

Network Working Group  
Request for Comments: 4494  
Category: Standards Track

JH. Song  
R. Poovendran  
University of Washington  
J. Lee  
Samsung Electronics  
June 2006

## The AES-CMAC-96 Algorithm and Its Use with IPsec

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2006).

### Abstract

The National Institute of Standards and Technology (NIST) has recently specified the Cipher-based Message Authentication Code (CMAC), which is equivalent to the One-Key CBC-MAC1 (OMAC1) algorithm submitted by Iwata and Kurosawa. OMAC1 efficiently reduces the key size of Extended Cipher Block Chaining mode (XCBC). This memo specifies the use of CMAC mode on the authentication mechanism of the IPsec Encapsulating Security Payload (ESP) and the Authentication Header (AH) protocols. This new algorithm is named AES-CMAC-96.

## 1. Introduction

The National Institute of Standards and Technology (NIST) has recently specified the Cipher-based Message Authentication Code (CMAC). CMAC [NIST-CMAC] is a message authentication code that is based on a symmetric key block cipher such as the Advanced Encryption Standard [NIST-AES]. CMAC is equivalent to the One-Key CBC MAC1 (OMAC1) submitted by Iwata and Kurosawa [OMAC1a, OMAC1b]. OMAC1 is an improvement of the eXtended Cipher Block Chaining mode (XCBC) submitted by Black and Rogaway [XCBCa, XCBCb], which itself is an improvement of the basic CBC-MAC. XCBC efficiently addresses the security deficiencies of CBC-MAC, and OMAC1 efficiently reduces the key size of XCBC.

This memo specifies the usage of CMAC on the authentication mechanism of the IPsec Encapsulating Security Payload [ESP] and Authentication Header [AH] protocols. This new algorithm is named AES-CMAC-96. For further information on AH and ESP, refer to [AH] and [ROADMAP].

## 2. Basic Definitions

|               |   |
|---------------|---|
| CBC           | Cipher Block Chaining mode of operation for message authentication code.  |
| MAC           | Message Authentication Code.<br>A bit string of a fixed length, computed by the MAC generation algorithm, that is used to establish the authority and, hence, the integrity of a message. |
| CMAC          | Cipher-based MAC based on an approved symmetric key block cipher, such as the Advanced Encryption Standard.   |
| Key (K)       | 128-bit (16-octet) key for AES-128 cipher block.<br>Denoted by K.   |
| Message (M)   | Message to be authenticated.<br>Denoted by M.   |
| Length (len)  | The length of message M in octets.<br>Denoted by len.<br>The minimum value is 0. The maximum value is not specified in this document.   |
| truncate(T,l) | Truncate T (MAC) in most-significant-bit-first (MSB-first) order to a length of l octets.   |
| T             | The output of AES-CMAC.   |

|             |   |
|-------------|---|
| Truncated T | The truncated output of AES-CMAC-128 in MSB-first order.  |
| AES-CMAC    | CMAC generation function based on AES block cipher with 128-bit key.  |
| AES-CMAC-96 | IPsec AH and ESP MAC generation function based on AES-CMAC, which truncates the 96 most significant bits of the 128-bit output. |

### 3. AES-CMAC

The core of AES-CMAC-96 is the AES-CMAC [AES-CMAC]. The underlying algorithms for AES-CMAC are the Advanced Encryption Standard cipher block [NIST-AES] and the recently defined CMAC mode of operation [NIST-CMAC]. AES-CMAC provides stronger assurance of data integrity than a checksum or an error detecting code. The verification of a checksum or an error detecting code detects only accidental modifications of the data, while CMAC is designed to detect intentional, unauthorized modifications of the data, as well as accidental modifications. The output of AES-CMAC can validate the input message. Validating the message provides assurance of the integrity and authenticity over the message from the source. According to [NIST-CMAC], at least 64 bits should be used against guessing attacks. AES-CMAC achieves the similar security goal of HMAC [RFC-HMAC]. Since AES-CMAC is based on a symmetric key block cipher (AES), while HMAC is based on a hash function (such as SHA-1), AES-CMAC is appropriate for information systems in which AES is more readily available than a hash function. Detailed information about AES-CMAC is available in [AES-CMAC] and [NIST-CMAC].

4. AES-CMAC-96

For IPsec message authentication on AH and ESP, AES-CMAC-96 should be used. AES-CMAC-96 is a AES-CMAC with 96-bit truncated output in MSB-first order. The output is a 96-bit MAC that will meet the default authenticator length as specified in [AH]. The result of truncation is taken in MSB-first order. For further information on AES-CMAC, refer to [AES-CMAC] and [NIST-CMAC].

Figure 1 describes AES-CMAC-96 algorithm:

In step 1, AES-CMAC is applied to the message M in length len with key K.

In step 2, the output block T is truncated to 12 octets in MSB-first order, and Truncated T (TT) is returned.

```

+++++
+                               Algorithm AES-CMAC-96                               +
+++++
+                               +
+   Input      : K (128-bit Key described in Section 4.1)                       +
+              : M   (message to be authenticated)                             +
+              : len (length of message in octets)                               +
+   Output     : Truncated T (truncated output to length 12 octets)           +
+
+-----+
+
+   Step 1.    T := AES-CMAC (K,M,len);
+   Step 2.    TT := truncate (T, 12);
+              return TT;
+++++

```

Figure 1: Algorithm AES-CMAC-96

## 5. Test Vectors

These test cases are the same as those defined in [NIST-CMAC], with the exception of 96-bit truncation.

```
-----
K          2b7e1516 28aed2a6 abf71588 09cf4f3c
Subkey Generation
AES_128(key,0) 7df76b0c 1ab899b3 3e42f047 b91b546f
K1         fbeed618 35713366 7c85e08f 7236a8de
K2         f7ddac30 6ae266cc f90bc11e e46d513b
```

```
Test Case 1: len = 0
M          <empty string>
AES_CMAC_96  bblld6929 e9593728 7fa37d12
```

```
Test Case 2: len = 16
M          6bc1bee2 2e409f96 e93d7e11 7393172a
AES_CMAC_96  070a16b4 6b4d4144 f79bdd9d
```

```
Test Case 3: len = 40
M          6bc1bee2 2e409f96 e93d7e11 7393172a
          ae2d8a57 1e03ac9c 9eb76fac 45af8e51
          30c81c46 a35ce411
AES_CMAC_96  dfa66747 de9ae630 30ca3261
```

```
Test Case 4: len = 64
M          6bc1bee2 2e409f96 e93d7e11 7393172a
          ae2d8a57 1e03ac9c 9eb76fac 45af8e51
          30c81c46 a35ce411 e5fbc119 1a0a52ef
          f69f2445 df4f9b17 ad2b417b e66c3710
AES_CMAC_96  51f0bebf 7e3b9d92 fc497417
-----
```

## 6. Interaction with the ESP Cipher Mechanism

As of this writing, there are no known issues that preclude the use of AES-CMAC-96 with any specific cipher algorithm.

## 7. Security Considerations

See the security considerations section of [AES-CMAC].

## 8. IANA Considerations

The IANA has allocated value 8 for IKEv2 Transform Type 3 (Integrity Algorithm) to the AUTH\_AES\_CMAC\_96 algorithm.

## 9. Acknowledgements

Portions of this text were borrowed from [NIST-CMAC] and [XCBCa]. We would like to thank to Russ Housley for his useful comments.

We acknowledge the support from the the following grants: Collaborative Technology Alliance (CTA) from US Army Research Laboratory, DAAD19-01-2-0011; Presidential Award from Army Research Office, W911NF-05-1-0491; NSF CAREER, ANI-0093187. Results do not reflect any position of the funding agencies.

## 10. References

### 10.1. Normative References

- [AES-CMAC] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, June 2006.
- [AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.
- [ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [NIST-AES] NIST, FIPS 197, "Advanced Encryption Standard (AES)", November 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
- [NIST-CMAC] NIST, Special Publication 800-38B Draft, "Recommendation for Block Cipher Modes of Operation: The CMAC Method for Authentication", March 9, 2005.

### 10.2. Informative References

- [OMAC1a] Tetsu Iwata and Kaoru Kurosawa, "OMAC: One-Key CBC MAC", Fast Software Encryption, FSE 2003, LNCS 2887, pp. 129-153, Springer-Verlag, 2003.
- [OMAC1b] Tetsu Iwata and Kaoru Kurosawa, "OMAC: One-Key CBC MAC", Submission to NIST, December 2002. Available from <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/omac/omac-spec.pdf>.
- [RFC-HMAC] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

- [ROADMAP] Thayer, R., Doraswamy, N., and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.
- [XCBCa] John Black and Phillip Rogaway, "A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC", NIST Second Modes of Operation Workshop, August 2001. Available from <http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/xcbc-mac/xcbc-mac-spec.pdf>.
- [XCBCb] John Black and Phillip Rogaway, "CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions", *Journal of Cryptology*, Vol. 18, No. 2, pp. 111-132, Springer-Verlag, Spring 2005.

#### Authors' Addresses

Junhyuk Song  
University of Washington  
Samsung Electronics

Phone: (206) 853-5843  
EMail: [songlee@ee.washington.edu](mailto:songlee@ee.washington.edu), [junhyuk.song@samsung.com](mailto:junhyuk.song@samsung.com)

Jicheol Lee  
Samsung Electronics

Phone: +82-31-279-3605  
EMail: [jicheol.lee@samsung.com](mailto:jicheol.lee@samsung.com)

Radha Poovendran  
Network Security Lab (NSL)  
Dept. of Electrical Engineering  
University of Washington

Phone: (206) 221-6512  
EMail: [radha@ee.washington.edu](mailto:radha@ee.washington.edu)

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).