Network Working Group
Request for Comments: 3537
Category: Standards Track

J. Schaad
Soaring Hawk Consulting
R. Housley
Vigil Security

May 2003

Wrapping a Hashed Message Authentication Code (HMAC) key with a Triple-Data Encryption Standard (DES) Key or an Advanced Encryption Standard (AES) Key

Status of this Memo

Copyright Notice

Abstract

This document defines two methods for wrapping an HMAC (Hashed Message Authentication Code) key. The first method defined uses a Triple DES (Data Encryption Standard) key to encrypt the HMAC key. The second method defined uses an AES (Advanced Encryption Standard) key to encrypt the HMAC key. One place that such an algorithm is used is for the Authenticated Data type in CMS (Cryptographic Message Syntax).

1. Introduction

Standard methods exist for encrypting a Triple-DES (3DES) content-encryption key (CEK) with a 3DES key-encryption key (KEK) [3DES-WRAP], and for encrypting an AES CEK with an AES KEK [AES-WRAP]. Triple-DES key wrap imposes parity restrictions, and in both instances there are restrictions on the size of the key being wrapped that make the encryption of HMAC [HMAC] keying material difficult.

This document specifies a mechanism for the encryption of an HMAC key of arbitrary length by a 3DES KEK or an AES KEK.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [STDWORDS].

2. HMAC Key Guidelines

[HMAC] suggests that the key be at least as long as the output (L) of the hash function being used. When keys longer than the block size of the hash algorithm are used, they are hashed and the resulting hash value is used. Using keys much longer than L provides no security benefit, unless the random function used to create the key has low entropy output.

3. HMAC Key Wrapping and Unwrapping with Triple-DES

This section specifies the algorithms for wrapping and unwrapping an HMAC key with a 3DES KEK [3DES].

The 3DES wrapping of HMAC keys is based on the algorithm defined in Section 3 of [3DES-WRAP]. The major differences are due to the fact that an HMAC key is of variable length and the HMAC key has no particular parity.

In the algorithm description, "a || b" is used to represent 'a' concatenated with 'b'.

3.1 Wrapping an HMAC Key with a Triple-DES Key-Encryption Key

This algorithm encrypts an HMAC key with a 3DES KEK. The algorithm is:

1. Let the HMAC key be called KEY, and let the length of KEY in octets be called LENGTH. LENGTH is a single octet.

2. Let LKEY = LENGTH || KEY.

3. Let LKEYPAD = LKEY || PAD. If the length of LKEY is a multiple of 8, the PAD has a length of zero. If the length of LKEY is not a multiple of 8, then PAD contains the fewest number of random octets to make the length of LKEYPAD a multiple of 8.

4. Compute an 8 octet key checksum value on LKEYPAD as described in Section 2 of [3DES-WRAP], call the result ICV.

5. Let LKEYPADICV = LKEYPAD || ICV.

6. Generate 8 octets at random, call the result IV.

7. Encrypt LKEYPADICV in CBC mode using the 3DES KEK. Use the random value generated in the previous step as the initialization vector (IV). Call the ciphertext TEMP1.

8. Let TEMP2 = IV || TEMP1.

9. Reverse the order of the octets in TEMP2. That is, the most significant (first) octet is swapped with the least significant (last) octet, and so on. Call the result TEMP3.

10. Encrypt TEMP3 in CBC mode using the 3DES KEK. Use an initialization vector (IV) of 0x4adda22c79e82105.

Note: When the same HMAC key is wrapped in different 3DES KEKs, a fresh initialization vector (IV) must be generated for each invocation of the HMAC key wrap algorithm (step 6).

3.2 Unwrapping an HMAC Key with a Triple-DES Key-Encryption Key

This algorithm decrypts an HMAC key using a 3DES KEK. The algorithm is:

1. If the wrapped key is not a multiple of 8 octets, then error.

2. Decrypt the wrapped key in CBC mode using the 3DES KEK. Use an initialization vector (IV) of 0x4adda22c79e82105. Call the output TEMP3.

3. Reverse the order of the octets in TEMP3. That is, the most significant (first) octet is swapped with the least significant (last) octet, and so on. Call the result TEMP2.

4. Decompose the TEMP2 into IV and TEMP1. IV is the most significant (first) 8 octets, and TEMP1 is composed of the remaining octets.

5. Decrypt TEMP1 in CBC mode using the 3DES KEK. Use the IV value from the previous step as the initialization vector. Call the plaintext LKEYPADICV.

6. Decompose the LKEYPADICV into LKEYPAD, and ICV. ICV is the least significant (last) 8 octets. LKEYPAD is composed of the remaining octets.

7. Compute an 8 octet key checksum value on LKEYPAD as described in Section 2 of [3DES-WRAP]. If the computed key checksum value does not match the decrypted key checksum value, ICV, then error.

8. Decompose the LKEYPAD into LENGTH, KEY, and PAD. LENGTH is the most significant (first) octet. KEY is the following LENGTH of octets. PAD is the remaining octets, if any.

9. If the length of PAD is more than 7 octets, then error.

10. Use KEY as an HMAC key.

3.3 HMAC Key Wrap with Triple-DES Algorithm Identifier

Some security protocols employ ASN.1 [X.208-88, X.209-88], and these protocols employ algorithm identifiers to name cryptographic algorithms. To support these protocols, the HMAC Key Wrap with Triple-DES algorithm has been assigned the following algorithm identifier:

```
id-alg-HMACwith3DESwrap OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
    smime(16) alg(3) 11 }
```

The AlgorithmIdentifier parameter field MUST be NULL.

3.4 HMAC Key Wrap with Triple-DES Test Vector

```
KEK          :  5840df6e 29b02af1
             :  ab493b70 5bf16ea1
             :  ae8338f4 dcc176a8

HMAC_KEY     :  c37b7e64 92584340
             :  bed12207 80894115
             :  5068f738

IV           :  050d8c79 e0d56b75

PAD          :  38be62

ICV          :  1f363a31 cdaa9037

LKEYPADICV   :  14c37b7e 64925843
             :  40bed122 07808941
             :  155068f7 38be62fe
             :  1f363a31 cdaa9037

TEMP1        :  157a8210 f432836b
             :  a618b096 475c864b
             :  6612969c dfa445b1
             :  5646bd00 500b2cc1
```

```
TEMP3        :  c12c0b50 00bd4656
             :  b145a4df 9c961266
             :  4b865c47 96b018a6
             :  6b8332f4 10827a15
             :  756bd5e0 798c0d05

Wrapped Key  :  0f1d715d 75a0aaf6
             :  6f02e371 c08b79e2
             :  a1253dc4 3040136b
             :  dc161118 601f2863
             :  e2929b3b dd17697c
```

4. HMAC Key Wrapping and Unwrapping with AES

This section specifies the algorithms for wrapping and unwrapping an HMAC key with an AES KEK [AES-WRAP].

The AES wrapping of HMAC keys is based on the algorithm defined in [AES-WRAP]. The major difference is inclusion of padding due to the fact that the length of an HMAC key may not be a multiple of 64 bits.

In the algorithm description, "a || b" is used to represent 'a' concatenated with 'b'.

4.1 Wrapping an HMAC Key with an AES Key-Encryption Key

This algorithm encrypts an HMAC key with an AES KEK. The algorithm is:

1. Let the HMAC key be called KEY, and let the length of KEY in octets be called LENGTH. LENGTH is a single octet.

2. Let LKEY = LENGTH || KEY.

3. Let LKEYPAD = LKEY || PAD. If the length of LKEY is a multiple of 8, the PAD has a length of zero. If the length of LKEY is not a multiple of 8, then PAD contains the fewest number of random octets to make the length of LKEYPAD a multiple of 8.

4. Encrypt LKEYPAD using the AES key wrap algorithm specified in section 2.2.1 of [AES-WRAP], using the AES KEK as the encryption key. The result is 8 octets longer than LKEYPAD.

4.2 Unwrapping an HMAC Key with an AES Key

The AES key unwrap algorithm decrypts an HMAC key using an AES KEK. The AES key unwrap algorithm is:

1. If the wrapped key is not a multiple of 8 octets, then error.

2. Decrypt the wrapped key using the AES key unwrap algorithm specified in section 2.2.2 of [AES-WRAP], using the AES KEK as the decryption key. If the unwrap algorithm internal integrity check fails, then error, otherwise call the result LKEYPAD.

3. Decompose the LKEYPAD into LENGTH, KEY, and PAD. LENGTH is the most significant (first) octet. KEY is the following LENGTH of octets. PAD is the remaining octets, if any.

4. If the length of PAD is more than 7 octets, then error.

5. Use KEY as an HMAC key.

4.3 HMAC Key Wrap with AES Algorithm Identifier

Some security protocols employ ASN.1 [X.208-88, X.209-88], and these protocols employ algorithm identifiers to name cryptographic algorithms. To support these protocols, the HMAC Key Wrap with AES algorithm has been assigned the following algorithm identifier:

```
id-alg-HMACwithAESwrap OBJECT IDENTIFIER ::= { iso(1)
    member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9)
    smime(16) alg(3) 12 }
```

The AlgorithmIdentifier parameter field MUST be NULL.

4.4 HMAC Key Wrap with AES Test Vector

```
KEK          :  5840df6e 29b02af1
             :  ab493b70 5bf16ea1
             :  ae8338f4 dcc176a8

HMAC_KEY     :  c37b7e64 92584340
             :  bed12207 80894115
             :  5068f738

PAD          :  050d8c

LKEYPAD      :  14c37b7e 64925843
             :  40bed122 07808941
             :  155068f7 38050d8c
```

```
Wrapped Key  :  9fa0c146 5291ea6d
             :  b55360c6 cb95123c
             :  d47b38cc e84dd804
             :  fbcec5e3 75c3cb13
```

5. Security Considerations

Implementations must protect the key-encryption key (KEK). Compromise of the KEK may result in the disclosure of all HMAC keys that have been wrapped with the KEK, which may lead to loss of data integrity protection.

The use of these key wrap functions provide confidentiality and data integrity, but they do not necessarily provide data origination authentication. Anyone possessing the KEK can create a message that passes the integrity check. If data origination authentication is also desired, then the KEK distribution mechanism must provide data origin authentication of the KEK. Alternatively, a digital signature may be used.

Implementations must randomly generate initialization vectors (IVs) and padding. The generation of quality random numbers is difficult.

RFC 1750 [RANDOM] offers important guidance in this area, and Appendix 3 of FIPS Pub 186 [DSS] provides one quality PRNG technique.

The key wrap algorithms specified in this document have been reviewed for use with Triple-DES and AES, and have not been reviewed for use with other encryption algorithms.

6. References

6.1 Normative References

[3DES] American National Standards Institute. ANSI X9.52-1998, Triple Data Encryption Algorithm Modes of Operation. 1998.

[3DES-WRAP] Housley, R., "Triple-DES and RC2 Key Wrapping", RFC 3217, December 2001.

[AES] National Institute of Standards and Technology. FIPS Pub 197: Advanced Encryption Standard (AES). 26 November 2001.

[AES-WRAP] Schaad, J. and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm", RFC 3394, September 2002.

[HMAC] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[STDWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2 Informative References

[DSS] National Institute of Standards and Technology. FIPS Pub 186: Digital Signature Standard. 19 May 1994.

[RANDOM] Eastlake 3rd, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.

[RFC2026] Bradner, S., "The Internet Standards Process - Revision 3", BCP 9, RFC 2026, October 1996.

[X.208-88] CCITT. Recommendation X.208: Specification of Abstract Syntax Notation One (ASN.1). 1988.

[X.209-88] CCITT. Recommendation X.209: Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). 1988.

7. Authors' Addresses

Jim Schaad
Soaring Hawk Consulting

EMail: jimsch@exmsft.com

Russell Housley
Vigil Security
918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: housley@vigilsec.com

8. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.