR pMechanism,
CK_OBJECT_HANDLE hPublicKey,
CK_ATTRIBUTE_PTR pTemplate,
CK_ULONG ulAttributeCount,
CK_BYTE_PTR pCiphertext,
CK_ULONG_PTR pulCiphertextLen
CK_OBJECT_HANDLE_PTR phKey
)
```

New Signature Verification Interface

- PKCS#11 v3.2 (draft) introduces new interface **C_VerifySignature** (+ -Init, -Update, -Final) which sets signature before feeding multi-part message
- reason: all hash-based signature schemes prefix a randomization string ("randomizer") to the signed data
- randomizer is encoded in the signature



Multi-Part Operations



Online and Multi-Part Operations

- Online operation
  - = process arbitrarily large data with constant memory requirement
- PKCS#11 multi-part operation
  - = send message parts in a series of PKCS#11 commands
- without multi-part operation, whole message must be held in memory by PKCS#11 client program[1]

[1] PKCS#11 is the specification of a programming interface

Single and Multi-Part Verifications in PKCS#11 v3.2 (and v3.1)

Algorithm	**multi-part sign**	**multi-part verify**
ML-DSA	✔	✔ $\mathcal{M}\|\|\boxtimes$ (?!)
SLH-DSA	✔	✔ $\mathcal{M}\|\|\boxtimes$
XMSS	✖	✖
LMS/HSS	✖	✖

external-μ

- integration of ML-DSA external-μ computation in v3.3
  - prior publication in committee note[2]

[2] discussion on pkcs11-comment list (requires free OASIS account)

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	🔑[3]	✔	〰‖✉/✉‖〰(!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	🔑[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	〰‖✉/✉‖〰	✖	✖	✖	✔	〰‖✉/✉‖〰
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	〰‖✉/✉‖〰	✖	✔	🔑[3]	✖	✖

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	✔[3]	✔	〰‖✉/✉‖〰 (!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	✔[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	〰‖✉/✉‖〰	✖	✖	✖	✔	〰‖✉/✉‖〰
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	〰‖✉/✉‖〰	✖	✔	✔[3]	✖	✖

🔑‖✉/✉‖🔑: possible if public key encoded before message, not otherwise. Corresponds to all PKCS#11 signature verification functions

〰‖✉/✉‖〰: possible if signature encoded before message, not otherwise. Corresponds to C_VerifySignature in PKCS#11

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	🔑[3]	✔	〰‖✉/✉‖〰 (!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	🔑[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	〰‖✉/✉‖〰	✖	✖	✖	✔	〰‖✉/✉‖〰
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	〰‖✉/✉‖〰	✖	✔	🔑[3]	✖	✖

Without short message representative (computed "externally"), full message needs to be sent to HSM.
Usage of short message representative – if available – prevents that.

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖✍	✖	✔	✔[3]	✔	✍‖✉/✉‖✍(!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖✍	✔(μ)	✔	✔[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	✍‖✉/✉‖✍	✖	✖	✖	✔	✍‖✉/✉‖✍
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	✍‖✉/✉‖✍	✖	✔	✔[3]	✖	✖

pure ML-DSA: Online operation for signature verification not guaranteed. Hash-ML-DSA not defined for X.509 or CMS. But Hash-ML-DSA available for use in EE certificate for other protocols.

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	🔑[3]	✔	🔑‖✉/✉‖🔑(!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	🔑[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	🔑‖✉/✉‖🔑	✖	✖	✖	✔	🔑‖✉/✉‖🔑
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	🔑‖✉/✉‖🔑	✖	✔	🔑[3]	✖	✖

Very nice from efficiency point of view.

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	🔑[3]	✔	〰‖✉/✉‖〰 (!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	🔑[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	〰‖✉/✉‖〰	✖	✖	✖	✔	〰‖✉/✉‖〰
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	〰‖✉/✉‖〰	✖	✔	🔑[3]	✖	✖

Bound to C_VerifySignature interface.
Why signature before message for ML-DSA?
Unnecessary restriction. Should use traditional C_Verify interface. (Requested correction I)

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖✎	✖	✔	🔑[3]	✔	〰‖✉/✉‖✎ (!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖✎	✔(μ)	✔	🔑[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	〰‖✉/✉‖✎	✖	✖	✖	✔	〰‖✉/✉‖✎
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	〰‖✉/✉‖✎	✖	✔	🔑[3]	✖	✖

Stateful hash-based multi-part is possible and necessary. (Requested correction II)

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

Overview of Signature Schemes' Memory Efficiency

	Algorithm specification (NIST, IETF, final)			X.509 (cert, CRL, CMS) spec. (LAMPS drafts, not final)			PKCS#11 v3.2 draft / v3.3	
	short msg. repr.	**sign-online**	**verify-online**	**short msg. repr.**	**sign-online**	**verify-online**	**sign multi-part**	**verify multi-part**
ML-DSA	✖	✔	🔑‖✉/✉‖🔑	✖	✔	✔[3]	✔	✔‖✉/✉‖✔ (!?)
"μ-ML-DSA"	✔(μ)	✔	🔑‖✉/✉‖🔑	✔(μ)	✔	✔[3]	(v3.3)	(v3.3)
Hash-ML-DSA	✔(hash)	✔	✔	N/A	N/A	N/A	✖	✖
comp.-sigs-07	✔(hash)	✔	✔	✔(hash)	✔	✔	N/A	N/A
SLH-DSA	✖	✖	✔‖✉/✉‖✔	✖	✖	✖	✔	✔‖✉/✉‖✔
Hash-SLH-DSA	✔(hash)	✔	✔	✔(hash)	✔	✔	✖	✖
XMSS/LMS	✖	✔	✔‖✉/✉‖✔	✖	✔	✔[3]	✖	✖

Potential for critical performance issues: Firmware / image signature (long-term secure verification).
No pre-hash variants defined. Protocol redesign would currently be needed:
XMSS-sign(hash(rnd || message))
⚠ Parallel definition of "pure" and "pre-hash": need to preclude introduction of signature forgery vulnerability

[3] Depends on protocol. CMS: with SignedAttributes: yes ✔; without SignedAttributes: no ✖

- Further Feedback ?
- Agreement with the requested corrections presented ?