Comments Received on SP 800-175B

Comments Received #1

From: Lee Wilson <lwilson@securityinnovation.com>
Date: Monday, April 18, 2016 at 11:42 AM

This note is providing comment on SP 800-175B, *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*.

We'd like to add the following section (we've numbered it 5.3.7) to the document to cover hybrid key agreement. This "hybrid" method is in use today for making current SSL, TLS, etc. sessions quantum-safe and is covered by IETF internet drafts. By combining "classic" and post-quantum cryptography together in key generation/negotiation current key derivations using "classic" asymmetric cryptography can be made safe against future quantum attacks while retaining FIPS compliance.

Attached file:

5.3.7 Hybrid key agreement

Hybrid key agreement is an enhanced key-establishment procedure in which the resultant keying material is a combination of a key agreement and a key transportation. It provides a method that allows for the combination of a trusted, widely deployed classic key agreement protocol with a new cryptographic algorithm (e.g. post-quantum asymmetric cryptographic algorithm). For many, this can be a powerful transitional method helping them to migrate from current classic cryptography to post-quantum cryptography. By combining the strength of two approaches, the final key inherits both the strength of classical key agreements, as well as the new (quantum-safe) feature offered by the new cryptographic algorithm.

Such algorithms often are not FIPS-140 compliant. The hybrid method remains FIPS-140 compliant, as long as the key agreement part is FIPS-140 compliant. FIPS-140 allows for an extra data field during the key agreement procedure, where the new (quantum-safe) key transport data can be put.

The hybrid approach includes the following steps (Note: steps 1 and 2 are interchangeable or can be done in parallel):

1. Negotiation of classic keying material
    a. The initiator obtains the responder’s public key;
    b. The initiator generates a short-term key pair while retaining the ephemeral private key. The ephemeral public key is the initiator’s contribution to the key agreement process.
    c. Both parties use their own key pair and the other party's public key to generate a shared secret, which will be one of the inputs to the key derivation function where the symmetric key will be generated.

2. Negotiation of the new (quantum-safe) keying material
    a. A sender generates a short-term key pair retaining the ephemeral private key and sends the public key to the responder. (Note that the sender could have been either the initiator or the responder in the key agreement part of the transaction)
    b. The receiver generates (or otherwise obtains) a symmetric key to be transported. The symmetric key is wrapped with the ephemeral public key of the sender. The receiver then sends the resulting ciphertext to the sender.
    c. The sender unwraps the ciphertext using the ephemeral private key to obtain the symmetric key, which is used as the second part of the input to the key derivation function.
3. The final symmetric key is derived from both sets of keying material derived in steps 1 and 2 through a key derivation function.

Key confirmation can be performed as an optional step. It is highly recommended to provide assurance that both parties now have the same symmetric key.

NIST: While this approach is interesting, it is out-of-scope for SP 800-175B, which is intended to discuss the algorithms and techniques that are currently published in FIPS, SPs and other NIST publications.

Comments Received #2

From: "Flaherty, Colleen M. (CDC/OCOO/OCIO)" <cqr3@cdc.gov>
Date: Wednesday, April 27, 2016 at 10:41 AM

CDC has no comments to provide on the *Draft Special Publication 800-175B, Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*.

Thank you for the opportunity to review and comment.

Comments Received #3

From: NSA

General Comments: Overall the document does a nice job of giving a high level overview of certain aspects of both Symmetric and Public Key cryptography. It provides *MANY* references to other documents with more detailed descriptions of the ideas discussed which I think is extremely useful.

Most of my comments are editorial. I found nothing particularly wrong in the document, so it's really up to the folks at NIST what they want to take from this. My "issues" were mostly with some of the terminology used in the document which I felt was non-standard and should maybe be replaced.

As far as other stuff goes, my biggest suggestion would be to clarify the explanation of signature verification on page 33. My comments on Section 6.1 are just some suggestions of other things that *might* be worth mentioning in the discussion of *Required Security*.

Page 2 Line 65/66: I would not describe an *algorithm* as a *cryptographic methodology*. I've never heard that used as a definition. NIST: Revised

Page 5: Definition of *Digital Signature*: I prefer thinking of encryption and decryption as "transformations" and digital signatures as "computations". This, in fact, is the vocabulary used on page 4 in the definition of Cryptographic key. So how about defining a Digital signature as "A cryptographic computation performed on a set of data that, when properly implemented," etc… NIST: The definition was taken from SP 800-57, Part 1. No change.

Page 5: Definition of *ECDSA*. Just say "using elliptic curves" Drop the word "mathematics" NIST: Done

Page 5: Definition of *Ephemeral Key Pair*. Having just read on page 3 that a Certificate "contains the entity's public key", the statement "the public key is not certified" was confusing at first. Perhaps this statement should not be included in the definition of Ephemeral key. If it is included, clarification might help. Perhaps something like "Ephemeral public keys are not certified, unlike static public keys which often are." NIST: Done

Page 7: Definition of *Mode of Operation.* I think I would say *block cipher* as opposed to *lower-level algorithm.* NIST: Done

Page 7: Definition of *Plaintext.* I would refer to plaintext simply as unencrypted data or data that has not been encrypted. NIST: Done

Page 7: Definition of *Private Key* at #4. Replace "a piece of common shared data" by "shared secret". I would do the same in the definition of *Public Key* right below. NIST: Done

Page 8: Definition *Secret Key.* I'm not sure why this needs to be defined. I think it will cause more confusion than anything (e.g. in asymmetric cryptography the private key must be kept secret too.) It would be more appropriate to say define a *Symmetric Key* as

one that is utilized for encryption and decryption in a symmetric key algorithm and note that it must be kept secret (which is sort of done later on down) NIST: Additional text inserted.

Several times in the document I have seen *Secret Key Algorithm* used in place of, or in conjunction with *Symmetric Key Algorithm*. I have never seen nor heard this expression used. NIST: Both terms have been used in the NIST documents, but the latter term (Symmetric Key Algorithm) has been primarily used in SP 800-175B, rather than Secret Key Algorithm.

Page 8: Definition of *Security Strength.* Again this is a non-standard definition which I've never seen. Better to say perhaps *Security of an Algorithm.* I would say that security is measured in terms of bit operations rather than just bits. Also, consider dropping the 80-bit strength from the list. Then you could drop the sentence where you say it's not allowed. NIST: The definition was used in SP 800-57, Part 1. However, the definition in SP 800-175B has been shortened, with the list of strengths and a discussion of the 80-bit strength being moved to Section 3.4.

Page 8: Definition of *Shared secret*: There is a typo: change "as input to a derive a key" to "as input to derive a key" NIST: Done

Page 9: Definition of *Symmetric Key*. See comment above from page 8. NIST: The definition was amended.

Page 10, line 130: It would be useful to make it clear that RSA refers to an algorithm. NIST: Done

Page 13, lines 193-194: This comment is picky. It is the **use** of a standard that can save money, not the standard itself or its mere existence. Perhaps something like "Adherence to the commonly accepted specifications provided by standards can save money." NIST: Done

Page 13, line 194: There appears to be an extra space after the sentence ending with "specification." NIST: Deleted

Page 13, lines 201-203: This sentence seems awkward and confusing to me. Perhaps "if a Federal Information Processing Standard (FIPS) contains specifications for a service required by a federal agency to protect sensitive information, then its use by the Federal Government is mandatory." NIST: Reworded

Page 13, line 215: There is a typo: change "(e.g. for encryption)" to "(e.g. encryption)". NIST: Done

Page 18, line 386: Another typo: change "in a particular classes" to "in a particular class". NIST: Done

Page 20, lines 439-441: This sentence seems awkward to me. While attacks on SHA-1 indicate that it provides less security than originally thought, the attacks don't indicate that SHA-1 is now disallowed (as the sentence seems to state). Perhaps change it to "Note that the attacks against SHA-1 have shown that it provides less security than originally thought when used to generate digital signatures. Consequently, SHA-1 is now disallowed for that purpose." NIST: Done

Page 20, Line 448: Drop reference to *secret-key algorithms*. NIST: The term is used only once in the document. See a previous comment for more information.

Page 20, Line 449: Might just say, "Symmetric key algorithms use a single key to encrypt and decrypt data." NIST: Symmetric keys are used not only for encryption and decryption, but also for authentication (e.g., CMAC and HMAC). Symmetric keys are also used in authenticated encryption modes, such as CCM and GCM.

Page 23, Line 528: I would not even get into the issue of the speed of AES with different CV sizes. I don't think it's really relevant to this document and (as noted) it's not necessarily true that one block cipher with a 256-bit CV is automatically slower than another utilizing a 64-bit CV. NIST: The sentence was deleted.

Page 23, lines 532-537: To me, this paragraph would make more sense if the sentences were reordered. The observation that repeated blocks of plaintext are apparent in the ciphertext seems to be a more direct consequence of the first sentence than the observation that the ciphertext might be altered without detection. How about "With a symmetric key block cipher algorithm, the same input block always produces the same output block when the same key is used. So if such an algorithm were used for encryption without an appropriate mode of operation, then certain data patterns in the plaintext, such as repeated blocks, would be apparent in the ciphertext. Furthermore, without a mode to provide cohesion between the blocks, an adversary could substitute individual blocks of his own choosing, perhaps without detection."? NIST: Reworded

Page 23, Line 538: Change "this problem" to "these problems". NIST: No longer applicable/used

Page 24, Line 560: Replace "easily" with "efficiently" NIST: Done

Page 24, lines 576-578: This is picky, but key pairs are not generated by symmetric-key algorithms. Thus symmetric and asymmetric algorithms are "unlike" not "like" in this regard. Perhaps "As previously noted for symmetric algorithms, asymmetric algorithms should not use the same keys for different purposes. For example, a key pair used to generate and verify digital signatures should be distinct from one used for key establishment."? NIST: Reworded

Page 25, Line 617: Just say "…using finite fields." Drop the word "mathematics" and the comment in the parentheses. NIST: Done

Page 25, Line 626: As above, drop the comment in parentheses. It's just confusing. The distinction between arithmetic in a finite field versus that on an elliptic curve is not important for a document at this level. NIST: Done

Page 26, Line 654: After the footnote (41), Replace with "…using finite field or elliptic curves…" NIST: Done

Page 26, Line 661: Again, people either talk about the "security of an algorithm" or the "strength of an algorithm." The term *security strength* is one I have never heard. NIST: This concept is used in Section 5.6 of SP 800-57, Part 1

Page 27, line 673: Change "is further discussed" to "are further discussed". NIST: Done

Page 29, lines 738-739: I don't think the parenthetical comment (e.g. some data) is necessary. The fact that data integrity pertains to data seems to be a tautology rather than an example. Perhaps "Data integrity (often simply referred to as integrity) is concerned with whether nor not a data set has changed between two specified times…" NIST: Reworded

Page 29, lines 741-744: This sentence seems long and awkward to me. Also, a change in the data is detected not when the data integrity code is verified (i.e. the prior and post data codes agree) but rather when the data integrity code cannot be verified (i.e. the prior and post data integrity codes differ). Perhaps "While data integrity cannot be guaranteed, the use of data integrity codes provides a means to detect changes with high probability. A data integrity code is computed on a data set when it is created and, once again, when the data is either received or retrieved from storage. Verification that these two computations agree provides a measure of assurance of data integrity." NIST: Reworded

Page 30, Line 763-764: This is a little bit misleading because when a hash is used in conjunction with digital signatures; the method works fine (as is discussed later). I would clarify this just a little. NIST: Done

Page 31, lines 789-795: I don't think the word "party" needs to be repeated so much, nor everything repeated for the second case. How about "For example, suppose two parties (A and B) share a key. If A generates the MAC and sends it to B, and B successfully verifies the received MAC, then B knows that A generated the original MAC, and source authentication has been accomplished. However, if the above situation is modified so that a third party, C, shares the key, then B knows that either A or C generated the original MAC, but B cannot determine which. Note that…." NIST: Removed one recurrence

Page 33, line 873: "D_2" should be "DS_2" and "D_1" should be "DS_1" (two occurrences). NIST: Done

Page 33: I find the diagram describing signature verification confusing or at least not illuminating. NIST: We disagree. Nothing changed.

I think the written description of the signature verification process could be more clear as well. For example in Bruce Schneier's *Applied Cryptography (2nd edition)*, p. 38, he gives a very simple 4-line description of the basic signature/verification process which I think the reader would find more helpful. Worth taking a look. NIST: We disagree. Nothing changed.

Page 41, Lines 1139-1140: Is "distribute" the right word here? NIST: Changed to "provide"

Page 43, Line 1202: Probably want to replace "and" with "or". NIST: Done

Page 55, Line 1563: Again, I would not say "security strength" NIST: This is the term we use in our documents.

There's a lot that might be added to this section (6.1) to help clarify the concept of "Required Security".

First, is the idea that ***all*** the algorithms one utilizes need to have the appropriate level of security. It might sound obvious but I can't tell you how many times I've seen people use (for example) AES-256 for encryption along with 1024-bit RSA for digital signatures.

Just because you are using AES-256 does not mean your IPSec or TLS session has 256-bits of security! You are only as strong as the weakest algorithm in your cipher-suite and that is something people have a terrible time understanding (the fact that 1024 is bigger than 256 means nothing in this context.) NIST: Text was added to discuss this issue.

Second, It might actually be worth including some examples of cipher suites that as a whole provide a given bit-level of security (112-bit, 128-bit etc..). The cipher suites specified by NSA Suite B IPSec for various bit levels of security might be useful. For example (if I remember correctly),

128-bit level

Encryption: AES-128

Key Agreement (ECDH): Elliptic Curve p-256 (Group 19)

Hashing: SHA2-256

Signatures: 256-bit ECDSA

192-bit Level

Encryption: AES-256

Key Agreement (ECDH): Elliptic Curve p-384 (Group 20)

Hashing: SHA2-384

Signatures: 384-bit ECDSA

NIST: A generic example was included in the new text, and references to SP 800-57, Part 3 and SP-800-152 have been included.