PKCS#11 GOES TO 3.0!!!
ICMC17

Valerie Fenwick, Director of Software Engineering, Intel

18 May 2017

Open Standards Cat

Who am I?

- I've been involved working on the PKCS#11 standard since ~2003
  - Implemented Sun's Userland Cryptographic Framework in PKCS#11 at the same time (libpkcs11). Coincidence?
- Was co-chair of the OASIS PKCS11 TC 2013 to 2017
- Secretary from Feb 2017 to current

PKCS#11 What is it and why use it?

- Vendor independent cryptographic services API
- Available on most (ALL?) operating systems
- Used by Smartcard, browsers, file systems, authentication systems, HSMs, etc
- Standard developed by multiple vendors in an **OPEN** standards body

PKCS#11: A Brief History

- Jan 1994: Project launched
- April 1995: PKCS#11 v 1.0 published
- June 2004: PKCS#11 v 2.20 published
  - Several other versions in between
- March 2013: PKCS#11 standard moved to OASIS
- May 2016: PKCS#11 2.40 Errata 01 published

Why 3.0?

- Well, lots of things, but …
- A.5 Key/IV Pair Uniqueness Requirements from SP 800-38D
  - PKCS#11 v 2.40 and older did not allow for internally generated IVs
  - We had never changed function signatures in a dot release

What's in it?

- The expected
  - Algorithm updates, fixes to comply with NIST SPs
- The new ideas
  - Additional functions, data types
- The necessary
  - New function entry points, message based encryption

Algorithm Updates

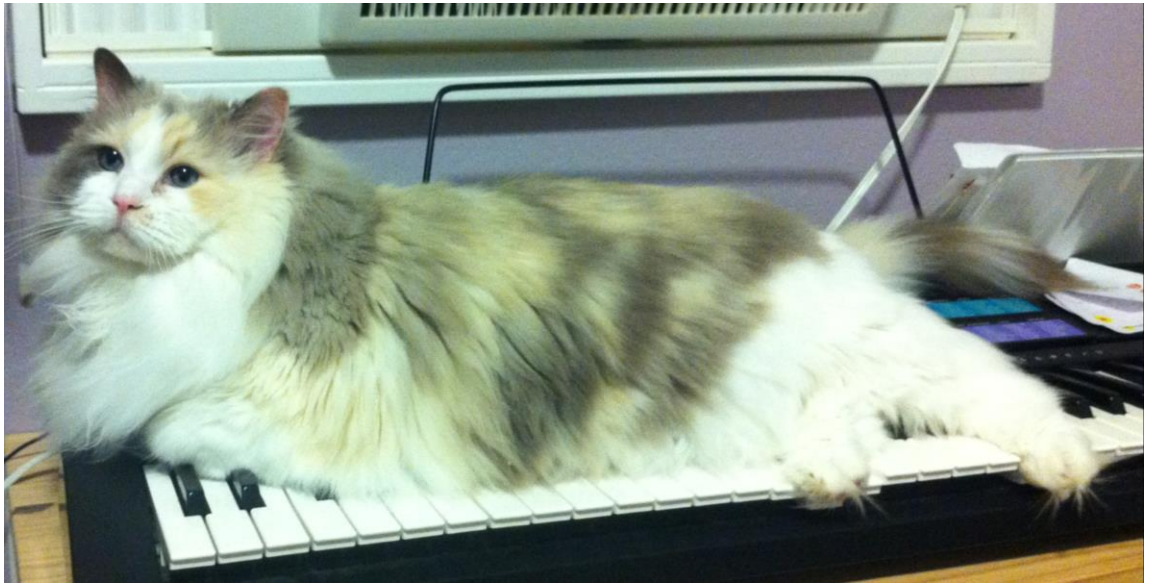
- AES XTS mode introduced (again!)
- ChaCha20
- Poly1305
- SHA3/SHAKE
- Errors:
  - Cleaned up in SHA1/SHA2 text as well as AES-GMAC

CKA_UNIQUE_ID

- Unique Identifier assigned to the object
  - Unique per session and should be per token
  - Copy, create, etc will create this unique ID
- Issue discovered by developers trying to use PKCS#11 underneath KMIP implementations

CKA_DERIVE_TEMPLATE (proposed)

- Used to partition the derivation keys so they can only derive a subset of keys



RSA and ECDH AES Key Wrap Improvements

- To help implementations comply with NIST SP800-38F (Dec 2012)
- Calls for a temporary random AES key that is inaccessible to the user to wrap the target key using CKM_AES_KEY_WRAP_KWP

KDF improvements

- ANSI-style KDF differed from NIST SP800-56A KDF
- New KDF types added: CKD_SHA*_KDF_SP800 for SHA1, SHA2 and SHA3

Flexible Symmetric Key Derivation Mechanism (proposed)

- Introduced to comply with NIST SP800-108
- Allows KDF type, PRF type and PRF input as input

EC Key Generation: Extra bits! (proposed)

- Method specific to and described in FIPS 186-4, Appendix B.4.1
  - Another difference between ANSI and NIST
- New mechanism: CKM_EC_KEY_PAIR_GEN_W_EXTRA_BITS

C_SessionCancel()

- Terminates active session based activities
- Morphed from desire to cancel an in progress TLS operation
- New error code: CKR_OPERATION_CANCEL_FAILED
- C_[Encrypt|Decrypt|Digest|Sign|Verify]Init with pMechanism set to NULL will have the same effect

C_LoginUser() (proposed)

- More fine grained than legacy PKCS#11 authentication support
- Allows for a regular use or an SO (Security Officer) to login to the token

Message Based Encryption

- Allows encrypting multiple messages using the same encryption mechanism and key
- The encryption mechanism can be either an authenticated encryption with associated data (AEAD) algorithm or a pure encryption algorithm.

Message Based Encryption (cont'd)

- New functions:
  - C_MessageEncryptInit
  - C_EncryptMessage (single part)
  - C_EncryptMessage(Begin|Next|Final)
    - Similar for Decrypt, Sign, Verify, MAC

AES GCM changes

- Allows for both specified and internally generated IV
  - The generation method is not defined by PKCS#11
- CKG_GENERATE_COUNTER
  - non-fixed portion of the IV is generated internally with incrementing counter
- CKG_GENERATE_RANDOM
  - Generated internally using a PRNG

Forking behavior change

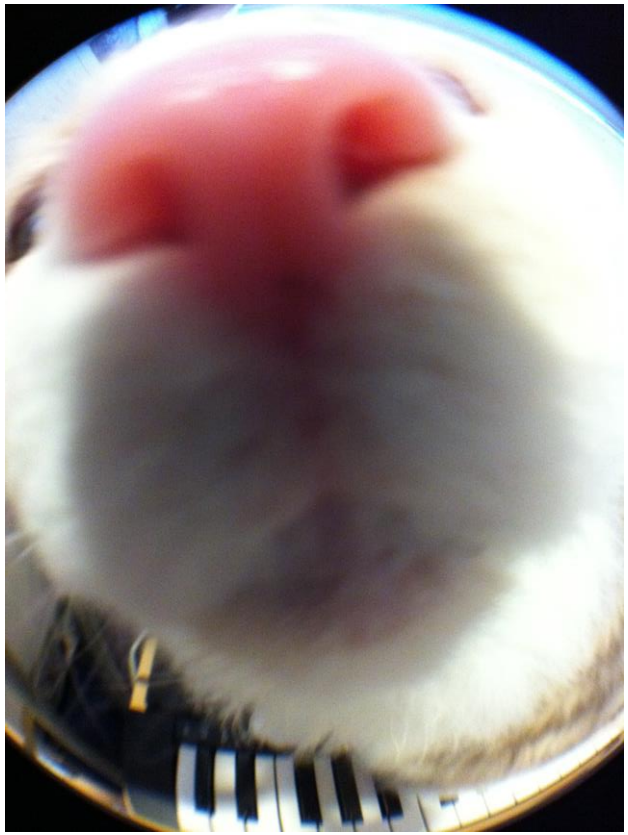
- New flag: CKF_FORK_SAFE_INTERFACE
- Means the returned interface is fork tolerant semantics
- When application forks each process
  - Gets own copy of all session objects, session states, login states, and encryption states
  - Will maintain access to token objects with their previously supplied handles

Expandable Function Table!

- Calling C_GetFunctionList() returns PKCS#11 2.40 errata 01 function list
  - Looks like classic PKCS#11
- C_GetFunctionLists() (plural)
  - Each interface is specified by a string
  - "PKCS 11 2.40" or "PKCS 11 3.0"
  - No support for older standard revisions via this entry point

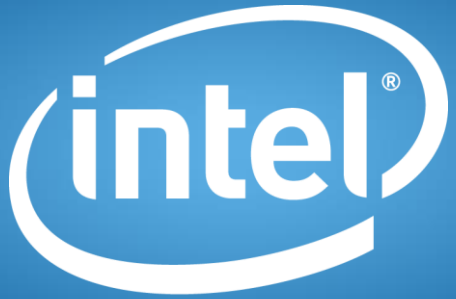
How Can You Help?

- Join OASIS, if your company is not already a member
- Join the PKCS11 TC
- Or, participate via public comment!
  - We'll take your help however we can get it



References

- Wikipedia: https://en.wikipedia.org/wiki/PKCS_11
- PKCS11 TC Public Page: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pkcs11
  - All proposals and approved changes can be found here
- Prior ICMC talk on PKCS#11 2.30/2.40 at ICMC 2014



Software