ERICSSON

Date: January 18, 2022

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-108 Rev. 1 (Draft): Recommendation for Key Derivation Using Pseudorandom Functions

Dear NIST,

Thanks for your continuous efforts to produce well-written open-access security documents. SP 800-108 is an important document that should be updated.

Please find below our comments on SP 800-108 Rev. 1 (Draft):

- HKDF [1], [2] is being used in a huge number of very important protocols such as IKEv2, TLS 1.3, Encrypted Content-Encoding for HTTP, COSE, CMS, ZRTP, HIPv2, IKEv2, OSCORE, HPKE, EDHOC, MLS, Signal, and WireGuard. TLS 1.3 and OSCORE is furthermore the basis for many other protocols such as DTLS 1.3, QUIC, DTLS/SCTP, EAP-TLS 1.3, EAP-TTLS 1.3, TEAP 1.3, PEAP 1.3, EAP-FAST 1.3 and Group OSCORE. We strongly recommend that NIST introduces HKDF as a compliant KDF for all use cases. We note that HKDF-Expand is similar but not equal to the Feedback mode.

- It would be good if the document gave some guidance to the reader. Are any of the modes defined in Sections 5.1, 5.2, 5.3, or 5.4 preferred for new implementations? Are the security properties different? Are any of the modes included mainly for interoperability with older systems? We suggest that KMAC and HKDF-Expand is recommended for new systems.

- In [1], Krawczyk writes the following regarding SP 800-56A

  "Another peculiarity of NIST's KDF is that the document ([57], Sec. 5.8) explicitly forbids (with a strong shall not) the use of the KDF for generating non-secret randomness, such as IVs. If there is a good reason to forbid this, namely the KDF is insecure with such a use, then there must be something very wrong about this function (or, at least, it shows little confidence in the strength of the scheme). First, it is virtually impossible to keep general applications and implementors from using the KDF to derive auxiliary random material such as IVs. Second, if the leakage of an output from the KDF, in this case the public IV, can compromise other (secret) keys output by the KDF, then the scheme is fully broken; indeed, it is an essential requirement that the leakage of one key produced by the KDF should not compromise other such keys."

  As SP 800-108 Rev. 1 defines the output of the KDFs as secret (pseudorandom) parameters, our understanding is that SP 800-108 Rev. 1 also forbids the use of KDFs for generating non-secret randomness, such as IVs. Krawczyk's comment then applies also to SP 800-108

Rev. 1. We note that it is very common practice to generate non-secret randomness with KDFs, in addition to IVs, other examples are the challenge in EAP-TTLS, the RAND in 3GPP AKAs, and the Session-Id in EAP-TLS 1.3. We recommend NIST to update both SP 800-108 and SP 800-56A to allow limited generation of non-secret randomness. Alternatively, NIST should describe how both secret and non-secret randomness can be derived by a protocol in a compliant way. The amount of non-secret randomness could be much more limited than the generation of secret randomness. As there are no restrictions on the number of labels and contexts, we note that the generation of secret randomness from a single key-derivation key is currently unlimited. If NIST wants to continue forbidding the use of KDFs for generating non-secret randomness, SP 800-108 Rev. 1 should be explicit like SP 800-56A.

– SP 800-108 Rev. 1 also states that the use of the keying material as a keystream is not compliant with the recommendation because it has not been investigated. In the known-plaintext model, keystream is just non-secret randomness and Krawczyk's statement above apply here as well. If HMAC-SHA256 in any of the SP 800-108 modes is a secure PRF, it is also a secure stream cipher and vice versa. It is hard to imagine that AES-CTR which due to the birthday bound quickly deviates from a PRF would provide stronger confidentiality then HMAC-SHA256 in feedback mode. AES-CTR is used because it is efficient and secure enough, not because it is a particularly good PRF. In constrained IoT devices it is beneficial if the same primitive can be used for many different use cases (key derivation, confidentiality, integrity protection, authentication, generation of non-secret randomness). We recommend that the use of the KDFs for generating keystream is treated in the same ways as generation of non-secret randomness, i.e., allowed but much more limited than generation of secret randomness.

[1] "Cryptographic Extraction and Key Derivation: The HKDF Scheme",
https://eprint.iacr.org/2010/264

[2] "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)",
https://datatracker.ietf.org/doc/html/rfc5869

Best Regards,
John Preuß Mattsson,
Senior Specialist, Ericsson