February 19, 2020 Meeting Minutes - Face To Face

Meeting commenced 9:00 AM PST

- Roll call (Tony) - quorum achieved.
- Valerie taking minutes.

Attendance

- Attendance noted in KAVI

Proposed agenda

- Welcome, roll call & agenda approval
- Status Review
- PR Comments v2.40
- Break
- PR Comments v3.0
- HSS (Michelle)
- Asynch Processing (Jonathan)
- Break
- v3.1 Wiki Review
- PKCS#11 Document Structure
- XML Test Cases
- ICMC Planning
- Status Summary
- Adjourn

Motion to approve Agenda

- Tim moved, Dieter seconded. Comment from Tim: If we get ahead of schedule, can we move the XML topic up? TC agreed. No other comments. No objections or abstentions. Agenda approved.

Motion to approve meeting minutes

- February 05, 2020
- Daniel moved, Greg seconded. No objections, comments or abstentions. Minutes approved.

Status Review (Tony C)

- 2.40 Errata 01 is an official standard
- v3.0 is stuck waiting for one more Statement of Use to become official, needed in the next week to prep for ICMC 2020.
Valerie noted that Intel won't likely be able to meet that deadline.
Tim: Need to use something contained within the spec. Jonathan (ISC) will also pursue an SoU as well.
Tony walked through high level view of process, takes about 395 days, and additionally reviewed our TC's historical cadence.
How long do we want to work on v3.1?
  - Valerie noted we always want to go fast, but we take long. Definitely wants to go faster. Tim noted that we always think of more stuff the longer we go. Also, now that TC is locking in the identifiers immediately, implementation can start more immediately. Tony noted we could also technically have multiple versions going at the same time. General consensus seemed to lean towards setting a deadline.

Public Review Comments for v2.40 and v2.40 Errata 1

- Daniel: Item 17 was defined in the header file, but not described in any of the documents. Doesn't seem to have been resolved, yet. SHould assign someone to verify and if necessary to write up the text. If you don't know how it will be applied, it's useless. we need the text.
  - Bob asked the room if anyone has implemented it, yet? no answers... Bob suggests let's start it now, it will be ready for 3.2. It is likely too complicated for the 3.1 time frame.
  - Tim found one vendor that has implemented it and documented the use - that could be a good first step to doing this.
- Daniel notes that many of the "resolutions" are "will be fixed in the next version", but no clear path that shows they were fixed.
- Tony did a cursory review and believes most have been addressed. Daniel has asked that the TC verify that is true.
- **AI: Oscar** will take a pass at the 2.40 and 2.40 errata 01 comments, see what has been addressed and get back to the TC. (Tony will share the links)

XML Test Cases (Tim H)

- We're ahead of schedule, so moving Tim's talk up earlier as he's in Australia.
- It is possible to write code that can run anywhere and be portable, but it's also easy to accidentally diverge. Learnings from KMIP was to have a way to test it.
- Solution: Data Driven Testing. Discussed in 2016, 2017, 2018, demonstrated in 2019 F2F and at ICMC. Lots of good input to feed into this. There was general consensus in 2019 F2F to move forward.
- Tim's presentation has examples - using same names from the specification. We now have something that can feed into a program.
- Contains both request and expected response. We were able to demonstrate this at ICMC 2019 with 4 vendors (Cryptsoft, Intel, Utimaco and nCipher). Additionally, were able to show how PKCS#11 and KMIP worked together.
- Don't have "find me a token" option, which requires a lot of work to find the token we are looking for. In demo used slot index.
- Tim walked through his examples.
- Discovered at ICMC 2019 that we had incompatible RSA KeyGens - different allowed values in templates, some in surprising ways (like defaults not being allowed, different enforcement levels). Most vendors will set what you provide, whether it is logical or not. this can lead to interoperability issues.
  - The changes to work around are straightforward, but sometimes change between one vendor's own release.
Given how mature PKCS#11 is, we should be more consistent. How to fix? More testing! We can all contribute test cases, even before your code is written - as it's driven from the spec.
- Would love to do this work with another vendor, gets more cross pollination of ideas (and share the load).
- Feedback thusfar is people are really annoyed with CKR_TEMPLATE_INCONSISTENT, leads to lots of trial and error. Some vendors have elegant solutions to this, but they are not mirrored in the spec. The problems aren't new, have been solved, but haven't been pushed into the specification.
  - this was a goal for 3.0... but we didn't get to all of them. something we always need to keep our eyes open for these. Vendors: Please consider contributing your cool fixes. Tim suggested reaching out directly to vendors. Tony suggested a brain storming wish list.
  - Tim has tried reaching out to vendors, but hasn't seen progress. May try approach of "this one thing" to each vendor, see if a focused request is more productive.
  - Tim said at a very early F2F we discussed a list of "annoying things we want to fix"... that are still not fixed. Hopefully testing will help here.
- Tim proposed opensource concept: you can't just add a new feature, unless you fix an issue at the same time/first.
- Valerie & Tim to collect these older known "annoyances" and bring them to TC to think about addressing.
- We don't see the same issues on KMIP, since the test cases were around much longer, no legacy implementations.
- Tim: Test cases should be straightforward enough to help people discover what is really not working.
- Tim noted that historically we would try to interoperate with Netscape.
- Tim: As a TC we should probably determine which are the right ways when there are divergence.

Break!

Public Review Comments for V3.0 (Tony)

- Working off of the public comment list.
- C_Decrypt/DecryptFinal: Buffer too small. Tim to make sure it's on the 3.1 wiki.
- CKM_AES_GCM & CCM possibily vulnerable to 2 pad attack. Tim to add to the 3.1 wiki as something that we need to either address or explicitly not address (as opposed to forget)
- Discussion if anyone is using this, and possible solutions (internal counter tied to the key...). The token could maintain state internally.
  - Tim suggests going back to Robert and asking for his proposal on how he'd like to fix it.
  - Daniel noted there's a lot of context here suggesting an idea for fixing. More discussion ensued, noting issues with how NIST also handles this.
  - Bob - Is anyone actually using wrap and unwrap at this point?
    - Valerie notes maybe the fix is to deprecate the function on this mechanism?
    - Daniel noted that he has implemented this - what do they do in FIPS mode? NIST don't support it. Daniel found the history for why they implemented it: 2 separate customers requested, so it's likely they are...
using it. So, deprecation is not a good option. Discussed more design options.

- If we don't have a passionate owner, it simply won't get resolved. Partnering preferred - Tim is willing to partner with someone. No takers.
- **AI: Tony** - draft email to Robert (original commenter) asking if he has a proposal to fix, Tim tracked on wiki.

### HSS (Michelle)

- It's a quantum safe algorithm, but slow, so generally will only be used for things like code signing. Michelle gave an overview of the algorithm. It's stateful (linked Merkle tree). Not suitable for TLS.
- Want to add the NIST reference into the proposal, now named and have identifiers assigned. 📄 NIST SP-800-208
- Tim: There's no ASN1 defined for this, like we have for ECC. It's non-specified and unusable as an octet blob. We need to have this broken out.
- Mike: Look at RFC 8708, ASN1 for HSS and CMS
- Bob notes that PKCS#11 typically prefers not to crack ASN1 (except where it does). 😊
- Some tricky issues with backups. Tim noting we need to not prohibit moving from one token to another. Tim said we're not here to preclude an insecure usage. Bob noted that if you reuse the key, someone could compromise your signatures - the key is dead.
- What about long living/deployed things (like airplanes). Bob: If it's hard to change the key, this algorithm is not appropriate.
- KMIP has remote update.
- Greg: Do we put something in there like "beware, don't use this in these contexts" - Tony suggested usage guide? may be useful to note here, as this is very different than the other keys.
  - **AI Michelle**: Add warning to proposal text, as well as reference to the official NIST draft documents.
- The private key is the seed - needs to be protected.
- Any common components that can be broken out should be- how they are stored is up to each vendor.
- The state has to go with the key.
- **AI Michelle**: should specify key size min/max to 0
- **AI Michelle**: Send all new identifiers to Bob, he will assign identifiers.
- SP-800-208 is out in draft, open for comments until next Friday. Should be pretty stable here. There is a note in the draft about exporting keys - forbidding it. Heard it may take 2 years (from 6 months ago) to be finalized. If we allow exporting keys, should note in our spec that the action is not allowed in FIPS mode.
  - Tony: Given our cadence, we should be fine 😊
  - Bob: We also have an RFC, so we have a lot to go on.
- Dieter: if export is actually prohibited, then we don't need an definition for export. Bob agrees, but we need definitions for things that are public/allowed.

### Lunch Break

### Async Processing (Jonathan)

- HSS key generation could take days, but C_GenerateKey is a blocking process....not ideal
- Add functions - possibly: C.GenerateKeyPairStart, C.GenerateKeyPairStatus, but seems to be application dependent.
  - Mike: this could be an implementation specific problem, not a specification. Bob: could there be an error code we could use/check?
  - Tim: C_Login could also take a long time (for example, waiting for an operator to interactively put in a pin or passwd)

- Some discussion on C_CancelFunction (obsolete)
- Could be lots of ways (like FindObjects) to see if it's done... but some of those are not really easy or reliable to use.
- other async functions are application specific, so not necessarily widely adopted
- How often do you need to do this? Do we need to do all of them all the time? The top ones, all the time. Depending on the tree size, impacts when things need to be generated.
  - Tim: not sure we can assume the application will always be around. This was an area we made issues in KMIP, let's learn from that. In KMIP allowed the server to determine if something should be async or not, the client doesn't get a choice (must handle both). Not a great model - the client needs to know what it's going to get.
- is there a need for a unique identifier? does it need to work for all functions, or just keygen? anything that could take a long time. What about Get/SetOperationState? Maybe... if we change the semantics....
- Bob - some good ideas kicking around, would feel more comfortable if someone who has already solved this for KMIP would step up and look at this....
- TC Generally agreed to need for something generic (maybe session based) vs specific, avoiding mistakes KMIP made, with a way to query status.
  - Al Tony: Add this item (async) to 3.1 wiki "to-do" list.

V3.1 wiki review

- Too many TBA for Tony's liking.
- we've started digging into some of these items.
- Tim's XML item - Tim is looking for a dance partner to help create this. (see notes from Tim's presentation above). Tim will start first pass, given Bob is happy with where we are.
- KMIP mappings - About to move to bottom of "weird stuff" (aka not 3.1)
- BOB commits to update IKE document with Daniel's comments by March 15.
- Michelle's HSS: see above discussion. updates by March 15.
- C_SetPIN: does it need to be deprecated now that we have C_LoginUser? which users pin, now that we have multiple users? likely need a new function with multiple parameters. If we remove it, we'll hear more... CKU_CONTEXT_USER will not be addressed in v3.1, but in v3.2 will consider adding C_SetPINUser and deprecating CKU_CONTEXT_USER from v3.2
  - "Suggestion of 2 new functions" aka "Encrypt by Handle". Tim does not think we should consider until 3.2. Bob & Daniel think it doesn't even make sense. Requestor mixing up functionality.
  - Al Tony: Put this in the "no, we're not going to do this" pile, Tony to draft a response to Wully, including a brief "why".
- C_Decrypt/C_DecryptFinal issue around return buffer handling. Small semantic difference between when you are asking for size vs buffer is too small. You might have to do a decrypt to see what the buffer size should be. Discussion on the wording - can we find this in spec? Base Spec, section 5.2 😐 There is some inconsistency, but there's further discussion lines 2157-2165. Could be some gray area on where to buffer things.
  - Does TC think it's acceptable to loosen the buffer requirements? Seems like these are not consistently implemented. Tim updated wiki for Bob to fix NSS and fix the
Without having done the actual decrypt, we don't know the exact length and we aren't requiring applications to keep an internal buffer. **AI Bob** will have an update for document by March 15, no promise on when NSS will be updated.

- **CKM_ECDH_AES_KEY_WRAP/CKM_AES_KEY_WRAP_PAD**: very detailed comment. Historically, some interesting details. Daniel noted there could be a breaking change here. Hamish noted there is a conflict with RFC 5649.
  - **AI Dieter**: Update specification to remove backwards compatibility issues with **CKM_AES_KEY_WRAP_PAD** and and **CKM_ECDH_AES_KEY_WRAP / CKM_RSA_AES_KEY_WRAP**
  - **AI Bob**: Allocate a new identifier for **CKM_AES_KEY_WRAP_PAD** and **CKM_ECDH_AES_KEY_WRAP / CKM_RSA_AES_KEY_WRAP** as needed after Dieter completes his updates. Keep old values reserved, as there are existing applications and wrapped keys in the field.

- **HKDF issue**
  - Bob could update the spec, and/or the profiles (when this can be used to extract).
  - We can generate HKDF data, but you may not want to use the keys provided. The spec says tokens can filter out, must be able to use this label. Martin's comments noted that the label is specified one way (breaking string cmp), has a typo, and misses the quick interface.
  - You don't have to check the label, you could use the data to get the key data out.
  - Tim: we may not want to dictate the specific usage.
  - Bob: will update to have tokens advertise the tokens they support.
  - **AI Bob**: will update specification and profiles for HKDF by April 15, 2020.

**Break!**

**PKCS#11 Document Structure**

* Historically the documents were too large for Word. Tony is proposing rolling current mechanisms back into the base specification, and adding a test cases document.

- What to do with historical? We can always announce deprecation, but not remove it right away. But need to keep the header files in sync.
- Valerie likes the proposal. It comes up a lot about "how do we stop people from using deprecated algorithms", Tony notes we can't stop the folks implementing them in the field but by calling them obsolete, we are sending a message.
- There are things in the Historical Mechanisms document that just are not used anymore (like RC2 or algorithms that had names but never saw the light of day)
- Tim: could we have a "recommended configuration" that allows people to turn back, if absolutely needed? We don't have anything like this, now, but may be nice
  - Bob: we have something in NSS, where we can turn things off. Tied into os for security levels.
  - Tim: doesn't think we should be setting the policy, but creating a mechanism for doing so. Can also encourage vendors to move forward.
- Bob likes the separate documents, but isn't the one that has to get all the templates, etc.
- Tim: suggests we merge them, and if it sucks, change it next time :-). They were originally split due to a tooling issue (Word was crashing).
- Dieter said references are easy to break in this mode.
• Tim moves to merge the base specification and the current mechanism document into a new document simply called the specification. Greg seconded. Bob abstained. No other comments, abstentions or objections. Motion passed.
  o Dieter noted that some sections have way to many subsections. compaction would be nice. Tony notes those are editorial changes, no need for a motion, as long as there are no material changes. Tony suggests we do it as a separate update.
  o Valerie called out a comment Tony made: do the merge as a separate change, do it as a style change only (with change bars) - no technical content changed. Send as a review.
  o Tim amends that: also do it in a section or 2 first, send for review to make sure all like the style (before you spend too much time rotating on a style others may not like).
  o Nobody expects things like the SHA sections to be compressed all at once.
  o **AI Tony & Dieter:** Take a first pass at proposed merge and section cleanup in one or two areas, send to TC for style review.
• Now a test case document - will have normative text and potentially a zip file with tests. Would this mean you are compliant if you pass? only if you meet the conformance statements.
• **Tim moves to create a test cases document, which would be informative, designated as a committee note in its final form. Anthony seconded. No objections, abstentions or comments. Motion approved.**
  o Tim has volunteered as editor. Co-editor would be nice.

**Interop Planning ICMC 2020**

• Last year the booth was a bit small, have a larger space for 2020.
• April 28-May 01. PKCS#11/OASIS have a booth and hope to have some standard test cases for that. Email came from Jane Harnad this morning! Please consider joining and participating in the interoperability demonstration!
  o International Cryptographic Module Conference Website.
  o Price range is just based on the booth decoration/number of participants.
  o Talk to Tony offline for more details.
• We have a 1/2 day track dedicated to OASIS topics!!!
  o been offered a panel session. If you are participating in the interop, you could get a seat on the panel to do a Q&A. topic ideas?
    ▪ Moral imperative for degradation of shitty algorithms.
    ▪ How to enable people to move forward, algorithm agility should be there - but it's not in practice.
    ▪ 3.1 will have first quantum safe algorithm.
    ▪ Bob has some good history on why agility is hard.
    ▪ Deplorable applications, what we can do about it.

**Session Review**

• Moved to offline, in interest of time.

**Next meeting**

• March 4, 2020
Call for late arrivals

- Gerry

Thank you

- Tony moves to thank Bob & Red Hat for hosting and catering! Jonathan seconded. No objections, abstentions or comments. Motion approved.

Motion to Adjourn

- Tim moved. Gerry seconded. No objections, comments or abstentions. Meeting adjourned.

Meeting Adjourned at 4:20PM PST