FDO KDF Strength Analysis

Final Document, May 22, 2024

This version:
https://fidoalliance.org/specs/fidoiot/fdo-appnote-7-kdf-small-hmac.html

Issue Tracking:
GitHub

Editor:
Geoffrey Cooper (Intel)

Version:
1.0

Abstract

In the FDO specification, section 4.4 has a table called "Cipher Suite Names and Meanings" that gives the Key Derivation Function "KDF" to use for each supported cipher suite. In this table, the AES-128 and AES-256 authenticated encryption suites all use the same KDF with HMAC-SHA-256. This document discusses the implications of using this KDF with AES-256.

Table of Contents

1. Introduction§

In the FDO specification, section 4.4 has a table called "Cipher Suite Names and Meanings" that gives the Key Derivation Function "KDF" to use for each supported cipher suite. In this table, the AES-128 and AES-256 authenticated encryption suites all use the same KDF with HMAC-SHA-256. This document discusses the implications of using this KDF with AES-256.

The Key Derivation Function is described in section 3.6.4 of FIDO Device Onboard. It is based on*KDF In Counter Mode* from section 5.1 in [SP800-108]. The FDO specification section defines how the KDF variables from [SP800-108] are realized. The PRF is given as HMAC-SHA256 or HMAC-SHA384, and references the above-mentioned "Cipher Suite Names and Meanings" table in section 4.4.

1.1. SP800-108 and KDF description§

The description of the KDF in the FIDO Specification has been reviewed in the past and meets the security requirements of the FDO protocol. SP800-108 has since been updated by the NIST, as SP800-108r1-upd1 [SP800-108r1-upd1]. However, the KDF in Counter Mode is unchanged across the versions.

The original document [SP800-108] refers to the "machine integer size" for some variables (e.g., the variable L), and the FDO specification gives specific values for all machines to use, so that the computation will be the same on all machine architectures, including where machine word sizes and byte ordering might vary.

This means it is possible that any given crypto package might support this KDF, but not be compatible with FDO. We have seen this to be true, for example, in a Java-based crypto package. Although this incompatibility is annoying to implementors, it does not represent a weakness in the FDO specification. It may pose a burden on implementors who wish to obtain FIPS certification since the pre-certified KDF in the crypto package cannot be used with FDO.

1.2. Choice of Hash for KDF§

The KDF for FDO uses an HMAC based on SHA256 or SHA384. The definition of the HMAC is given in [FIPS198-1] "The Keyed-Hash Message Authentication Code". The readers and implementors are directed to this source; the following is for illustration purposes only.

The HMAC in [FIPS198-1] can be simplified to:

```
HMAC(text) ::= SHA-N( (secret XOR 0x6363...) ||
                      (secret XOR 0x3636...) ||
                      text )
```

Where:

- SHA-N is SHA256 or SHA384
- the elipsis (...) indicates a repeating pattern for 256 or 384 bits
- the || operator is bitwise concatenation
- "secret" has been extended to cover 256 or 384 bits

We can see here that the cryptographic strength of the HMAC is based on the strength of the hash function in use, which is either SHA256 or SHA384. The cryptographic strength is thus bound by the strength of SHA used.

The defined KDF uses a SHA256 for all cipher suites containing AES128. For cipher suites containing AES256, the defined KDF is inconsistent, with SHA256 used for A256GCM and AES-256-CCM-64-128-256 ciphers, while SHA384 is used for the AES256/CTR/HMAC-SHA384 suite.

To be honest, the authors intended that SHA384 be used as the KDF for all the AES256 suites. The smaller SHA256 value for AES256GCM and AES-256-CCM-64-128-256 can be thought of as a typographic error in the spec. At question is whether this error results in a weaker strength of cryptography.

NIST SP800-57, part 1, revision 5, table 3[SP800-57pt1r5] gives maximum security strengths for hash and hash-based functions. The table gives security strength for digital signatures and also for Key Derivation Functions (the difference is that KDF does not require collision resistance).

We excerpt the part of the table that is apropros to this discussion:

Security Strength	KDF
128	SHA-1
≥ 256	SHA256, SHA384, SHA512 ...

The cryptographic strength of AES-128 is 128 bits and AES-256 is 256 bits.

So we can see that:

- SHA256 combined with AES-128 gives a security strength of 128 bits

- SHA256 combined with AES-256 gives a security strength of 256 bits
- SHA384 combined with AES-256 gives a security strength of 256 bits

As a side note, NIST recommendations for resistance to PQ computing for 2019-2030 is a key strength of 192 bits [SP800-57pt1r5].

[As of this writing, the following website has a nice summary of multiple NIST and other sources: https://www.keylength.com/en/4/]

So we can see that in the FDO specification table in section 4.4, that the smaller HMAC size does not impact the cryptographic strength of 128 bits for all 4 ciphers, since the strength of AES is the limiting factor in all cases.

Cipher	KDF strength	Security strength
A128GCM	128 bits	**128 bits**
AES-CCM-64-128-128	128 bits	**128 bits**
AES128/CTR/HMAC-SHA256 AES128/CBC/HMAC-SHA256	256 bits	**128 bits**
A256GCM	256 bits	**256 bits**
AES-CCM-64-128-256	256 bits	**256 bits**
AES256/CTR/HMAC-SHA384 AES256/CBC/HMAC-SHA384	384 bits	**256 bits**

For FDO 1.2 we can consider bringing upgrading AES-256 suites to use the larger hash for consistency.

As an aside, for FDO 1.2, we can consider using SHA-512 in place of SHA-384:

- SHA-384 uses SHA-512 with part of the key fixed (i.e., computation of SHA-384 and SHA-512 is essentially the same)
- Additional overhead of storing 512-bit HMAC key versus 384-bit HMAC key is small.

1.3. Conclusion§

For users of FDO 1.1, for situations where existing users of FDO need the strongest cryptographic strength, we recommend using any of the cipher suites with AES-256:

- A256GCM
- AES-CCM-64-128-256
- AES256/CTR/HMAC-SHA384.

For FDO 1.2, we may choose to provide larger cipher suites to protect against attacks from early quantum computers. This could involve updating the hash further from SHA256 and even from SHA-384 to SHA-512.

As an alternative, we notice that TLS-1.3 uses the HKDF as defined in REF #RFC5869. This KDF is very similar to the KDF used in FIDO Device Onboard:

- HMAC is used over text with a counter to create random bits.
- Context information is added into the mix (e.g., the string "FDO 1.2")
- A counter is used to expand the KDF to more output if needed.

In addition, the KDF formula is commonly implemented by crypto packages and (because it is used in TLS) is implemented consistently on multiple computer architectures.

References§

Informative References§

[FIPS198-1]

FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC). July 2008. URL: http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[SP800-108]

Lily Chen. *NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions*. October 2009. URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf

[SP800-108r1-upd1]

Lily Chen. *NIST Special Publication 800-108r1-upd1: Recommendation for Key Derivation Using Pseudorandom Functions*. August 2022. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-108r1-upd1.pdf

[SP800-57pt1r5]

Elaine Barker. *NIST Special Publication 800-57 part 1, revision 5: Recommendation for Key Management*. May 2020. URL: https://doi.org/10.6028/NIST.SP.800-57pt1r5