Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 25, 2016

A. Langley
Google Inc

August 24, 2015

CMAC-based Extract-and-Expand Key Derivation Function (CKDF)
draft-agl-ckdf-01

Abstract

This memo describes a KDF based on AES-CMAC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 25, 2016.

Table of Contents

1. Introduction

The HKDF key derivation function, described in [RFC5869], is currently the de-facto KDF for use in a variety of protocols. However, in hardware orientated designs, significant space savings can be achieved if the underlying primitive is AES rather than a cryptographic hash function.

The memo specifies CKDF, the CMAC-based Key Derivation Function. It is, succinctly, HKDF but with HMAC [RFC2104] replaced by CMAC [RFC4493].

2. CKDF

CKDF follows exactly the same structure as [RFC5869] but "HMAC-Hash" is replaced by the function "AES-CMAC" throughout. The "AES-CMAC" function also takes two arguments: the first is a 16 byte key and the second is an input. It returns the AES-CMAC MAC of the input using the given key as an AES key.

Thus, following HKDF, the "CKDF-Extract(salt, IKM)" function takes an optional, 16-byte salt and an arbitrary-length "input keying material" (IKM) message. If no salt is given, the 16-byte, all-zero value is used. It returns the result of "AES-CMAC(key = salt, input = IKM)", called the "pseudorandom key" (PRK), which will be 16 bytes long.

Likewise, the "CKDF-Expand(PRK, info, L)" function takes the PRK result from "CKDF-Extract", an arbitrary "info" argument and a requested number of bytes to produce. It calculates the L-byte result, called the "output keying material" (OKM), as:

```
N = ceil(L/16)
T = T(1) | T(2) | T(3) | ... | T(N)
OKM = first L octets of T

where:
T(0) = empty string (zero length)
T(1) = AES-CMAC(PRK, T(0) | info | 0x01)
T(2) = AES-CMAC(PRK, T(1) | info | 0x02)
T(3) = AES-CMAC(PRK, T(2) | info | 0x03)
...
```

(where the constant concatenated to the end of each T(n) is a single octet.)

Note that AES-CMAC in [RFC4493] is only defined for AES-128 and likewise, so is CKDF. However, the dependency on AES-128 is stronger here because the length of the PRK from "CKDF-Extract" is the AES blocksize of 128 bits. Thus, if one wished to use AES-256 in the future, the PRK would, somehow, need to be 256 bits. Given the complexities of this, those wishing a higher security level should instead use HKDF with a suitable hash function.

3. Test Vectors

3.1. CKDF-Extract

This section contains test vectors for the "CKDF-Extract" function.

```
# These two test vectors are from RFC4493, section 4
Salt: 2b7e1516 28aed2a6 abf71588 09cf4f3c
IKM:  (empty)
PRK:  bb1d6929 e9593728 7fa37d12 9b756746

Salt: 2b7e1516 28aed2a6 abf71588 09cf4f3c
IKM:  6bc1bee2 2e409f96 e93d7e11 7393172a
PRK:  070a16b4 6b4d4144 f79bdd9d d04a287c

Salt: (none)
IKM:  73656372 6574206b 6579
PRK:  6f79b401 ea761a01 00b7ca60 c178b69d
```

3.2. CKDF-Expand

This section contains test vectors for the "CKDF-Expand" function.

```
PRK:  6f79b401 ea761a01 00b7ca60 c178b69d
Info: (empty)
L:    32
OKM:  922da31d 7e1955f0 6a56464b 5feb7032 8f7e6f60 aaea5735
      c2772e33 17d0a288

PRK:  6f79b401 ea761a01 00b7ca60 c178b69d
Info: 696e666f 20737472 696e67
L:    256
OKM:  6174e672 12e1234b 6e05bfd3 1043422c df1e34cd 29ee09f5
      bd5edb90 db39dcd4 c301e873 d91acbd5 333c8701 6dda05be
      3a8faade 2c3992c8 f3221f05 5efb3b51 76dbbe76 90cb4400
      f737298d 638b8026 d527c1e5 81f4e37d a0499c31 abfd8908
      207160de 343c126e cb460e38 8481fa9f 73391fe6 35a0e4b6
      cde3d385 78bcb8b5 5a60952b ac6f840f d87c397a c2477992
      ac6cbd64 3100e3ca d660373b 44e2fc0e 4867b15a cd9a070a
      3229ee40 76bf9851 7ccc656f 5bf1f8bb 41ce7e2d 48db670f
      1b2921ee 462d9cf1 987eb983 e5c2ce4e a9ceea10 c301dcca
      f16c4b57 67daa4bf 6ecc8161 77da31a5 9a9b1972 86259bd6
      598d2874 a4f605fb 877bee1b 5529873f
```

4. Security Considerations

Since CKDF is so closely based on HKDF, the security considerations are the same and sections 3, 4 and 5 of [RFC5869] are included here by reference.

5. IANA Considerations

None.

6. References

6.1. Normative References

[RFC4493] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, DOI 10.17487/RFC4493, June 2006, <http://www.rfc-editor.org/info/rfc4493>.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <http://www.rfc-editor.org/info/rfc5869>.

6.2. Informative References

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, DOI 10.17487/RFC2104, February 1997, <http://www.rfc-editor.org/info/rfc2104>.

Author's Address

Adam Langley
Google Inc

Email: agl@google.com