

Network Working Group  
Request for Comments: 4434  
Obsoletes: 3664  
Category: Standards Track

P. Hoffman  
VPN Consortium  
February 2006

The AES-XCBC-PRF-128 Algorithm for  
the Internet Key Exchange Protocol (IKE)

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

Some implementations of IP Security (IPsec) may want to use a pseudo-random function derived from the Advanced Encryption Standard (AES). This document describes such an algorithm, called AES-XCBC-PRF-128.

1. Introduction

[AES-XCBC-MAC] describes a method to use the Advanced Encryption Standard (AES) as a message authentication code (MAC) whose output is 96 bits long. While 96 bits is considered appropriate for a MAC, it is too short to be useful as a long-lived pseudo-random function (PRF) in either IKE version 1 or version 2. Both versions of IKE use the PRF to create keys in a fashion that is dependent on the length of the output of the PRF. Using a PRF that has 96 bits of output creates keys that are easier to attack with brute force than a PRF that uses 128 bits of output.

Fortunately, there is a very simple method to use much of [AES-XCBC-MAC] as a PRF whose output is 128 bits: omit the step that truncates the 128-bit value to 96 bits.

### 1.1. Differences from RFC 3664

This document specifies the same algorithm as RFC 3664 except that the restriction that keys be exactly 128 bits from [AES-XCBC-MAC] is removed. Implementations of RFC 3664 will have the same bits-on-the-wire results as this algorithm; the only difference is that keys that were not equal in length to 128 bits will no longer be rejected but instead will be made 128 bits.

IKEv2 [IKEv2] uses PRFs for multiple purposes, most notably for generating keying material and authentication of the IKE\_SA. The IKEv2 specification differentiates between PRFs with fixed key sizes and those with variable key sizes.

When the PRF described in this document is used with IKEv2, the PRF is considered fixed-length for generating keying material but variable-length for authentication. That is, when generating keying material, "half the bits must come from Ni and half from Nr, taking the first bits of each" as described in IKEv2, section 2.14; but for authenticating with shared secrets (IKEv2, section 2.16), the shared secret does not have to be 128 bits long. This somewhat tortured logic allows IKEv2 implementations that use the fixed-length-key semantics from RFC 3664 to interoperate with implementations that use the variable-length-key semantics of this document.

## 2. The AES-XCBC-PRF-128 Algorithm

The AES-XCBC-PRF-128 algorithm is identical to [AES-XCBC-MAC] except for two changes. First, the key length restriction of exactly 128 bits in [AES-XCBC-MAC] is eliminated, as described below; this brings AES-XCBC-PRF-128 in alignment with HMAC-SHA1 and HMAC-MD5 when they are used as PRFs in IKE. Second, the truncation step in section 4.3 of [AES-XCBC-MAC] is *not* performed; that is, there is no processing after section 4.2 of [AES-XCBC-MAC].

The key for AES-XCBC-PRF-128 is created as follows:

- o If the key is exactly 128 bits long, use it as-is.
- o If the key has fewer than 128 bits, lengthen it to exactly 128 bits by padding it on the right with zero bits.
- o If the key is 129 bits or longer, shorten it to exactly 128 bits by performing the steps in AES-XCBC-PRF-128 (that is, the algorithm described in this document). In that re-application of this algorithm, the key is 128 zero bits; the message is the too-long current key.

## 2.1. Test Vectors

Test Case AES-XCBC-PRF-128 with 20-byte input  
Key : 000102030405060708090a0b0c0d0e0f  
Key Length : 16  
Message : 000102030405060708090a0b0c0d0e0f10111213  
PRF Output : 47f51b4564966215b8985c63055ed308

Test Case AES-XCBC-PRF-128 with 20-byte input  
Key : 00010203040506070809  
Key Length : 10  
Message : 000102030405060708090a0b0c0d0e0f10111213  
PRF Output : 0fa087af7d866e7653434e602fdde835

Test Case AES-XCBC-PRF-128 with 20-byte input  
Key : 000102030405060708090a0b0c0d0e0fedcb  
Key Length : 18  
Message : 000102030405060708090a0b0c0d0e0f10111213  
PRF Output : 8cd3c93ae598a9803006ffb67c40e9e4

## 3. Security Considerations

The security provided by AES-XCBC-MAC-PRF is based on the strengths of AES and HMAC. At the time of this writing, there are no known practical cryptographic attacks against AES, AES-XCBC-MAC-PRF, or HMACs.

As is true with any cryptographic algorithm, part of its strength lies in the security of the key management mechanism, the strength of the associated secret key, and the correctness of the implementations in all the participating systems. [AES-XCBC-MAC] contains test vectors to assist in verifying the correctness of the AES-XCBC-MAC-PRF code. The test vectors all show the full MAC value before it is truncated to 96 bits. The PRF makes use of the full MAC value, not the truncated one.

## 4. IANA Considerations

Any reference to RFC 3664 needs to be updated to refer to this document when it is published.

## 5. Normative References

- [AES-XCBC-MAC] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.
- [IKEv2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

## Appendix A. Acknowledgements

Pasi Eronen suggested the easy method for shortening too-long keys.  
Saroop Mathur and John Black provided and verified the test vectors.

## Author's Address

Paul Hoffman  
VPN Consortium

EMail: [paul.hoffman@vpnc.org](mailto:paul.hoffman@vpnc.org)

## 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).