An Update on SATURNIN

Anne Canteaut[1], Sébastien Duval[2], Gaëtan Leurent[1], María Naya-Plasencia[1], Léo Perrin[1], Thomas Pornin[3] and André Schrottenloher[1]

[1] Inria, France, {anne.canteaut,gaetan.leurent,maria.naya_plasencia,leo.perrin,andre.schrottenloher}@inria.fr
[2] UCL Crypto Group, Belgium, sebastien.pf.duval@gmail.com
[3] NCC Group, Canada, Thomas.pornin@nccgroup.com

1 SATURNIN : The post-quantum LWC candidate

The aim of SATURNIN is to provide a lightweight suite of algorithms that performs well on small devices and that provides a high security, even against quantum adversaries. SATURNIN has the particularity of being the only candidate designed to be secure against quantum attacks. In particular, our security claims are the only ones that include resistance against superposition-query attacks. Even though the direct applicability of attacks in this model might be discussed, we believe that this is the only way of ensuring post-quantum security in intermediate scenarios with some level of quantum interaction between the attacker and the encryption device. There seems to be an agreement in the community that this is the right way to define post-quantum security in symmetric primitives.

Since the NIST is currently running a post-quantum cryptography standardization effort in prevision of future quantum attackers, we believe it would be important to have in parallel a post-quantum symmetric primitive as well.

2 Update of our Submission

In our original submission, we did not provide a one-pass AEAD because it is hard to achieve security against superposition queries. We have been working on this problem since the original submission, and we are now finalizing a proposal for a one-pass mode, QCB [BBC+20]. We would like to include in our submission an additional AEAD instance that combines QCB with SATURNIN_{16}. This mode offers improved performances compared to SATURNIN-CTR-Cascade, and can be parallelized, while offering a tighter proof of quantum security. We give a high-level overview of QCB in Section 5.

Should the need arise, we also propose a tweak of SATURNIN-Short that decreases the nonce length in order to accommodate longer messages.

3 On security

3.1 Inheriting from AES knowledge

To the best of our knowledge, no external cryptanalysis improving upon our submission results has been published. There is a very good reason for this. The AES block cipher [AES01] is arguably the best studied symmetric primitive. We purposefully built SATURNIN so as to inherit all this knowledge, obtained thanks to the AES during the last 18 years. Therefore, the bounds against most well known attacks directly apply to

Saturnin. It seems really hard, also based on our own experience, to improve these bounds without making new (surprising!) discoveries about the AES itself.

3.2 Launching a challenge

The main difference with respect to AES is our much simplified key schedule. In our submission, in order to claim classical related-key security[1], we proposed to use Saturnin_{16}, that has 16 super-rounds instead of 10, but this was not the main submission, and no third-party cryptanalysis on related-key security has been published so far.

We plan to launch in the next months a challenge to encourage cryptanalysts to study the related-key security of Saturnin: what is the highest number of super-rounds that can be broken? From our preliminary analysis we expect this to be far from 16, but we hope that a challenge will motivate third-party cryptanalysis. This will improve our knowledge of the security of the hash function Saturnin-Hash and of the new mode Saturnin-QCB, where the number of rounds was chosen to offer security against classical related-key attacks.

4 On Performance

The original submission of Saturnin already included three extra implementations:

- in portable C;
- in assembly for the ARM Cortex M3;
- in assembly for the ARM Cortex M4.

All three implementations are *constant-time*, i.e. naturally immune to timing-based side-channel attacks (including cache attacks and similar attacks on systems where memory access timing may depend on the target address). The ARM Cortex M4 implementation can encrypt or decrypt data with the AEAD mode Saturnin-CTR-Cascade at a speed of 144 cpb. The equivalent NIST standard (AES/GCM) on an ARM Cortex M4 with a constant-time implementation can be estimated to the following cost:

- AES-128 in CTR mode: 101 cpb (using the speed reported in [SS16], for an assembly-optimized bitslice implementation).
- GHASH: 60 cpb (obtained with a custom assembly implementation using integer multiplications, following the method used in the "`ghash_ctmul`" implementation in BearSSL[Por]).

Therefore, the performance of a constant-time AES/GCM implementation on an ARM Cortex M4 should be about 161 cpb, which is slower than Saturnin-CTR-Cascade on the same hardware. We may also note that Saturnin-CTR-Cascade uses larger keys and blocks (256 bits instead of 128 bits) and, contrary to AES/GCM, does not suffer from a relatively high probability of nonce collision.

We also remark that Saturnin-Hash fairs fairly well on Rhys Weatherley's micro-controller benchmarks,[2] and that, for short messages, Saturnin-Short is highly competitive.

[1] As all known block ciphers, Saturnin cannot achieve related-key security in the very strong model of an adversary that can ask for a superposition of related keys (with chosen differences with a secret key), as shown by [RS13].

[2] https://rweather.github.io/lightweight-crypto/index.html

5 Short Description of Saturnin-QCB

The QCB mode is an AEAD based on a Tweakable Block Cipher, similar to the TAE mode [LRW02, LRW11] and to ΘCB [Rog04, KR11]. Full details of the mode and its security proof will be given in a separate paper [BBC+20], but we give the high-level ideas below.

Following TAE and ΘCB, each block of plaintext is encrypted by a call to the TBC, with a tweak that includes a domain separator, a nonce, and the block number. The tag is computed by encrypting a checksum of the message, with a tweak that includes a domain separator, a nonce, and the message length. The definition ensures that, if nonces are not reused, then the TBC is never called twice with the same tweak. For simplicity, the message is always padded with a "01*" padding, so the ciphertext can be up to 512 bits longer than the plaintext.

We define the TBC as $\tilde{E}_{k,t,D}(x) = E^D_{k\oplus t}(x)$, where E^D denotes Saturnin$^D_{16}$ (the version of Saturnin with 16 super-rounds, used with 4-bit domain separator D), and t is a 256-bit tweak value (thus the tweak space is of 260 bits in total). In the ideal-cipher model, we can prove the indistinguishability and unforgeability of QCB under quantum chosen-plaintext attacks as defined in the specification of Saturnin. The proof assumes that nonces are not controlled by the adversary and not reused [BBC+20]. Saturnin-QCB is represented on Figure 1.

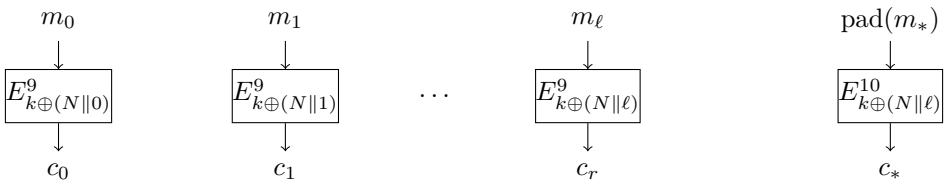

Figure 1: Saturnin-QCB, encryption.

Comparison with Saturnin-CTR-Cascade. Saturnin-QCB is of rate one: each block of plaintext requires approximately one call to the TBC $E^D_{k\oplus t}(x)$. Thus, we perform 16 super-rounds for each message block instead of $10 + 10 = 20$ super-rounds in Saturnin-CTR-Cascade, reducing the average time by a factor 0.8. Note that for each AD block, we now perform 16 super-rounds instead of 10. Moreover, Saturnin-QCB is easily parallelized, which can be useful when a powerful server communicates with a large number of lightweight devices.

Contrary to Saturnin-CTR-Cascade, each encryption in Saturnin-QCB requires rekeying. However, the key-schedule of Saturnin is linear and the rekeying requires only to change a value XORed in place to the key.

References

[AES01] Advanced Encryption Standard (AES). National Institute of Standards and Technology (NIST), FIPS PUB 197, U.S. Department of Commerce, November 2001.

[BBC+20] Ritam Bhaumik, Xavier Bonnetain, André Chailloux, Gaëtan Leurent, María Naya-Plasencia, Yannick Seurin, and André Schrottenloher. QCB: Efficient quantum-secure authenticated encryption, 2020. To appear.

[KR11] Ted Krovetz and Phillip Rogaway. The software performance of authenticated-encryption modes. In Antoine Joux, editor, *FSE 2011*, volume 6733 of *LNCS*, pages 306–327. Springer, Heidelberg, February 2011.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer, Heidelberg, August 2002.

[LRW11] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable block ciphers. *Journal of Cryptology*, 24(3):588–613, July 2011.

[Por] Thomas Pornin. BearSSL. https://www.bearssl.org/.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In Pil Joong Lee, editor, *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer, Heidelberg, December 2004.

[RS13] Martin Roetteler and Rainer Steinwandt. A note on quantum related-key attacks. Cryptology ePrint Archive, Report 2013/378, 2013. http://eprint.iacr.org/2013/378.

[SS16] Peter Schwabe and Ko Stoffelen. All the AES you need on Cortex-M3 and M4. In Roberto Avanzi and Howard M. Heys, editors, *SAC 2016*, volume 10532 of *LNCS*, pages 180–194. Springer, Heidelberg, August 2016.