Public Comments on Draft SP 800-38F

On August 12, 2011, NIST announced a period of public comment, ending October 1, 2011, on draft Special Publication 800-38F: *Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping*. The announcement was posted on the News and Events page at NIST's Computer Security Resource Center.

Below are the public comments that NIST received.

Marc Boire

As an FYI, page 1 (page 9 of pdf file) of the draft appears to have the incorrect header, labeling it as version SP 800-38E.

Regards,

Marc Boire
CGI IT Security Evaluation & Test Facility
Security & Privacy Services
CGI Information Systems & Management Consultants Inc.

Brian Weis and David McGrew

September 30, 2011
To: Morris Dworkin
Re: Comments on NIST SP 800-38F DRAFT – August 2011

Dear Mr. Dworkin,

There has been a need for a NIST Recommendation describing approved methods of key wrapping for several years. This document will provide clarity to the standards community as well as a stable reference.

We also laud the statements in Clause 3.1 explicitly stating that other NIST approved authenticated encryption modes (and combinations of NIST approved encryption and authentication modes) are approved for the protection of cryptographic keys. While the KW, KWP, and TWK methods are applicable to many standards needing to explicitly pass a key as part of a network message, the ability for network protocols to use other approved methods is also valuable. In particular, group key management protocols securely distribute keys using other NIST approved modes where the use of KW methods would be inefficient.

We would like to point out one area where additional clarification might be warranted. The authentication strength is discussed in Appendix 3, but it is not readily apparent whether or not AES-KW can be used in the absence of a separate message authentication code. We expect that the intent of the specification is to allow AES-KW to be used without additional authentication. In any case, it seems worthwhile to include a statement on when KW can be used without a separate MAC or digital signature.

We believe that the approval of this DRAFT will result in a valuable addition to the NIST 800-38 series.

Sincerely,

Brian Weis

David McGrew

Todd Arnold

I looked through the draft SP 800-38F and I had a few comments, mainly minor.

Page 9, bottom of the page - The paragraph beginning "For each key wrap variant..." says the unwrap function is defined on sequences of three or more semiblocks. The paragraph following that one says KW-AD and TKW-AE are defined on any sequence of two or more semiblocks. Is this correct, or is one of these an error?

Page 22, comparison with earlier specifications - (1) The last paragraph needs to also say something about whether this is equivalent to [2], the ANSI X9.102 standard. It does not mention that standard - only the other earlier ones. (2) I think it would be very useful to have a statement saying that if you implemented something according to one of the earlier standards (NIST AESKW document, X9.102, RFC) that your implementation is compliant with this new SP 800-38F, as long as you meet a set of clearly described conditions. For example, we have products in the field using AESKW according to X9.102 - I would really like to be able to point to something in SP 800-38F that says we are compliant with it as well.

General - I think it badly needs both examples and test vectors. Each section should have examples showing the steps described, using tangible example data. An appendix should have a set of test vectors that can be used to verify an implementation has been done correctly.

- Todd

Dan Harkins

Hello,

I know these comments are 2 weeks late but hopefully you will still consider them.

You refer to Rogaway's and Shrimpton's paper "Deterministic Authenticated Encryption" only to say that one of their criticisms on the AES Key Wrap schemes is their lack of a proof. In that paper, the authors defined another key wrapping scheme (SIV), one that is proven secure. In addition, as they note, SIV is more efficient (when compared to AESKW "the number of blockcipher calls is reduced by a factor of at least six") and it also accepts a vector of inputs which are, effectively, authenticated but not encrypted.

This last feature is very powerful for systems that require key wrapping.
It allows the message containing the wrapped key to be bound to the wrapped key itself. It also allows other data which might not be part of the message or part of the wrapped key to be similarly bound into the key.
The inability to reproduce this data, or any tampering with the message will result in a key that is not unwrappable.

While a separate HMAC on the message containing the wrapped key can approximate the security service provided by SIV, it does not quite match it because a key can still be successfully unwrapped from a message that was tampered with. Also, separating HMAC and KW opens up the possibility of them being used incorrectly-- authenticate then encrypt or authenticate and encrypt or authenticate then encrypt-- by implementers who might lack the security wherewithal to use these tools properly. SIV provides this service in an integrated fashion and achieves a level of misuse-resistance that AESKW lacks as a result.

SIV has been specified for use in 802.11, to send a wrapped broadcast key to a peer in a mesh network. SIV is powerful and attractive. It will probably be proposed in more specifications in the future.

NIST should include the specification for SIV in SP800-38F and accept its use for conformance to the SP. To not do so would be to tip the scale (in a negative way) when an organization or standards body considers the use of SIV. It would also unfairly prevent implementations that use a provably secure, misuse-resistant key wrapping scheme from large markets.

Thank you for your consideration of these comments. Regards,

Dan Harkins.

Rafal Gliwa

Dear Sir / Madame,

Below are my comments regarding Draft NIST Special Publication 800-38F "Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping".

1.

In Abstract we read:

"This publication describes cryptographic methods that are approved for "key wrapping," i.e., the protection of the confidentiality and **integrity** of cryptographic keys. In addition to describing existing methods, this publication specifies two new, deterministic **authenticated encryption** modes of operation (...)."

Furthermore, in 3.1 Overview we read:

"This Recommendation specifies a deterministic **authenticated encryption** mode of operation of the Advanced Encryption Standard (AES) algorithm.

But, from 4.1. Definitions we get:

"authenticated encryption function - A function that encrypts plaintext into ciphertext and provides a means for the associated authenticated decryption function to verify **the authenticity** of either the plaintext or the ciphertext."

It is not clear if KW, KWP and TKW provide both **confidentiality and authenticity** (authenticity includes integrity) or rather **confidentiality and integrity** (and **no** authenticity) of cryptographic keys.

2.

In 4.1. Definitions we read:

"block cipher mode of operation (mode) - A function, or a pair of related functions, e.g., for encryption, authentication, or authenticated encryption, of which a block cipher is the main component."

a) The definition is different from “Mode of Operation (Mode)” definition in NIST Special Publication 800-38D. Defining a mode as an algorithm was, in my opinion, more appropriate than as “a function, or a pair of related functions”.

b) The most comprehensible definition of block cipher mode of operation gives, in my opinion, Bart Preneel in ENCYCLOPEDIA OF CRYPTOGRAPHY AND SECURITY (Editor-in-chief Henk C.A. van Tilborg, Springer Science+Business Media, Inc., 2005):
“A mode of operation of a block cipher is an algorithm which specifies how one has to apply an *n*-bit block cipher to achieve this.” i.e. to achieve the security goal e.g. confidentiality, authenticity or both simultaneously.

3.
In 4.1. Definitions we read:
“integrity check value - A fixed string that is prepended to the plaintext within the authenticated encryption function of a key wrap algorithm, in order to be verified within the authenticated decryption function.”

After words “in order to” I would point at the security goal i.e.
“(...) in order to enable plaintext integrity verification within the authenticated decryption function.

4.
In 4.1. Definitions we read:
“unwrapping function - The inverse of the wrapping function.”

It would be better to give the definition directly, because trying to define it on one’s own basing on “wrapping function” definition is not so clear:
“wrapping function - The keyed, length-preserving permutation that is applied to an enlarged form (**unwrapping function is applied just to the ciphertext and not to its enlarged form**) of the plaintext within the authenticated encryption function to produce the ciphertext.

5.

In 4.5 Examples of Basic Operations and Functions on Strings we read:

"Given a positive integer *s*, 0^s denotes the string that consists of *s* '0' bits. For example, 0^8 = 00000000"

The definition is imprecise, because of lack of word "consecutive" i.e. "denotes the string that consists of *s* consecutive '0' bits.

6.

In Algorithm 1, Algorithm 2 we have: **key, K**.

In Algorithm 3, Algorithm 4, Algorithm 5, Algorithm 6 we have: **key encryption key, K**.

In Algorithm 7, Algorithm 8 we have: **key, K**.

In Algorithm 9 we have: **KEK, K.**

In Algorithm 10 we have: **key, K**.

Because all these Algorithms use only one key, which is key encryption key (KEK) and is denoted K (according to 5.1) I would suggest using the name **key encryption key K** everywhere**.**

7.

The sentence before Algorithm 8 in section 7.1 ends with two dots.

8.

Output in Algorithm 4, output in Algorithm 6 and output in Algorithm 10 is:

"Plaintext *P* or indication of inauthenticity, *FAIL*.",

To be precise, it would be good to assign P to the return value of the last step (e.g. $P=LSB_{64(n-1)}(S)$ in Algorithm 4).

Yours faithfully,

Rafal Gliwa

Rene Struik

Reference: NIST SP 800-38F – Recommendations for Block Cipher Modes of Operation, Draft, August 2011

Review notes:

1. Summary

The draft publication specifies deterministic mechanisms for wrapping keys, both for NIST-approved block ciphers with 128-bit block size (such as AES) and for 3DES, a block cipher with 64-bit block length. For 128-bit block ciphers, a wrapping mechanism with padding is described as well. The mechanisms all result in data expansion of roughly half the block size of the underlying cipher, thereby providing a mechanism to check data authenticity of unwrapped keys. The specified mechanisms are claimed to be consistent with key wrap mechanisms specified with, e.g., IETF and ANSI X9F.

2. Review comments

2.1 General comments

The draft specification has some limitations that may hamper wide-spread deployment, including

- *No support for binding keys to other keying information*. While the key wrap mechanisms provide for (some) authenticity of the symmetric-key being wrapped, this does not extend to other (publicly known) keying information that is usually associated with this key (such as, e.g., key validity period, key originator, key identifier, or key usage policy field). Thus, if one wishes to bind this other keying information, one has to resort to another as yet unspecified mechanism on top of the key wrap mechanism, thereby increasing implementation cost and data expansion. It may be highly beneficial to extend the key wrap mechanisms specified with such a more flexible mechanism, e.g., by considering the SIV construct described in reference [9] of the draft specification or another construct with similar extended security services.
- *Lack of support in existing implementations*. The mechanisms described seem to assume implementation of the "forward cipher mode" of the underlying block cipher (for key wrap) and "inverse cipher mode" hereof (for key unwrap). In practical applications, it may be useful to reverse this enciphering/deciphering functionality (thereby, effectively swapping the block cipher and its inverse). As case in point, many wireless devices have hardware support for the AES-128 block cipher in "forward cipher" mode only. By reversing enciphering/deciphering functionality, these wireless devices could implement the key unwrap functionality with existing hardware support, while only requiring a key distribution center (that wraps keys, e.g., as inline KDC) to implement the corresponding inverse cipher mode. Considering that wireless devices may far outnumber KDCs and may not have a trusted computing base on board, this seems to be the preferred mode going forward (and would remove a practical hurdle in deploying the specified mechanisms with, e.g., WiFi devices and a plethora of "internet of things" style devices). Thus, I would recommend explicitly allowing for this "reversed" capability in the specification, so that others can reap the benefits hereof.

2.2 Technical comments

The draft specification has some inaccuracies, including the following:

- §4.1, pp. 4-6: I would recommend introducing the terms "forward cipher" and "inverse cipher", which are used in the sequel of the document (e.g., §4.4, p. 6 – description of $CIPH_k(X)$ and $CIPH^{-1}_k(X)$; §7.1, p. 16, just before Alg. 7 and Alg. 8). Here, the definition should accommodate that one could swap block cipher and its inverse (so as to accommodate existing implementations – cf. General comments above) or one should make it otherwise clear that if a particular block cipher is approved, then so is its inverse. (The latter could be realized by introducing "approved cipher – FIPS-approved or NIST-recommended block cipher or its inverse" or similar).
- §6.1, p. 11, Alg. 1 – Prerequisites: make clear that CIPH can be an approved block-cipher or its inverse. A similar remark applies to $CIPH^{-1}$ (Alg. 2, p. 12), TDEA (Alg. 7, p. 16, resp. Alg. 8, p. 16).
- §6.2, p. 13, l. -1 – Prerequisites: the valid plaintext lengths should be an integral number *n* of semiblocks, where $n \geq 1$ (otherwise, the wrapping function has inputs of wrong types). Note that *n* could be equal to 1 (for consistency, since the padded wrap function KWP-AE (Alg. 5) allows as plaintext a single semiblock). This could be curtailed further, as suggested in the draft, of course.
- §6.2, p. 14: The first step should be to check whether the plaintext has a valid length (i.e., bit length consistent with number of semiblocks, potentially further constraints) and result in abortion of the protocol if invalid; similar to what is done with, e.g., Alg. 4, Step 1. A similar remark applies to a missing plaintext length check in Alg. 9 (p. 17) and incomplete language in Step 1 of Alg. 10 (p. 18) {there, one only learns in Step 2 that the ciphertext size is a multiple of the semiblock size, where this number is at least 2 (I presume – again, assuming NIST's desire for consistency here).
- §6.3, p. 15, Alg. 5 – Input: one should mention that the plaintext *P* has bit-length 0 (mod 8), since otherwise the formulae are incorrect.
- §6.3, p. 15, Alg. 5 – Step 5: It is unclear why one could not invoke the wrapping function W(S) if the plaintext *P* consists of a single (potentially incomplete) semiblock only. After all, in that case the string S has as bit-size the block-size of the underlying cipher. Or, is this an artifact explained by history of the key wrap functions, as introduced over the last decade?
- §7.1, p. 16, l. 3: (as reminder – already mentioned elsewhere) the "forward cipher" function with AES and TDEA means something very specific and precludes swapping the block cipher and its inverse. Similar remark (again, just a reminder) for "inverse cipher" mention just before Alg. 8.
- §A.4, p. 21: The approximation of the success probability *E* of forgery is somewhat unclear. Moreover, it first assumes *m* to be significantly smaller than 2^{64} (as ratio) to arrive an approximation for this value, while subsequently taking *m* to be not that much smaller than this number (a factor $2^{10}\times$ only) to arrive at a numerical bound. This makes one wonder whether the error term of the approximation may impact the accuracy of the derived numerical value. Some more explanation may definitely help the reader.

2.3 Editorial comments

- §4.5, p. 7, 6th para: Replace "in each bit position" by "in corresponding bit positions".
- §5.1, p. 8, 3rd para, l. 2: Replace "the AES algorithm" by "the AES block cipher" (AES is a set of algorithms, including the "forward cipher" and the "inverse cipher").

- §6.1, p. 11: (Alg. 1 – W(S)) The input constraint $n \geq 3$ can be safely replaced by $n \geq 2$ (this is an auxiliary function; no need to have constraints creeping in from the calling KW/KWP function at this point). A similar remark applies to Alg. 2 – $W^{-1}(S)$ (p. 12), Alg. 7 – TW(S) (p. 16), and Alg. 8 ($TW^{-1}(S)$ (p. 16).
- §6.1, p. 11, l. 2: Replace "block cipher" by "block cipher CIPH". Also with Alg. 2 and TDEA (Alg. 7 and Alg. 8).
- §6.1, p. 11, l. 2: The key encryption key is called "KEK" here, but "K" in the prerequisite underneath. Also with Alg. 2 and TDEA (Alg. 7 and Alg. 8).
- §6.1, p. 11, Step 1: Move Step 1d) to the top (i.e., set s=6(n-1) as Step 1a), etc.). This realizes "parallelism in style" between the steps in Alg. 1 and Alg. 2. A similar remark applies to the corresponding wrapping functions for TDEA (Alg. 7 and Alg. 8).
- §6.1, p. 13, Step 2 c): Replace "For i=3, …, n, $R_i^{t-1} = R_{i-1}^{t}$" by "For i=2, …, n-1, $R_{i+1}^{t-1} = R_i^{t}$". (Again, "parallelism in style" between corresponding steps in Alg. 1 and Alg. 2.) A similar remark applies to the corresponding wrapping function for TDEA (Alg. 7 and Alg. 8).
- §6.3, p. 14, l. 3: add "… as prerequisites" (so as to make this consistent with language used in §6.2, p. 13, l. 3). Also elsewhere.
- §7.2, p. 18, Step 2: Replace "n+1" by "n" (so as to make this consistent with language used in §6.3 – KWP-AD (Alg. 6)). Make corresponding change with Step 6 and replace this step by "Return $LSB_{32(n-1)}(S)$".
- §6.1, pp. 11-13 vs. §7.1, pp. 16-17: The description of the wrapping function for W and W^{-1}, resp. TW and TW^{-1}, could be unified in one single section: to this end, simply introduce a new variable b that denotes half the block-size of the underlying cipher and use this to define the LHS and RHS split of the output of the block-cipher in each round. Having unified this, one could simplify invoke the algorithm with AES-type and TDEA-type block-ciphers in the description of the key wrap function.
- §A.3, p. 19, l. 2: Replace "upon the execution" by "upon execution".
- §B, p. 22, 3rd para, l. 3: Replace "Center web" by "Center web pages".
- References, p. 23: Remove the *italics* font in "Version 1.1" with reference [5].