Remarks on Respons to Key Recovery Attacks on AES-GCM-SIV – Response #2

Shay Gueron and Yehuda Lindell

We thank you very much for your response, and greatly appreciate your feedback and analysis.

We would like to point out what we recommend (and plan to post) a maximal number of usages (q_{max}) of a single master key, with AES-GCM-SIV. This would better put things in context. As we promised to the CFRG community since we posted the specifications, we are working on a paper that analyzes the scheme and derives this recommendation. Hopefully, it will be ready in a few weeks. In order to explain our bound, let us first denote the 2-key specification in the CCS paper by GCM-SIV (K1, K2, N, AAD, M). The CFRG proposal can then be viewed as a two step protocol:

1. *Key derivation*: Compute (Record_Encryption_Key, Record_Hash_Key) = KDF (Master_Key, N)
2. *Encryption*: Compute (Tag, C) = GCM-SIV (Record_Encryption_Key, Record_Hash_Key, N, AAD, M)

We now consider the recommended restriction regarding the number q_{max} of times that the Master_Key can be used (which actually translates to how many different nonces can be used, but since a different nonce should be used for every message this is the same as the number of different messages, but messages can be long). Our calculation is that $q_{max} \sim 2^{47}$ is an appropriate recommendation in order to preserve the 2^{-32} security margins recommended by NIST. This 2^{47} limit is based on a failure of indistinguishability of the generated nonce-based keys, unlike the deeper failure that AES-GCM would suffer by a nonce misuse. This number shows that the additional key derivation increases the number of allowed usages of a (master) key for AES-GCM-SIV, compared to the CCS version that was limited to $\sim 2^{32}$ usages (and the security margins). We are working on a way to increase q_{max} even further, and this will be announced in the upcoming paper.

From a practical viewpoint, we comment that if a user sends, say, 1 million messages per second, then a key replacement after 2^{47} usages would be necessary after ~4 years.

Since AES-GCM-SIV is designed for cases where a user needs to encrypt multiple messages, but nonce uniqueness may be a concern, we suggest to compare q_{max} to the number of times that a key can be used with AES-GCM with a randomly selected 96-bit IV (i.e., 2^{32}).

With this limit on q_{max}, we conclude that your mentioned attacks do not apply. We understand that you agree with this analysis.

The last question that remains is regarding the current KDF that uses a hierarchy of keys, and this cascade generates some relation between the derived keys. We are not extremely concerned with 2^{128} work to extract additional keys after one key is compromised (with no cryptographic method know so far). However, we agree that for 256-bit keys, this should not be possible. This brings us to the question about the KDF. In fact, we have been contemplating amongst ourselves about a better KDF that not only has better indistinguishability bounds, but is also faster than the current one. Also, some CFRG member have very recently asked about this. Therefore, we do intend to replace the KDF, as follows. AES-GCM-SIV will receive a 96-bit nonce. The KDF will compute AES (Master_Key, IV || IntToString32 (j)) for j=0, …, 3 (128-bit key) or j=0, …, 5 (256-bit key). From each of these 4 (or 6) generated blocks, 64 bits will be discarded, and then pairs will be combined into 2 (3) 128-bit values. These will be used as the Record_Hash_Key and the Record_Encryption_Key. The indistinguishability advantaged of this

KDF (assuming AES is a close approximation to a random permutation), is bounded by $6q/2^{96}$ where q is the number of times that the KDF was called with a single (master) key (see [1]). Now, for q different random nonces, the probability of an r-multi-collision (i.e., at least one value appears at least r times), and hence also on the per-nonce keys, can be bounded by $1/u^{r-1}$ * Binomial (q, r) (where $u=2^{96}$ here). Note that if for different nonces, there is a collision on the derived keys, this will not be a case of nonce misuse.

We can require a limited number of allowed nonce repetitions. Based on the above calculation, the probability that selecting the nonce (uniformly) at random will show r-multi-collisions (for a large r), is negligible, when staying with the bounds of q_{max}. Assume now that for the q_{max} (randomly) generated nonces, each value appears at most 4 times. Using some bound on the (intentional) number of messages encrypted by one selected nonce (and the lengths of the messages and AAD's), we can apply the security bounds that we have on GCM-SIV.

We plan to post the updated specification next week.

Thank you again, Shay and Yehuda

[1] S. Gilboa, S. Gueron, "The Advantage of Truncated Permutations", https://arxiv.org/abs/1610.02518 (submitted on 8 Oct 2016).

[2] K. Suzuki, D. Tonien, K. Kurosawa, K. Toyota, "Birthday Paradox for Multi-collisions", Proceedings of the 9th International Conference on Information Security and Cryptology, 29-40 (2006).

http://dl.acm.org/citation.cfm?id=2172962