Public Comments on SP 800-135r1: Recommendation for Existing Application-Specific Key Derivation Functions

Comment period: July 27, 2023 – September 27, 2023

On July 27, 2023, NIST's Crypto Publication Review Board announced the review of SP 800-135r1 *Recommendation for Existing Application-Specific Key Derivation Functions*.

The comments that NIST received during the comment period are collected below.

More information about this review is available from NIST's Crypto Publication Review Project site.

LIST OF COMMENTS

1. Comments from Panos Kampanakis, Adam Petcher, Nevine Ebeid, Amazon, September 22, 2023

We would like to provide some feedback regarding the NIST Request for Public Comments on SP 800-135 Revision 1:

1. We suggest for section 4.2 of the SP to add an explicit mention of TLS 1.3 key derivation.

The TLS 1.3 key derivation already complies with the Extract-and-Expand methodology described in NIST SP 800-56Cr2 and SP 800-133rev2 and the key expansion of the TLS 1.3 key schedule follows the expansion described as KDF in Feedback mode in SP 800-108. This TLS 1.3 key derivation is already approved by FIPS as the FIPS 140-3 Implementation Guidance already mentions

The TLS 1.3 key derivation function documented in Section 7.1 of RFC 8446. This is considered an approved CVL because the underlying functions performed within the TLS 1.3 KDF map to NIST approved standards, namely: SP 800-133rev2 (Section 6.3 Option #3), SP 800-56Crev2, and SP 800-108.

Thus, TLS 1.3 should be called out in section 4.2 or a subsection of the SP.

We suggest to add a mention in Section 4 about the TLS 1.3 HKDF-Expand using the concatenated classical and post-quantum shared secrets as being FIPS approved.

This is already approved in Section 2 of SP 800-56Cr2., but given that SP 800-135 is getting updated, it could benefit by specifically calling out that concatenating the classical and the post-quantum (PQ) shared secret in TLS 1.3 key derivation as suggested in the IETF drafts draft-ietf-tls-hybrid-design and draft-kwiatkowski-tls-ecdhe-kyber is approved.

We suggest to add a mention in Section 5.2 about the SSH key derivation using the concatenated classical and post-quantum shared secrets as being approved.

This is already approved in section 2 of SP 800-56Cr2., but given that SP 800-135 is getting updated, it could benefit by specifically calling out that concatenating the classical and the PQ shared secret as suggested in the IETF drafts draft-kampanakis-curdle-ssh-pq-ke and draft-josefsson-ntruprime-ssh is approved. Note that both drafts concatenate and hash the two shared secrets before passing it through the SSH KDF, but draft-josefsson-ntruprime-ssh uses X25519 which is not FIPS approved.

We suggest to add QUIC key derivation for encrypting data sent over the tunnel

QUIC uses TLS 1.3 to generate data encryption keys (section 5.1 of RFC9001). It leverages the HKDF-Expand-Label with a zero-length context as per TLS 1.3 to derive data encryption keys from a shared secret with HKDF-Extract. Such a Extract-then-Expand mechanisms are approved for key agreement as per NIST SP 800-56Cr2. HKDF-Expand-Label with a zero-length context is approved as a KDF in Feedback mode in SP 800-108. The FIPS 140-3 IG says

Every module that implements a full key agreement scheme shall use only the approved key derivation functions documented in SP 800-56Crev1 or rev2 or in IG 2.4.B. Note that all SP 800-135rev1 KDFs and the TLS 1.3 KDF are included in IG 2.4.B.

Thus, although not specifically called out, QUIC key establishment and key derivation for data encryption are FIPS approved because they follow TLS 1.3. It would be beneficial to explicitly call out QUIC key derivation for data encryption in the SP.

Note that QUIC generates keys for encrypting parts of the headers, initial packets and protecting the integrity of specific packets. Encryption is provided by AES-GCM and integrity for specific packets by GMAC. The key derivations used are based on HKDF-Extract and HKDF-Expand with keys coming from the packets or that are static. The protocol does not do a key exchange for protecting these packets or fields because the goal is to protect and anonymize the negotiation, not to protect the data. In essence these mechanism's use key derivations which could be approved as KBKDF compliant, but overall, we don't think it is necessary. We think that NIST does not need to approve protocol packet protection mechanisms which ensure the integrity of protocol fields and protect some protocol frames from eavesdroppers, but do not protect data. Approving only data encryption key derivation for QUIC would suffice.

If QUIC key derivation ends up getting added as we suggested (bullet 4), we also suggest to call out the PQ hybrid key derivation for QUIC (inherited by TLS 1.3).

This suggestion is similar to the ones about PQ hybrid key derivation suggested for TLS 1.3 (bullet 2). If PQ hybrid TLS 1.3 is called out in the updated SP 800-135, then QUIC consequently will inherit it. This, it would only need one sentence to specifically mention that the PQ hybrid key exchange in TLS 1.3 automatically means it is approved for use within QUIC as well.

We suggest to add the key derivation used in HPKE for deriving symmetric keys used in AEAD.

RFC9180, the HPKE spec, specifies that Extract-then-Expand is used for deriving a shared-secret which is then used in the key schedule that derives the encryption keys and the nonce. The key schedule itself is using an HKDF-Expand-Label and HKDF-Extract-Label like in TLS 1.3. So, the key derivation for this use-case can be approved.

We expect that in the long term HPKE (base mode) can be the replacement of RSA for asymmetric encryption, so given that SP 800-135 is getting updated, it could benefit by specifically calling out HPKE key derivation.

If HPKE key derivation ends up getting added as we suggested (bullet 6), we also suggest to call out the PQ hybrid key derivation for HPKE.

This suggestion is similar to the ones about PQ hybrid key derivation suggested for TLS 1.3 (bullet 2) and SSH (bullet 3) above. This is already approved in section 2 of SP 800-56Cr2.

PQ hybrid HPKE was explored in ia.cr/2022/414. IETF draft draft-westerbaan-cfrg-hpke-xyber768d00 is the first attempt for a specification. Note that draft-westerbaan-cfrg-hpke-xyber768d00 uses two not yet FIPS-approved KEMs (X25519 and Kyber), so it could probably not be referenced in the updated SP 800-135 anyway, but this draft has a security proof caveat. The hybrid KEM described in Section 3 runs the two component KEMs in parallel and then returns the concatenation of the secrets, public keys, ciphertexts. Section 4 describes how the shared secret is passed through HKDF (via KeySchedule), but it does not use the ciphertexts when deriving the key. It is unclear what assumptions are made about the component KEMs and the KDF in order for this construction to be IND-CCA.

A combiner that mixes both ciphertexts into the KDF results in IND-CCA KEM under the sole assumption that one component KEM is IND-CCA. Such combiners are used in the PQ TLS 1.3 and SSH drafts referenced above. We intend to publish a new draft with P256+Kyber which will more appropriately stir the ciphertexts in the KDF. We think this could be referenced in the updated SP.

2. Comments from Jonathan Smith, Dekra, September 25, 2023

Here are my comments on ways the SP 800-135 Rev. 1 document could be updated and improved.

1. I noticed a number of references to FIPS 180-3; which I assume will be updated to either point to 180-4, or else to refer automatically to whatever the latest release in the FIPS 180 series is. Suggest also explicitly allowing approved hash algorithms from FIPS 202 in the places that currently allow ones from just FIPS 180-3.
   I'd suggest updating the SP with the newer TLS KDFs
   a. The "extended master secret" TLS 1.2 KDF from RFC 7627.

      Note, the TLS 1.2 PRF is unchanged for RFC 7627 – what did change is the content of the parameters fed into it to calculate the master secret. So including this RFC shouldn't require any technical change to section 4.2.2; and mostly just an update to section 1's list of standards. (Though a short sentence in 4.2.2 mentioning that the text applies to both RFC 5246 and 7627 would be helpful)

   b. The TLS 1.3 KDF from RFC 8446, August 2018.
      (Currently CMVP has adopted those for 140-3 modules in 140-3 IG D.Q and SP 800-140Dr1 respectively – but it'd be nice to have then wrapped into the main special pub on industry KDFs)

   Also, I'm not sure on the timing of the release of the revised SP 800-135, but I believe there is effort in the IETF to extend TLS 1.3 to support hybrid (classical + post quantum) key establishment and that effort might end up impacting the KDF (or at least explicitly specifying how the two different keying materials are combined and fed into it). It might be good to hold off on releasing the updated SP until that's been settled and can be address in the new 800-135.

   It would be helpful to note which KDF's in SP 800-135 rely on SHA-1 (e.g. TLS 1.0/1.1, or SNMP) and warn that SHA-1 is scheduled to become disallowed on December 31, 2030 (which will also make any KDF using SHA-1 disallowed).

   Suggest updating section 4.2 to highlight that the TLS RSA key transport scheme uses PKCS #1 v1.5 encapsulation (which is important to know because it is not SP 800-56Br2 compliant, and thus cannot be used in any new FIPS 140-3 module submissions). It might also be helpful to note here that TLS 1.3 doesn't support RSA key transport.

Suggest updating the example hash and standard in section 5.2 from “SHA-1 as specified in FIPS 180-3” to “SHA2-256 as specified in FIPS 180-4” (because SHA-1 is going away).

Section 5.4 – RFC 3411 obsoleted RFC 2571, and RFC 3414 (updated by RFC 5590) obsoleted RFC 2574. However while RFC 3414 says that future additional authentication protocols (not based on MD5 or SHA-1) may be defined as needs arise; none appear to have been defined yet (though I guess the section could be written to allow SNMP KDF with any.

Section 5.5, request updating the section name to call out version 1.2 of the TPM spec (as TPM 2.0 has been out for a while and no longer uses this KDF; instead using an SP 800-108 compliant one). Making it even clearer that this section only applies to the older TPM version would be somewhat helpful.

For TPM 2.0 the KDF is used in the derivation of asymmetric keys. While I think NIST is working on more general guidelines for that it might be useful to add a TPM 2.0 section talking about how it is approved to use the output of the 800-108 KDF to derive asymmetric keys in the way TPM 2.0 calls for.

3. Comments from Canadian Center for Cyber Security, September 27, 2023

SP-800-135 Rev-1

SP 800-135 provides recommendations for Application-Specific Key Derivation Functions. We provide some general comments, followed by more specific comments by section.

General Comments

Most of the RFCs referenced have since been updated and need to be replaced.

ANSI documents – several of the ANSI references have been updated and changes need to be addressed. Specific documents are listed in the section comments below.

Protocol versions- Several protocol versions referenced in the document have been deprecated such as TLS versions 1.0 and 1.1 and IKEv1. These can probably be removed completely.

Hash algorithm SHA-1 - referenced quite often and should be replaced where possible (by SHA-2 most likely).

Choice of 'other' key derivation functions - ANSI X9-42 and 63 and KDFs in protocols SSH, SNMP, SRTP and TPM, are shown as examples of 'other existing key derivation functions'. We recommend providing a more comprehensive list of protocols that use a custom KDF that do not follow the generic mechanism.

Diffie-Hellman – DH is the only key agreement algorithm defined in the standard, but definitions for ECDH should also be included. Depending on the timeline of this publication, consider adding post-quantum key agreement schemes.

Quantum protection - almost all protocols support the inclusion of a quantum-safe component to their key derivation process. These versions are important to add along with comments on their secure use.

Review of test vectors

The KDFs of SSH/TLS 1.2/IKEv2 were successfully implemented and run against the NIST test vectors. Results show that:

The SSH test vectors do not test the recursion of the algorithm.

The TLS 1.2 KDF as described in the document is not as clear as the IKE and SSH sections, and as a result was more difficult to implement. It would be helpful to include diagrams and/or equations as in the IKE and SSH sections.

Comments by section

Section 4.1 IKE

The next version of SP800-135 will need to consider recent IKEv2 post-quantum RFCs that propose modified key derivation processes. These include the use of a PPK (RFC 8784), IKE intermediate exchange (RFC 9242) to transfer large public keys, which requires calculation of multiple IKE SA keys and the Hybrid exchange (RFC 9370) which uses the IKEv2 intermediate exchanges to establish a shared key based on multiple key exchanges. Key derivation mechanisms proposed in these RFCs are different enough from the original IKEv2 to warrant discussion and approval statements.

Also, general statements about key exchange algorithms need to be reviewed, for example in the statement “In IKEv2, an HMAC is used as a randomness extraction step to extract randomness from a Diffie-Hellman shared secret (g’’)”, the term Diffie-Hellman needs to be replaced by the more appropriate “Key Exchange Method", to reflect the possibility of a Hybrid key exchange and/or the addition of a PPK.

Section 4.2 TLS

Add a new subsection for Key derivation in TLS version 1.3. Include a full description of the protocol with diagrams and equations and describe the conditions that are required for it to be an approved KDF. Will need to review the latest terminology proposed in draft-ietf-tls-rfc8446bis-06 (if approved) that is meant to replace RFC8446.

Next version will need to address post-quantum versions of key derivations and general statements such as a “Diffie-Hellman (DH) key agreement or RSA key transport scheme is used to generate a pre-master secret” are no longer true.

Section 5.1 X9.42-2001 and X9.63-2001

This section suffers seriously from references to outdated documents, including the ANSI documents themselves and the outdated RFCs. ANSI X9.63 was revised in 2011, and ANSI X9.42 is currently under revision and a new version will be published soon. FIPS 180-3 should be updated to FIPS 180-4 or later since it is also under review.

Section 5.2 SSH

The draft draft-kampanakis-curdle-ssh-pq-ke-00 proposes a post quantum hybrid key exchange for SSH, consisting of two key exchanges: one classical and one post-quantum. The resulting key material K will be the combination of the shared secrets of the two exchanges, K_CL and K_PQ: K = K_CL || K_PQ. Although the draft may not make it to RFC status, this may need to be addressed since it is supported in OpenSSH. We recommend maintaining awareness of draft RFCs in development that, if standardized, may modify SSH based key derivation.

Section 5.3 SRTP

No mention is made of the DTLS-SRTP key derivation process (RFC 7983/RFC 5764) when DTLS is used for key transport.

A different key derivation process is described in RFC 8723 Double Encryption Procedures for the Secure Real-Time Transport Protocol (SRTP). Should this be added to the next version of SP800-135 with recommendations on its secure use?

Section 5.5 TPM

Session key calculation has drastically changed in TPM2.0. There are now four types of sessions with distinct session key calculation. Discuss whether these remain conformant? Will some versions not be recommended?

4. Comments from Sophie Schmieg, Google, September 27, 2023

Comments for NIST SP 800-135 Revision 1

Recommendation for Existing Application-Specific Key Derivation Functions

Hash-based Key Derivation Function (HKDF):

Proposed Change:

Request for a clear statement from NIST acknowledging approval of HKDF for all purposes explicitly in this document/or a separate publication.

Justification:

HKDF, as specified in RFC 5869 with its two component algorithms HKDF-extract and HKDF-expand is by far the most popular key derivation function in practice, used in TLS 1.3, IPSec, and many more protocols. It is well researched and more robust than many other SHA2 based key derivation methods. Meanwhile, it is only standardized in NIST SP 800-56C, which allows usage in key establishment schemes. It is not present in NIST SP 800-135 (where TLS 1.3 is missing) and only implicitly standardized in NIST SP 800-108r1, which mentions counter and chaining modes, but only separately and not in the combined fashion that HKDF employs, with the NIST standardized "double pipeline mode" being materially different from HKDF-expand. HKDF-extract is missing entirely from NIST SP 800-108r1. It would be good to clarify that HKDF as specified in RFC 5869 is an approved key derivation function, either by explicitly listing TLS 1.3 in NIST SP 800-135r1 or by listing HKDF in NIST SP 800-108r1, or both.

5. Comments from Joachim Vandersmissen, Atsec, September 29, 2023

Clause/Subclaus e(e.g. 3.1), Paragraph/ Figure/ Table(e.g.Table1)	Type of comment(e.g., ge = general, te = technical, ed = editorial)	Comment (Include rationale for comment)	Suggested change
Section 4.1.1	te	Section 4.1 refers to RFC 2409 for IKEv1, which only standardizes MD5 and SHA-1 for usage in the protocol. This RFC was updated by RFC 4109, but again only MD5 and SHA-1 are specified. However, bullet (3) states that "The HMAC and HASH are NIST-approved algorithms and are used as specified in FIPSs 198-1 [FIPS 198-1] and 180-3, respectively." This leaves ambiguity as to exactly which hash algorithms are approved for the IKEv1 KDF. A restrictive reading would allow only SHA-1.	Change bullet (3) to: "The HMAC is a NIST- approved algorithm used as specified in FIPS 198-1. The HASH is SHA-1 used as specified in FIPS 180-4."
Section 4.1.2	ed	Section 4.1.2 explains that the IKEv2 KDFs are compliant with SP 800-56C, so it follows that these KDFs are approved for any approved hash function. A reference to FIPS 202 should be added to clarify that SHA-3 is part of the approved hash functions.	Change the final paragraph of Section 4.1.2 to "The IKEv2 KDFs, which are compliant with SP 800-56C, are approved when used with an approved HMAC function using an approved hash function; see FIPSs 198-1, and 180-4 or 202, respectively."
Section 4.2.1	ed	The reference to FIPS 180-3 should be updated.	Replace "180-3" with "180-4".
Section 4.2.2	ed	The reference to FIPS 180-3 should be updated.	Replace "180-3" with "180-4".
Section 4.2	te	Consider adding the TLS 1.3 KDF to the standard. This KDF is currently an approved component per FIPS 140-3 IG 2.4.B.	A new Subsection 4.2.3 is added to the standard which approves the TLS 1.3 KDF as specified in RFC 8446, with the SHA-256 and SHA-384 hash algorithms. The RFC does not allow any other hash algorithms (Appendix B.4).

Section 5.1	te	It is unclear which hash functions are approved for use in the ANS X9.42-2001 and ANS X9.63- 2001 KDFs. According to discussions, the CAVP is of the opinion that SHA-1 would not be approved for the ANS X9.63-2001 KDF because that standard only allows its usage to generate 80-bit symmetric keys. A similar rationale might apply to ANS X9.42-2001. SP 800-315 should explicitly state which hash functions are approved, and preferably SHA-1 would be allowed still.	Change bullet (2) to: "The hash function is one of SHA-1, SHA-2 (FIPS 180-4), or SHA-3 (FIPS 202)."
Section 5.2	te	Section 5.2 refers to RFC 4253 for the SSH key derivation function, which only standardizes MD5 and SHA-1 for usage in the protocol. This RFC was updated by RFC 6668, which adds the SHA-256 and SHA-512 algorithms. However, bullet (2) states that "The hash function is one of the hash functions specified in FIPS 180-3." This leaves ambiguity as to exactly which hash algorithms are approved for the SSH KDF. A restrictive reading would allow only SHA-1, SHA-256, and SHA-512. RFC 6668 considered SHA-224 and SHA-384 but did not add them to the list "as they have the same computational requirements of HMAC-SHA2-256 and HMAC- SHA2-512, respectively, and do not seem to be much used in practice." On the other hand, there is no technical reason for SHA-224, SHA- 384, SHA-512/224, SHA-512/256, and SHA-3 not to be allowed. SP 800-315 should explicitly state which hash functions are approved.	Change bullet (2) to "The hash function is one of SHA-1, SHA-2 (FIPS 180-4), or SHA-3 (FIPS 202)."
Section 5.4	ed	The reference to FIPS 180-3 should be updated.	Replace "180-3" with "180-4".

Section 5.5	ed	The reference to FIPS 180-3 should be updated.	Replace "180-3" with "180-4".