Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 22, 2016

A. Langley
Google Inc

August 21, 2015

# CMAC-based Extract-and-Expand Key Derivation Function (CKDF)
# draft-agl-ckdf-00

## 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 22, 2016.

## Copyright Notice

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 f3e99629 5165f6c6 0e08ba43 2dd9058b
```

```
PRK:  6f79b401 ea761a01 00b7ca60 c178b69d
Info: 696e666f 20737472 696e67
L:    256
OKM:
  6174e672 12e1234b 6e05bfd3 1043422c 7ab6dc31 5db7d98d 013ab332 924b7fe9
  0ae9a89d 09c93be4 0ce525e0 b6f0d37d f3818191 3aa3d588 f75a3594 ef7a93ac
  d791331e 7929de8b c8c8a6ee 2dd9960e c57fe159 610676a7 c118c4aa c2d34a89
  6edd3691 f0e922a3 0eecc7b3 ec3eaa91 13d4ee51 8b0a4c7e d0b475df bd07ee02
  a3470832 da247ef3 b07f9acd 8ddbb765 7369e1c5 2942fab2 11d47c44 0d6818f8
  29cdd8da d84b825e 1166cbdc dbb13904 d6753de7 6070a145 a8572496 c2808567
  9459d801 f14449fb f3430a83 685a4b8d 091dc2fc 85b8209d 7cfd5dbd 39d79a8d
  d7c6f981 af064ce6 9e58a99f bd9ffd58 a2d93d60 972ec873 f27feaed eed73f0a
```

## 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