Comments Received on SP 800-57, Part 1

From: "Harris, Michael W. (CDC/OCOO/OCIO)" <fnb0@cdc.gov>
Date: Monday, October 26, 2015 at 11:59 AM

CDC has no comments to provide on the *Draft Special Publication 800-57 Part 1 Revision 4, Recommendation for Key Management: Part 1: General*.

From: "Austin, Richard (Technology Office, Cyber Security)" <raustin@hpe.com>
Date: Tuesday, October 27, 2015 at 11:20 AM

#	Type	Page #	Line #	Section	Comment (with rationale)	Suggested Change	Resolution
1	E	20	--	Glossary	"Identifier" – it is not immediately clear how a "bit string" relates to a person.	Add a footnote on "person" explaining that the bit string might be derived, for example, from a biometric such as a fingerprint.	An identifier is not a password or biometric information about a person; it is the stated username, identity or subject name (e.g., in a certificate); no action taken.
2	T	21	--	Glossary	"Integrity protection" is stated as being equivalent to "Integrity authentication". "Integrity authentication" is one means of demonstrating "integrity protection" but they are not the same.	Do we really need a glossary entry for "integrity protection"? I'd suggest deleting it.	The term is used in the document; no action.
3	T	23	--	Glossary	"Operational period" is defined but it is not clear how it relates to Figure 1, p.47.	Either clarify its meaning versus "cryptoperiod" or delete the term from the glossary.	"Operational period" is not included on the glossary. No action taken.
4	G	29	91	3.2	There is a muddle in the document between MAC and HMAC extending from the glossary through the usage of the terms elsewhere.	Generally, to be useful in assuring integrity, MAC's have to be HMAC's or protected by a digital signature. I'd suggest adding some explanatory text around MAC and how HMAC protects the code from modification. From that point onwards, I would use HMAC in the document.	A MAC can be generated using HMAC, CMAC or GMAC. See Section 4.2.3. No action taken.
5	T	230-233	30	3.5	Non-repudiation provides assurance that a subject performing an action may	Use a better example such as the classic "Jane buys 100 shares of stock and after the shares tank	The example cited is, in essence, a contract. No action taken.

					not later deny having performed that action. The example given of signing an contract is misleading as quite commonly certificate authorities and organizations limit the obligations conferred by a particular signature.	denies having authorized the purchase.	
6	T	349-350	33	4.1	“Difficult to reverse” – “reverse” suggests recreating the input from the hash which is impractical. What is being described is actually first preimage resistance and second preimage resistance.	Replace “difficult to reverse” with “hard to duplicate” or something similar. Preimage resistance is explained in lines 392-393.	Changed to "... difficult to find an input that will produce a given output", which is consistent with the glossary.
7	E	364-367	33	4.1	This is a topic that confuses my students – the difference between a HMAC and a digitally signed message for integrity assurance (which relies on asymmetric crypto)	Insert a footnote noting that digitally signed messages use asymmetric crypto to provide integrity assurances and point the reader to 4.4.	Symmetric-key algorithms can generate MACs based on either block ciphers using the MAC mode or based on hash functions using HMAC. See Section 4.2.3, as referenced.
8	E	543	38	4.2.5.4	“Integrity protect the key to be protected” is an awkward reading.	Substitute “protect the integrity of the key”.	Removed "the key to be protected" from the sentence.
9	T	543-544	38	4.2.5.4	It is not clear in the document how key unwrapping verifies the integrity of the key. Some suggestions are made in 5.4.1 but this is much later in the document.	The glossary entry also asserts that key wrapping provides integrity protection but doesn’t specify how integrity protection is provided. Lines 646/647 on page 40 indicate that integrity protection is optional. Add material describing how integrity is protected and	See the specifications in SP 800-38F, as referenced, for more detailed information.

						harmonize the different sections as to whether it is optional or not.	
10	E	899	47	5.3.5	The figure is identified as "Symmetric Key Cryptoperiod" but the concepts also apply to asymmetric key pairs as discussed in lines 842-848	Re-title the figure as "Key Cryptoperiod" as it applies to both types of cryptography.	Figure 1 applies to a (single) symmetric key, which is used to both apply protection and to process already-protection information (e.g., to decrypt already-encrypted information. In the case of digital signature and key-transport asymmetric-key algorithms, each key of the key pair has its own cryptoperiod, which is either an originator-usage period or a recipient-usage period, depending on the cryptographic operation in which the key is used. For key-agreement algorithms, the terms "originator-usage period" and "recipient-usage" period don't quite work because of the way the keys are used in the algorithms.
11	G	N/A	N/A	N/A	As described in NIST 800-88 R2, cryptographic erases is a very efficient way of sanitizing large volumes of data. In order for this technique to be applied, effective key management is an absolute requirement.	Consider adding a use case to the document noting that deliberate destruction of the keying material is an effective sanitization technique and provide guidance on key management capabilities to support it (audit, proof of sanitization, etc.). A good place for such a discussion might be around 6.2.2	The sanitization of large volumes of data protected by cryptographic is out-of-scope for SP 800-57. However, a paragraph was inserted at the end of Section 6.2.2.1 to mention the use case and point to SP 800-88.
12	E	2175	86	7.1	The term "certified	Though it's longer in length, I'd	Done.

					asymmetric key" seems stilted terminology for keys associated with a certificate.	suggest substituting "asymmetric keys associated with a certificate" or something equivalent.	
13	T	2375-2377	92	7.6	It should be noted that when cryptographic erase is used as a sanitization method, proof of destruction of all copies of the relevant keys must be available (see 800-88r2 for details).	Add a note to the effect that some cryptography uses, such as cryptographic erase, require that certain key metadata be retained	Inserted "for audit purposes" to the third line, which is consistent with the wording in Section 8.4, which is referenced.
14	E	N/A	94	Figure 5	Note that the outgoing line from "Suspended" toward "Compromised" extends inside the "Suspended" box.	Remove the portion of the line inside the "Suspended" box.	Done.
15	T	2900-2901	104	8.1.5.3.2	It is not clear to me, possibly due to my ignorance, why IV's need protection. As noted earlier, IV's are often transmitted in the clear during establishment of a cryptographic session.	If the intent is to assert that IV's require protection then insert material explaining why that is so.	A sentence was added to Section 8.1.5.3 that points to Table 6 in Section 6.1.2 for the required protections. In the case of IVs, integrity protection is required.
16	T	2223-2233	88	7.2	The "suspended" state adds risk and complexity with little discernable benefit over "Deactivated" except the counterintuitive ability to transition back to the "Active" state. The example of an employee going on leave of absence is unconvincing – if the leave is long enough to justify a change in the key status, it	Delete the "Suspended" state.	The suspended state is sometimes used by a PKI. No action taken.

					could easily be deactivated and a new key issued on their return rather than complicating the key management process.		
17	E	2223	88	7.2	This transition is labelled as “Transaction 7” rather than “Transition 7”.	Correct the labeling to “Transition 7”.	Correction made.

From: "Lloyd, Paul C (Cyber Security)" <paul.lloyd@hpe.com>
Date: Thursday, October 29, 2015 at 7:20 PM

#	Type	Page #	Line #	Section	Comment (with rationale)	Suggested Change	Resolution
1	T	20		2.1	Defn of hash: To be complete and precise, the arbitrary length may be bounded	"A function that maps a bit string of arbitrary, though possibly bounded, length"	Inserted "(although bounded)" after "arbitrary."
2	T	20		2.1	Defn of hash: To be complete should there be explicit mention of resistance to 2nd preimage attacks?	3. Given a message m1, it is computationally infeasible to find a message m2 with the same hash	Both pre-image resistance and 2nd pre-image resistance are covered under the first listed property (one-way). No action taken.
3	E	21		2.1	Defn of integrity authentication "that data has" data is technically plural	"data have" or "a data item has"	While data is technically plural, it is commonly used as singular these days. No action taken.
4	E	25		2.1	Defn of security strength: typo: "not longer"	"no longer"	Corrected.
5	E	28		2.2	SMIME is officially S/MIME	S/MIME	Corrected.
6	E	37	784	4.2.5	Typo: ", ,"	", "	Line 498 corrected.
7	E	39	937	4.2.7	"additional entropy never be introduced again	"additional entropy may never be introduced again	Line 591 corrected.
8	T	41	1038	5.1.1	It can be argued that IVs are not true keying material. In fact, this doc defers IVs to §5.1.2. This appears in multiple places in this document	Refer to IVs as something like a parameter	IVs have historically been included in the definition of keying material. See Section 2.1. No action taken.
9	T	46	1310	5.3.4	"The sum of the validity periods"	Resolve, perhaps by not expressing as a sum	The sentence in lines 877 to 879 has been reworded: "The range of

					Is there an assumption about these certs having contiguous validity periods? For example if cert 1 is valid 2015-10-01 to 2015-10-31 and cert 2 is valid 2016-10-01 to 2016-10-31, how do we reconcile this if the intended cryptoperiod of the key is 2015-10-01 to 2015-12-31? If we approach things in purely arithmetic terms (a sum as written here), we might not get what we really intended. Is the intent that no cert shall have a *notAfter* field that exceeds the end of the key's cryptoperiod? Remember, the earlier definition of cryptoperiod simply refers to a "time span." Does this definition have any assumption about being a CONTIGUOUS period of time?		time covered by the validity periods of the original certificate and all renewed certificates for the same public key **shall not** extend beyond the beginning and end dates of the cryptoperiod for the key of the key pair used to apply protection (i.e., the key with the originator-usage period)."
10	T	56	1786	5.3.7	Another example of referring to an IV as keying material	Perhaps the title of §5.3.7 could be changed to mirror §5.1.2 (Other Cryptographic or Related Information)	Done.
11	T	56	1795	5.4	If we agree that IVs are not keying material (per §5.1.2), then do we need something	Perhaps a new sub-§ in §5.4	These particular assurances have been addressed in a number of publications (e.g., SP 800-89, SP

					dedicated in §5.4 to speak to their assurance?		800-56A and B). No action taken.
12	T	92	3530	8	The text now includes the deactivated state in the operational phase, but Figure 5 below only shows the deactivated state in the post-operational phase	Reconcile	In item 2 of Section 8, the deactivated state was removed from the operational phase.
13	T				Although the document provides much guidance on the topic of establishing trust/assurance in public keys, the document does not explicitly mention the now contemporary practice of certificate pinning	Give consideration	This document cannot address every issue associated with key management. SP 800-57, Part 1 is intended as a general guide for understanding key management. Pinning is a TLS issue. No action taken.

From: Lars Nielsen <s042903@student.dtu.dk>
Date: 10/30/15

Hash: SHA1

This message is included in the attached template as well:

- ----- # 1-----
Type: T
Page # 65
Line # 1558 (2281 in diff document)
- -----Comment (with rationale)-----
2030 is very far ahead, the previous milestones were 2010, ’11-’13, ’14-’30,’31+
With Moore’s law the supercomputers of today will be desktop computers in 14 years (128 times the GPU in the same space).
Following the Moore rationale there should be an extra bit of security each 2nd year.
To be cautious, a bit could be added each year, giving the following suggestion.
With “Disallowed” being 14 bits lower than required and legacy spanning the 14 years in between.

Keys of sizes not conforming to byte sizes are odd, but gives an easily understandable rule and makes sense in regards to number of secure bits in truncated messages as described in SP 800-107, section 5.
1

- -----Suggested Change-----
Required number of secure bits:
2030: 112 bits
2029: 111 bits
2028: 110 bits

…
2020: 102 bits
2018: 100 bits
2016: 98 bits
2014: 96 bits

Disallowed:
2030: 98 bits
2029: 97 bits
2028: 96 bits
…
2020: 88 bits
2018: 86 bits
2016: 84 bits
2014: 82 bits

- ------ # 2-----
Type: T
Page # 65
Line # 1558 (2281 in diff document)
- -----Comment (with rationale)-----
Alternative values, based around 128 bit requirement by 2031
- -----Suggested Change-----
Required number of secure bits:
2030: 128 bits
2029: 127 bits
2028: 126 bits
…
2020: 118 bits
2018: 116 bits
2016: 114 bits
2014: 112 bits

Disallowed
2030: 114 bits
2029: 113 bits
2028: 112 bits
…
2020: 104 bits
2018: 102 bits
2016: 100 bits
2014: 98 bits

- ----- # 3-----
Type: T
Page # 65
Line # 1558 (2281 in diff document)
- -----Comment (with rationale)-----
Alternative, centered around a rule that is easy to recall:
Required number of secure bits:
Years after 2000 + 100

Legacy until:
Required number of secure bits -14
or: Years after 2000 + 100-14
- -----Suggested Change-----
Required number of secure bits:
2030: 130 bits
2029: 129 bits
2028: 128 bits
…
2020: 120 bits
2018: 118 bits
2016: 116 bits

2014: 114 bits

Disallowed
2030: 116 bits
2029: 115 bits
2028: 114 bits
…
2020: 106 bits
2018: 104 bits
2016: 102 bits
2014: 100 bits

#	Type	Page #	Line #	Section	Comment (with rationale)	Suggested Change	Resolution
1	T	65	1558 (2281 in diff document)	5.6.2 (Table 4)	2030 is very far ahead, the previous milestones were 2010, '11-'13, '14-'30,'31+ With Moore's law the supercomputers of today will be desktop computers in 14 years (128 times the GPU in the same space). Following the Moore rationale there should be an extra bit of security each 2nd year. To be cautious, a bit could be added each year, giving the following suggestion. With "Disallowed" being 14 bits lower than required and legacy spanning the 14	Required number of secure bits: 2030: 112 bits 2029: 111 bits 2028: 110 bits … 2020: 102 bits 2018: 100 bits 2016: 98 bits 2014: 96 bits Disallowed: 2030: 98 bits 2029: 97 bits 2028: 96 bits … 2020: 88 bits 2018: 86 bits 2016: 84 bits	The timeframes in Table are, at best, "guesstimates." The dates will be refined as more definitive results are available. The dates are provided to give a heads-up that the increased strengths will be required over time. No action taken.

					years in between. Keys of sizes not conforming to byte sizes are odd, but gives an easily understandable rule and makes sense in regards to number of secure bits in truncated messages as described in SP 800-107, section 5.1	2014: 82 bits	
2	T	65	1558 (2281 in diff docum ent)	5.6.2 (Table 4)	Alternative values, based around 128 bit requirement by 2031	Required number of secure bits: 2030: 128 bits 2029: 127 bits 2028: 126 bits … 2020: 118 bits 2018: 116 bits 2016: 114 bits 2014: 112 bits Disallowed 2030: 114 bits 2029: 113 bits 2028: 112 bits … 2020: 104 bits 2018: 102 bits 2016: 100 bits 2014: 98 bits	
3	T	65	1558 (2281	5.6.2 (Table 4)	Alternative, centered around a rule that is easy to	Required number of secure bits: 2030: 130 bits	

			in diff docum ent)		recall: Required number of secure bits: Years after 2000 + 100 Legacy until: Required number of secure bits -14 or: Years after 2000 + 100-14	2029: 129 bits 2028: 128 bits … 2020: 120 bits 2018: 118 bits 2016: 116 bits 2014: 114 bits Disallowed 2030: 116 bits 2029: 115 bits 2028: 114 bits … 2020: 106 bits 2018: 104 bits 2016: 102 bits 2014: 100 bits	

From: Chuck White <chuck@fornetix.com>
Date: Saturday, October 31, 2015 at 10:51 AM
Good Morning NIST 800-57 Team!

On behalf of the OASIS KMIP Technical Committee I have attached our organization's collective comments in regards to proposed changes

Please feel free to follow-up if you have any questions.

Thanks!

Chuck

#	Type	Page #	Line #	Section	Comment (with rationale)	Suggested Change	Resolution
	T	88	2255	7.3	Having a Suspended State that has transitions back and forth from Active, Disabled, Compromised, and Destroyed creates complexity. It is arguable that the Suspended state is not a Key Management state but an authentication\authorization state outside the scope of Key Management.	Remove Suspended State	Figure 3 in Section 7 is provided as an example. A suspended state is not required. Some communities are using it. No action taken.
	T	87	2190	7.2	Addressing key state transitions directly from Active to Destroyed presents operational complexity in regards to connections, management, and implications for creating instability in	Remove State Transition from Activated to Destroyed	No action taken. The transition is appropriate. The actual transition would include any management necessary to make it happen "gracefully", e.g., notifications, etc.

					systems by removing disabled transition for key material from the process.		
	T	88	2289	7.3	State transitions should be unidirectional, non-looping as reflected in the Key Management States. Key management states need to be alignment with key state model to address key transitions. Having a unidirectional model for key management states and a bi-directional model for key states creates systemic complexity and presents the opportunity for unstable states between keys and the systems that manage the keys. A unidirectional Model represents less complexity and greater likelihood for adoption.	Remove Transition from Suspended to Active	The use of a suspended state is optional (see the paragraph under Figure 3). The states and transitions in in Figure 3 are an example. The inclusion of a suspended state has been introduced to some PKIs. No action taken.
	T	91,92	2361 - 2379	7.X	We have seen no justification for dropping the state of Destroyed/Compromised as it is already established and industry has taken steps to implement. As the key state model is a form of a data contract implemented by industry, it is imperative not to remove aspects of the data	Restore Compromised\Destroyed State	Figure 3 is just an example. Other states are allowable (see the paragraph under Figure 3). No action taken.

					contract without a path of deprecation. Having a timeline for removing a given aspect of a data contract allows industry to adapt technology and take steps to implement changes within product release cycles.		
	T	124	3573 - 3576	10.2.3	Open Standards such as Key Management Interoperability Protocol provide a reference model for a communications format that implements alignment with Key State and Key Management States.	Reference KMIP Specification Standard	No action taken.