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:

- If the key is exactly 128 bits long, use it as-is.
- If the key has fewer than 128 bits, lengthen it to exactly 128
  bits by padding it on the right with zero bits.
- 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    : 000102030405060708090a0b0c0d0e0f101111213
PRF Output : 47f51b4564966215b8985c63055ed308

Test Case AES-XCBC-PRF-128 with 20-byte input
Key        : 00010203040506070809
Key Length : 10
Message    : 000102030405060708090a0b0c0d0e0f101111213
PRF Output : 0fa087af7d866e7653434e602fdde835

Test Case AES-XCBC-PRF-128 with 20-byte input
Key        : 000102030405060708090a0b0c0d0e0fedcb
Key Length : 18
Message    : 000102030405060708090a0b0c0d0e0f101111213
PRF Output : 8cd3c93ae598a9803006ff67c40e9e4

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


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