Network Working Group
Request for Comments: 4615
Category: Standards Track

J. Song
R. Poovendran
University of Washington
J. Lee
Samsung Electronics
T. Iwata
Nagoya University

August 2006

The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)

Status of This Memo

Copyright Notice

Abstract

Some implementations of IP Security (IPsec) may want to use a pseudo-random function (PRF) based on the Advanced Encryption Standard (AES). This memo describes such an algorithm, called AES-CMAC-PRF-128. It supports fixed and variable key sizes.

Table of Contents

1. Introduction

[RFC4493] describes a method to use the Advanced Encryption Standard (AES) as a Message Authentication Code (MAC) that has a 128-bit output length. The 128-bit output is useful as a long-lived pseudo-random function (PRF). This document specifies a PRF that supports fixed and variable key sizes for IKEv2 [RFC4306] Key Derivation Function (KDF) and authentication.

2. Basic Definitions

VK Variable-length key for AES-CMAC-PRF-128, denoted by VK.

0^128 The string that consists of 128 zero-bits, which is equivalent to 0x00000000000000000000000000000000 in hexadecimal notation.

AES-CMAC The AES-CMAC algorithm with a 128-bit long key described in section 2.4 of [RFC4493].

3. The AES-CMAC-PRF-128 Algorithm

The AES-CMAC-PRF-128 algorithm is identical to AES-CMAC defined in [RFC4493] except that the 128-bit key length restriction is removed.

IKEv2 [RFC4306] 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 using AES-CMAC-PRF-128 as the PRF described in IKEv2, AES-CMAC-PRF-128 is considered to take fixed size (16 octets) keys for generating keying material but it takes variable key sizes 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 16 octets and the length may vary.

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                        AES-CMAC-PRF-128                          +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+                                                                  +
+ Input  : VK (Variable-length key)                                +
+        : M (Message, i.e., the input data of the PRF)            +
+        : VKlen (length of VK in octets)                          +
+        : len (length of M in octets)                             +
+ Output : PRV (128-bit Pseudo-Random Variable)                    +
+                                                                  +
+------------------------------------------------------------------+
+ Variable: K (128-bit key for AES-CMAC)                           +
+                                                                  +
+ Step 1.   If VKlen is equal to 16                                +
+ Step 1a.  then                                                   +
+              K := VK;                                            +
+ Step 1b.  else                                                   +
+              K := AES-CMAC(0^128, VK, VKlen);                    +
+ Step 2.   PRV := AES-CMAC(K, M, len);                            +
+           return PRV;                                            +
+                                                                  +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Figure 1. The AES-CMAC-PRF-128 Algorithm

In step 1, the 128-bit key, K, for AES-CMAC is derived as follows:

o If the key, VK, is exactly 128 bits, then we use it as-is.

o If it is longer or shorter than 128 bits, then we derive the key, K, by applying the AES-CMAC algorithm using the 128-bit all-zero string as the key and VK as the input message. This step is described in step 1b.

In step 2, we apply the AES-CMAC algorithm using K as the key and M as the input message. The output of this algorithm is returned.

4. Test Vectors

```
---------------------------------------------------------

Test Case AES-CMAC-PRF-128 with 20-octet input
Key        : 00010203 04050607 08090a0b 0c0d0e0f edcb
Key Length : 18
Message    : 00010203 04050607 08090a0b 0c0d0e0f 10111213
PRF Output : 84a348a4 a45d235b abfffc0d 2b4da09a

Test Case AES-CMAC-PRF-128 with 20-octet input
Key        : 00010203 04050607 08090a0b 0c0d0e0f
Key Length : 16
Message    : 00010203 04050607 08090a0b 0c0d0e0f 10111213
PRF Output : 980ae87b 5f4c9c52 14f5b6a8 455e4c2d

Test Case AES-CMAC-PRF-128 with 20-octet input
Key        : 00010203 04050607 0809
Key Length : 10
Message    : 00010203 04050607 08090a0b 0c0d0e0f 10111213
PRF Output : 290d9e11 2edb09ee 141fcf64 c0b72f3d

---------------------------------------------------------
```

5. Security Considerations

The security provided by AES-CMAC-PRF-128 is based upon the strength of AES and AES-CMAC. At the time of this writing, there are no known practical cryptographic attacks against AES or AES-CMAC. However, as is true with any cryptographic algorithm, part of its strength lies in the secret key, VK, and the correctness of the implementation in all of the participating systems. The key, VK, needs to be chosen independently and randomly based on RFC 4086 [RFC4086], and both keys, VK and K, should be kept safe and periodically refreshed. Section 4 presents test vectors that assist in verifying the correctness of the AES-CMAC-PRF-128 code.

If VK is longer than 128 bits and it is shortened to meet the AES-128 key size, then some entropy might be lost. However, as long as VK is longer than 128 bits, then the new key, K, preserves sufficient entropy, i.e., the entropy of K is about 128 bits.

Therefore, we recommend the use of VK that is longer than or equal to 128 bits, and we discourage the use of VK that is shorter than or equal to 64 bits, because of the small entropy.

6. IANA Considerations

IANA has allocated a value of 8 for IKEv2 Transform Type 2 (Pseudo-Random Function) to the PRF_AES128_CMAC algorithm.

7. Acknowledgements

Portions of this text were borrowed from [RFC3664] and [RFC4434]. Many thanks to Russ Housley and Paul Hoffman for suggestions and guidance. We also thank Alfred Hoenes for many useful comments.

We acknowledge support from 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; ONR YIP N00014-04-1-0479. Results do not reflect any position of the funding agencies.

8. References

8.1. Normative References

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, June 2006.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4086] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

8.2. Informative References

[RFC3664] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 3664, January 2004.

[RFC4434] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.

Authors' Addresses

JunHyuk Song
Samsung Electronics
University of Washington
Phone: (206) 853-5843

EMail: junhyuk.song@samsung.com, junhyuk.song@gmail.com

Radha Poovendran
Network Security Lab
University of Washington
Phone: (206) 221-6512

EMail: radha@ee.washington.edu

Jicheol Lee
Samsung Electronics
Phone: +82-31-279-3605

EMail: jicheol.lee@samsung.com

Tetsu Iwata
Nagoya University

EMail: iwata@cse.nagoya-u.ac.jp

Full Copyright Statement

Intellectual Property

Acknowledgement

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