Comments on NIST Special Publication 800-57, Recommendation for Key Management - Part 1: General

From: "RDavis" <rdavis@femto-second.com>
Date: Sun, 1 May 2005 11:05:44 -0400

Please find the following comments regarding NIST SP 800-57 Part I, Recommendation for Key Management (April 2005).

Page 34, section 4.2.1, reference to SHA-1:

Suggest footnote 30 be used with this first reference to SHA-1. Also, as footnote 30 suggests, the SHA-1 has an effective strength of 69-bits. Therefore, the references on pages 63 and 122 should be corrected.

Page 35, section 4.2.2.3, reference: the same plaintext block will always encrypt to the same ciphertext block whenever the same key is used.

Suggest including when the same initialization vector and the same key is used.

Page 48, section 5.3.6.2.a. reference A long cryptoperiod for the public signature verification key poses a relatively minimal security concern.

Comment: Some digitally signed records may have retention periods measured in centuries. If Moores Law is used to reduce one bit of key strength for every 3/2 years, then there needs to be a mechanism to verify that a key was valid at a given time. Suggest that the NIST come up with recommendations for long retention period digitally signed objects. The time stamping (or notary) technique referenced in 5.3.6.2.b might be a start. However, there would need to be additional cryptographically bound timestamps using current for the time algorithms applied at intervals when the original signature key is at risk. This is to show that at that point in time, the signature was valid. For if the private key is cracked then the public key will validate old signed objects.

Page 54-55, Table 1:

There are currently products on the market, such as the Encrypted File System (EFS), that perform encryption and decryption on the fly. This type of encryption is used to protect information on laptops and for additional access control (for shared machines). As such, the retention period may be long. Suggest that the NIST specifically address this as an exception. For example, if a laptop is stolen, the encryption and key should be sufficient to protect the encrypted data for the life of the data.

Page 64, section 5.6.2:

The dates presented when algorithms should no longer be used are very optimistic. Given that the SHA-1 went from a key strength of 80-bits to 69-bits indicates there may be additional attacks that will effectively reduce effective key strength further. Suggest the NIST move the requirement to use 128-bits well before 2030.

Page 66, section 5.6.3, reference: Other AES key sizes would also be appropriate, but perform a bit slower.

Suggest the NIST provide an estimate of the additional time required so that informed decisions can be made when selecting AES key sizes. That is, if the processing power of computers doubles every 3/2 years, then it might be cost effective to implement stronger algorithms. For example, if my communications channel or disk I/O takes longer that the longer key encryption, they there is no observed difference in key sizes to the computer user.

Page 78, section 6.2.1.2.2.a, reference: Note that a CRC is not sufficient.

While this may be true for a finite set of well defined cyclic redundancy checks, the NIST should be aware that the core of a CRC, the polynomial generator can be randomly generated. Unlike a CRC that the checksum is added to an objects end to product a value of 0 following Mod 2 division, the checksum value could be used standalone with cryptographic strength. If the time to determine what polynomial generator was used to generate the checksum value exceeds the life of the data, then this may indeed be a viable approach. A simple example can be found in Software Checking with the Auditors Aid, Proceedings of the Sixth Annual Computer Security Applications Conference, 1990.

Page 83, section 7.1.5, reference: For example, a signature may be validated if it can be shown that the signed data with its signature has been physically protected since a time before the compromise occurred.

Suggest that the NIST prepare guidance on this. That is, if the control is a cryptographically bound time stamp, this will be important for long retention period documents were the original signer may have passed away.

Page 112, section 9.1, reference: As a minimum, the key management system should account for all individuals who are able to view plaintext cryptographic keys.

Recommend that there should be no user able to see the plain text keys. For example, if its a validated hardware cryptographic module, keys are exported in encrypted format. Suggest replacing this with have access and ability to use cryptographic keys.

Page 114, section 9.3.3.1:

Suggest that the format information was protected in is also important. For example, not many years ago, the military exclusively used the Baudot code to represent information. Today, this format is non-standard. As another example, if ASCII goes away so that only Unicode used, then the bit-wise ASCII data that was protected would be lost. This might be another example where the NIST could provide guidance for: 1) verifying that information was okay at the time of conversion and 2) re-protecting the information following conversion.

Regards,

Dr. Russell J. Davis
Femtosecond Inc.
9747 Water Oak Drive
Fairfax, VA 22031-1029
(703) 282-1837
RDavis@femto-second.com
www.femto-second.com

From: Robert Zuccherato <robert.zuccherato@entrust.com>
Date: Fri, 20 May 2005 10:02:37 -0400

Entrust Comments on NIST SP 800 57 Recommendation for Key Management Part 1 General

This document provides a significant step forward in guidance on key management. Entrust provides the following comments:

1. Footnotes 18 and 21 should add the words “at most” so that they read “The 80-bit security of 2TDEA is based on the availability of at most 240 matched plaintext and ciphertext blocks to an attacker (see [ANSX9.52], Annex B).”

2. This document refers to FIPS 186-3 DSS and NIST SP 800 56 Recommendation on Key Establishment Schemes, neither of which been released in a current draft form yet; so a complete assessment of the implications of this document cannot be made.

3. Section 4.1: It is probably worth noting in the numbered list of uses for hash functions that they can also be used for producing small “authenticator” values for large cryptographic keys that must be manually verified upon installation. This is often an important operation that bootstraps trust in an infrastructure.

4. Section 4.2.2: Is not SKIPJACK also an Approved symmetric encryption algorithm? As it was on previous NIST lists, it should be mentioned explicitly if it is appropriate or not.

5. Section 4.2.5: Should not ANSI X9.44 be included in the fourth paragraph as an ANSI standard from which schemes will be adopted? It’s true that X9.44 hasn’t been approved yet, but it’s probably worth indicating that the intention is to adopt schemes from it. Note that X9.44 is mentioned in Section 4.2.5.3.

6. Section 4.2.6: In the first point in the numbered list, advice is given to not provide for an early exit from a protocol due to a single error. This is good advice, but perhaps too restrictive. Early exit from a protocol should be safe provided that the error is also recognizable to a third party. For example, if one party is not encoding its PDUs correctly and thus the other party cannot parse them, then an attacker would be able to notice this. Thus, we recommend that the phrase “… that is not recognizable to a third party,” be added.

7. Section 5.3.6: In point 2) there is a discussion of what to do if the time stamping hash algorithm becomes obsolete. This is not the only concern. There are similar concerns, for example, if the time stamping signature algorithm becomes obsolete. Perhaps it is best to just say “If the time stamping algorithm becomes obsolete ….
In point 3) there is a discussion of cryptoperiods that considers only temporal restrictions on cryptoperiods. Since some techniques have limits upon the number of messages that can be protected with a single key, it is probably worth mentioning

these limits as well. In fact, for many techniques basing cryptoperiods upon the number of messages is the superior method. Thus, limits of this type should be explicitly allowed.

In Sections 10) and 13) there are statements to the effect that “In certain email applications where received messages are stored and decrypted at a later time” the private decryption key may be used for longer periods. This is good guidance. However, the need isn’t limited to just email applications. For example, encrypted files on desktop computers or in external storage may not get decrypted until well after the encryption public key’s cryptoperiod expires. Thus, we recommend the removal of the word “email” in this context.
Additionally, these points go on to say that the private key shall not be used to decipher any new key transports or key establishments after the public key’s cryptoperiod has expired. This requirement cannot be enforced since the decrypting entity has no way to determine when the data was encrypted. All that they know for sure is that it was before now. Since this statement doesn’t provide any additional security (the data was already been encrypted using the expired key and this can’t be undone) this requirement seems to only restrict access to potentially crucial data. We recommend that this requirement be removed.

8. Section 5.4.1: It is probably worth noting that assurance of integrity can be obtained through inclusion of the public key in a valid X.509 certificate.

9. Section 5.4.2: Since RSA doesn’t have domain parameters it should be mentioned here that methods of obtaining assurance of domain parameter validity do not apply to it.

10. Section 5.5: In point 1) it is noted that in the event of unauthorized disclosure of a key, encrypted information could contain false information. However, this is always the case, even if the key is not compromised, for encryption techniques that do not include an integrity mechanism. More properly, assurance of correctness of data should NOT be assumed by any confidentiality mechanism, rather the assurance is strictly that of confidentiality. Therefore the “false information” sentence should be removed.

11. Section 5.6.1: In the third last line of the first paragraph is looks like the word “security” is missing. It should be “… that provides X bits of security would …”.

12. Table 2: We notice that the IFC key sizes in this table jump from 3072 bits to 7680 bits. This is a rather large gap and ignores the likelihood that some implementations will likely need to implement intermediate values for performance or other reasons. We suggest that an additional line be added to this table representing the IFC key size 4096 bits. We estimate the number of bits of security provided to be approximately 150. Clearly not all algorithms have options to allow usage at this security level (e.g., there is no symmetric cipher at this level), but including it in the table would provide guidance on security strength for implementations that need to use such an

intermediate value.

13. Section 6.1: The reference on the last line of the first paragraph should be to Section 5.6.4.

14. Section 6.1.1: In point c) of the numbered list, “availability” is mentioned as one of the types of protection. However, it is not included in the table at all for any key types. Thus, it should either be removed from this point, or included in the table where required.

15. Sections 6.2.1.5 and 6.2.2.5: It should be mentioned that valid X.509 certificates may be used to associate keying material with other entities (i.e., their owners).

16. Section 7.2: In transition 6 it states that a key may transition to the deactivated state if it is no longer to be used to protect data or to process protected data. However, we read earlier in this document that keys in the deactivated state may be allowed to process protected data, where required to access previously encrypted data. Thus the phrase “or to process cryptographically protected data,” should be removed.

17. Section 8: Transitions 3 and 5 refer to transitions to the “deactivated phase”. However, there does not appear to be a “deactivated phase”. This should be changed to “post-operational phase”.

18. Section 8.1: It is probably worth noting that the operations in Sections 8.1.1, 8.1.2 and 8.1.3 are often done before key generation, which is defined to be the start of the pre-operational phase.

19. Section 8.1.5: There is a requirement that all keys generated for key establishment purposes be generated within a FIPS 140-2 cryptomodule. The restriction to only these types of keys does not seem warranted. All keys should be generated in such a module. Thus, the phrase “for key establishment purposes” should be removed.

20. Section 8.1.5.1.1.2: This section restricts POP for key establishment keys by requiring that it not be provided by generating and verifying a digital signature. This is a good practice, but not only for this combination of keys and operations. More generally, “POP shall not be afforded by performing an operation not consistent with the key’s usage.”

The ANSI X9 standards are using the term “Assurance of possession” rather than POP, as what is provided is evidence, not a 100% proof. Since that terminology is already used earlier in this document it should probably be used here also.
In the last paragraph, it is noted that the CA shall perform any validation or other checks required. The RA will oftentimes perform some of the validations or other checks required by the CA and vouch for their validity. Thus, the sentence should be changed to add the phrase “… that have not already been performed by the RA.”

21. Section 8.1.5.1.1.3: In 2), the word “requirements” should be capitalized.

22. Section 8.2.2.1: Referring to the second paragraph, the keying material should also remain in backup for at least as long as it is maintained in storage for the postoperational phase.

23. Section 8.2.4: The restriction that keys derived from passwords not be used for encryption seems unwarranted, given that they can be used for authentication, which can be more important in some solutions. Please provide an explanation.

24. Section 10.2.1: The title and first sentence indicate that this section will also discuss the “Communications Environment”. However, the concept does not appear again. Thus, the phrase “and Communications Environment” should be removed from the title and first sentence.

25. Appendix A: This section still refers to collision attacks on SHA-1 requiring 280 operations. This should be updated in light of recent research.

26. Section B.5: It should be mentioned that an organization’s Key Recovery Policy can be included within its PKI Certificate Policy.

Robert Zuccherato
Chief Cryptographer
Phone: (613) 270-2598

Entrust
Securing Digital Identities
& Information
http://www.entrust.com

Date: Fri, 3 Jun 2005 12:31:23 -0400
From: "Randall, James" <jrandall@rsasecurity.com>

Comments on NIST Special Publication 800-57 part 1

June 3, 2005

From: James Randall (RSA Security Inc.)

First, we would like to commend NIST on the in-depth treatment of the subject material as well as on the quality and completeness of this document.

Following are RSAs comments on the document:

1. Table 1: Recommended Cryptoperiods for key types.

   We would like to suggest that the algorithm for determining cryptoperiods be based on more than the calendar. Keys protecting high value transactions or high volumes of sensitive data may need to be changed more frequently while keys infrequently used or used in low volume transactions could have an extended life.

2. Section 5.4.3 Assurance of public key validity.

   Add, after the last sentence, a description of owner assertion and/or owner self-assurances as additional methods of obtaining assurance of public key validity. (See ANS X9.44 Draft Section 12.5.1 for examples of owner assurances of key pair validity).

3. Table 3: Hash function security strengths for cryptographic applications.

   SHA-256 should be removed from the digital signature and hash-only application column for 192 bits of security and similarly, that SHA-256 and 384 be removed for 256 bits of security.

   See also the attached note and table as possible inclusions in a SHA-1 explanation.

4. Note b. following Table 4.

A caveat should be included explaining the SHA-1 restrictions. The forthcoming IETF RFC 4055 has some text on this topic and draft-hoffman-hash-attacks-03 has a more complete treatment.

5. Section 5.6.3 item c.

   Add the following after the last sentence: A 3072 bit RSA key could also be used to provide key establishment with 128 bits of security.

James Randall
RSA LABORATORIES
(781)515-6548
jrandall@rsasecurity.com

HASH MATRIX1.doc

SHA1 Update.doc