Updates on SpoC

Riham AlTawy[1], Guang Gong[2], Morgan He[2], Ashwin Jha[3], Kalikinkar Mandal[4], Mridul Nandi[3], and Raghvendra Rohit[5]

[1] Department of Electrical and Computer Engineering, University of Victoria, Victoria, Canada
[2] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
[3] Indian Statistical Institute, Kolkata, India
[4] Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
[5] University of Rennes, CNRS, IRISA, France

Abstract. SpoC is a family of lightweight authenticated encryption algorithm and a round 2 candidate of the NIST lightweight cryptography standardization competition [5]. In this article, we report updates on SpoC since its selection as a round 2 candidate. In particular, we report security proof of SpoC mode, new third party cryptanalysis and implementation results. We also highlight the target applications and use-cases of SpoC along with our plans of future tweaks.

1 Security Analysis

SpoC is a permutation based mode of operation for authenticated encryption with associated data functionality [3]. It is a variant of sponge-AEAD where data blocks are absorbed through the capacity part instead of rate. SpoC has two members: SpoC-64 and SpoC-128, with core primitives as the sLiSCP-light-192 and sLiSCP-light-256 permutations, respectively. For both variants of SpoC with 18 rounds of sLiSCP-light, we claim that they are secure against attacks targeting confidentiality and integrity while the data limit of 2^{50} bytes and time limit of 2^{112} offline permutation evaluations is not violated in a nonce-respecting setting [3, Table 3.1]. In the following, we give details on the security proof of SpoC mode and third party cryptanalysis results.

1.1 Security proof of SpoC mode

A formal security proof of SpoC has already been published in IACR ToSC Vol. 2020 Issue 2 [8], and the full version of the same paper is available in [7]. We refer the readers to [7, Section 7.2] for a detailed discussion and formal security results on SpoC. Here, we briefly discuss the implications of the formal security bounds proved in [8, 7] on the security claims made in the NIST submission [3].

Let D denote the total number of bytes in all queries to the SpoC mode of operation, and T denote the number of direct invocations of the underlying permutation. Traditionally, D is called the data complexity or online query limit and T is called the time complexity or offline query limit.

In [7, Section 7.2.1], Chakraborty et al. state that the attack advantage of any adversary $\mathcal{A}$ against SpoC-64 and SpoC-128 are bounded as follows:

$$\begin{aligned} \mathsf{Adv}^{\mathsf{aead}}_{\text{SpoC-64}}(\mathcal{A}) &\leq \frac{D}{2^{66}} + \frac{7T}{2^{121}} + \frac{3DT}{2^{192}}, \\ \mathsf{Adv}^{\mathsf{aead}}_{\text{SpoC-128}}(\mathcal{A}) &\leq \frac{9D}{2^{125}} + \frac{7T}{2^{120}} + \frac{7DT}{2^{259}}. \end{aligned} \tag{1}$$

From the above relations it is evident that SpoC-64 is secure while:

$$D < 2^{66} \text{ and } T < 2^{118.19} \text{ and } DT < 2^{190.41},$$

and SpoC-128 is secure while:

$$D < 2^{121.83} \text{ and } T < 2^{117.19} \text{ and } DT < 2^{256.19}.$$

Clearly, the above data and time limits justify our security claims (secure while $D < 2^{50}$ and $T < 2^{112}$), as given in [3, Table 3.1].

1.2 Third-party cryptanalysis results

Distinguishers on round-reduced sLiSCP-light permutation [9]. Hosoyamada et al. have analyzed sLiSCP-light against limited-birthday distinguishers (LBD) which can cover up to 16 out of 18 rounds of the permutation. The LBDs complexities are 2^{113} (time) and $2^{37.7}$ (memory) for sLiSCP-light-192, and $2^{154.6}$ (time) and $2^{48.3}$ (memory) for sLiSCP-light-256. Although this work improves the existing distinguishers on both variants of sLiSCP-light [4, 3] by 2 rounds, they do not pose a direct threat to SpoC. This is because the available degrees of freedom are 64 bits and 128 bits per call of the permutation, i.e., rate values are 64 and 128 bits for SpoC-64 and SpoC-128, respectively. On the other hand, the LBDs require much higher time complexities.

Cryptanalysis results on SpoC [10]. Kravela et al. have analyzed SpoC against tag forgery, message recovery and key-recovery attacks. We briefly discuss their analysis and implications of their results on the security of SpoC. In the following, we consider a 128-bit nonce as $N = N_0 \| N_1$ where N_0 and N_1 are 64-bit words.

1. **Tag forgery on 6 rounds of SpoC-128:** The idea is to inject an input difference at both key and nonce values, and then cancel the output state difference via domain separators. As such, the attack model is based on related-key and related-nonce setting, and finding a corresponding differential trail with high probability. The authors found a differential trail with probability $2^{-106.14}$ and accordingly presented a tag forgery attack on 6 out of 18 rounds of SpoC-128 with $2^{106.14}$ data and $2^{107.14}$ time.

   We emphasize that a similar analysis of differential trails has been investigated in detail by us in [4, 14]. The only difference is that we did not consider the related-key and related-nonce scenario as this is irrelevant in a single-key setting where the adversary can only inject differences in nonce, associated data and/or plaintexts. Although the presented attack do not pose any threat to SpoC-128, it shows that in the single-key setting, 6-round SpoC-128 provides higher security guarantees against differential cryptanalysis than the related-key and related-nonce based differential attacks.

2. **Tag forgery on 7 rounds of SpoC-64:** The main idea is similar to the previous attack with the only exception that the model is based on the single-key scenario and the input difference is injected at N_1. The attack has complexities of $2^{108.2}$ data and $2^{109.2}$ time which violates the prescribed data limit of 2^{50} bytes.

3. **Message recovery on 9 rounds of SpoC-64:** In this attack, the authors consider the differentials for which the state collides. More precisely, they insert input difference in the key and N_0 and then cancel the output difference at capacity part with the corresponding difference in N_1. The attack requires $2^{110.84}$ data and $2^{109.84}$ time. Note that the number of attacked rounds is 2 more than the previous case as the related-key setting gives additional degrees of freedom.

4. **Key recovery attack on SpoC-64:** The authors claim a key recovery attack on full round SpoC-64 with complexities of 2^{67} data and 2^{110} time, which has a success probability of 2^{-15}. We emphasize that their attack is neither surprising nor does it invalidate our security claims. This is justified in the following discussion:
   - **Attack idea:** Their attack strategy involves first guessing the state and then extracting the key via inverse call of permutation. *This scenario is already considered in our submission document (cf. Section 4.1.1 of [3]).* As noted in [3, Section 4.1.1], the attack advantage of such attacks follow the relation $DT = 2^b$ (also present in the security bounds), where D, T, and b, denote the data, time (number of offline permutation evaluations) and state size, respectively. Indeed, this is one of the strategies that contributes a $\frac{DT}{2^b}$ term in the security bound of SpoC (see Eq. (1)).
   - **Attack complexity:** The authors choose $D = 2^{67}$, $T = 2^{110}$, and $b = 192$, which naturally translates to a success probability of 2^{-15} (using $DT = 2^b$ curve). However, *this violates our prescribed data limit of* 2^{50} *bytes by a significant factor of* 2^{17}.

– **Multi-key setting:** The authors justify their use of 2^{67} data blocks by stating that they aim to query SpoC in multi-key (or multi-user) setting, i.e., multiple instances of SpoC with independent keys. We emphasize that *our security claims are given in the single-key setting, as is the requirement of NIST LwC project (cf. Section 3.1 [1]).*

  However, for the sake of discussion, we point out that even under the multi-user setting, their attack follows from the single-key to multi-key security degradation (see [6, 11] for more details), i.e., it follows the relation $\mu DT = 2^b$, where μ denotes the number of users (for brevity, we assume identical data complexity for all users), i.e., if we fix $D = 2^{50}$, and $T = 2^{112}$, then the *attack requires $D = 2^{50}$ bytes of data per user, from $\mu = 2^{17}$ users to get a success probability of 2^{-15} in recovering the key of a single user, which is not that significant given the large number of users attacked.*

5. **Observations on constants.** The authors further mentioned that *"the bits rc_1^{8+n} and rc_1^n are equal ... "*. We would like to point out this is a property of a primitive polynomial of LFSR and as such **it does not add anything new to the security**. The constants were chosen so that: 1) they can be generated by a single LFSR, and 2) **the pair of round constants and step constants at each round and step are distinct**. See [3, Section 4.2.4] for further analysis.

Summary. The known cryptanalysis results on SpoC and its underlying permutation sLiSCP-light do not pose any threats to our security claims. Based on the best known third-party results (which are in fact in the related-key setting), the security margin, i.e., $1 - \frac{\#\text{ attacked rounds}}{\#\text{ total rounds}}$ of SpoC-64 and SpoC-128 are at least 50% and 66%, respectively.

2 Implementation Results

2.1 Hardware

Implementation and benchmarking by Rezvani et al. [13]. The hardware implementation of SpoC-64 and its performance across different hardware platforms has been extensively analyzed in a recent work of Rezvani et al. Below, we highlight few results from the same work.

– SpoC-64 has the **smallest area** in Artix-7, Spartan-6 and Cyclone-V when compared to other round 2 candidates such as Ascon, Comet, GIFT-COFB and Sparkle.

– SpoC-64 achieves the **maximum frequencey** in Artix-7 and Cyclone-V, and has the **lowest power** consumption among the aforementioned round 2 candidates.

– SpoC, Comet and GIFT-COFB achieves the **lowest power gradient**, i.e., lowest increase in power consumption with increasing frequency.

Implementation by SpoC team. We have provided the implementation of SpoC in two technologies: ST Micro 65 nm and IBM 130 nm. The codes are available at https://uwaterloo.ca/communications-security-lab/lwc/spoc. Note that our codes currently do not follow the LWC API framework of [12].

A threshold implementation of SpoC for protection against side-channel attacks will be available shortly.

2.2 Software

Implementation and benchmarking by Weatherley [16]. SpoC is included in Rhy's Weatherley software benchmarking framework where the performance of round 2 candidates is evaluated on 8-bit and 32-bit microcontrollers. Further, a protected implementation of SpoC for up to 6 shares has been incorporated in the same framework. For the details on the performance, the reader is referred to [16].

Bitslice implementation by SpoC team. We implemented SpoC in the bit-slice fashion using SIMD instruction sets which provides resistance against cache-timing attacks. Our implementation allows to execute multiple independent SpoC instances in parallel. We consider SSE and AVX instruction sets in Intel processors where the SSE and AVX instruction sets, support 128-bit and 256-bit SIMD registers, known as XMM and YMM, respectively.

We implemented the sLiSCP-light permutation and both modes in C using SSE2 and AVX2 instruction sets and measured their performances on two different Intel processors: Coffee Lake and Whiskey Lake. The codes on Coffee Lake and Whiskey Lake were compiled using `gcc 7.5.0` and `gcc 9.2.1`, respectively on 64-bit machines with the compiler flags `-O2 -funroll-all-loops -march=native`. For both implementations, we evaluated four parallel instances and compute the throughput of the permutation and its modes. Table 1 presents the performance results in cycles per byte (cbp) for both the permutation and entire SpoC where encryption is done for a 1024-bit message and a 128-bit associated data, including the tag computation. The codes are publicly available at https://uwaterloo.ca/communications-security-lab/lwc/spoc.

Table 1: Benchmarking the results for the sLiSCP-light permutation and SpoC.

Primitive	**Speed** [cpb]	**Instruction Set**	**CPU Name Spec.**
sLiSCP-light-192	15.78	SSE2	Coffee Lake
	8.80	AVX2	`Intel i7-9700`
	13.60	SSE2	Whiskey Lake
	7.80	AVX2	`Intel i5-8265U`
sLiSCP-light-256	11.72	SSE2	Coffee Lake
	7.32	AVX2	`Intel i7-9700`
	9.12	SSE2	Whiskey Lake
	6.20	AVX2	`Intel i5-8265U`
SpoC-64	64.65	SSE2	Coffee Lake
	41.43	AVX2	`Intel i7-9700`
	50.16	SSE2	Whiskey Lake
	30.44	AVX2	`Intel i5-8265U`
SpoC-128	36.10	SSE2	Coffee Lake
	22.40	AVX2	`Intel i7-9700`
	25.88	SSE2	Whiskey Lake
	14.58	AVX2	`Intel i5-8265U`

3 Features

Applications and use cases. SpoC is primarily a hardware-oriented lightweight authenticated cipher and designed to achieve low *area, power and energy* which is crucial for RFID and sensor network applications. A concrete use case of SpoC in device-to-device communication for 5G Internet of Things networks has been investigated in [15].

Comparison with AES-GCM. SpoC outperforms AES-GCM in area, frequency and power as shown in [2, 13]. For the sake of brevity, we include some performance figures from those papers: [2, Table 4] and [13, Table 4].

- In Artix-7, SpoC-64 consumes 1344 LUTs while the area of AES-GCM is 1532 LUTs. Similarly, the area of SpoC-64 is less than AES-GCM in Spartan-6 and Cyclone-V.
- SpoC-64 achieves a higher frequency than AES-GCM in Artix-7 (268 vs 222 MHz) and Cyclone-V (224.4 vs 165.9 MHz).

- In Artix-7, at 50 MHz, the average power values of SpoC-64 and AES-GCM are 34.7 and 35.9 mW, respectively. Further, SpoC-64 has lower $dP/dFreq$ (0.15 vs 0.18) value while energy values of both ciphers are comparable.

4 Proposed Tweaks

Currently, we do not have any plans to tweak the design of SpoC.

References

[1] Submission requirements and evaluation criteria for the lightweight cryptography standardization process. NIST Lightweight Cryptography Standardization Process.

[2] A. Abdulgadir, W. Diehl, and J. Kaps. An open-source platform for evaluation of hardware implementations of lightweight authenticated ciphers. In *2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, pages 1–5, 2019.

[3] Riham AlTawy, Guang Gong, Morgan He, Ashwin Jha, Kalikinkar Mandal, Mridul Nandi, and Raghvendra Rohit. SpoC: Submission to NIST-LWC standardization process (round 2), 2019. https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/spoc-spec-round2.pdf.

[4] Riham Altawy, Raghvendra Rohit, Morgan He, Kalikinkar Mandal, Gangqiang Yang, and Guang Gong. Sliscp-light: Towards hardware optimized sponge-specific cryptographic permutations. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(4):81, 2018.

[5] Lawrence Bassham, Cagdas Calik, Donghoon Chang, Jinkeon Kang, Kerry McKay, and Meltem Sonmez Turan. Lightweight cryptography: Round 2 candidates, 2019. https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates.

[6] Mihir Bellare and Björn Tackmann. The multi-user security of authenticated encryption: AES-GCM in TLS 1.3. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 247–276. Springer, 2016.

[7] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of sponge-type authenticated encryption modes. *IACR Cryptol. ePrint Arch.*, 2019:1475, 2019.

[8] Bishwajit Chakraborty, Ashwin Jha, and Mridul Nandi. On the security of sponge-type authenticated encryption modes. *IACR Transactions on Symmetric Cryptology*, pages 93–119, 2020.

[9] Akinori Hosoyamada, María Naya-Plasencia, and Yu Sasaki. Improved attacks on sliscp permutation and tight bound of limited birthday distinguishers. Cryptology ePrint Archive, Report 2020/1089, 2020. https://eprint.iacr.org/2020/1089.

[10] Liliya Kraleva, Raluca Posteuca, and Vincent Rijmen. Cryptanalysis of the permutation based algorithm spoc. Cryptology ePrint Archive, Report 2020/1072, 2020. https://eprint.iacr.org/2020/1072.

[11] Atul Luykx, Bart Mennink, and Kenneth G. Paterson. Analyzing multi-key security degradation. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2017.

[12] Ekawat Homsirikamol William Diehl Jens-Peter Kaps Michael Tempelmeier, Farnoud Farahmand and Kris Gaj. Implementer's guide to hardware implementations compliant with the hardware api for lightweight cryptography, v1.0.1 (nov 2019). https://cryptography.gmu.edu/athena/LWC/LWC_HW_Implementers_Guide.pdf.

[13] Behnaz Rezvani and William Diehl. Hardware implementations of nist lightweight cryptographic candidates: A first look. *IACR Cryptol. ePrint Arch.*, 2019:824, 2019.

[14] Rohit, Raghvendra. Design and cryptanalysis of lightweight symmetric key primitives, 2020.

[15] Byoungjin Seok, Jose Costa Sapalo Sicato, Tcydenova Erzhena, Canshou Xuan, Yi Pan, and Jong Hyuk Park. Secure d2d communication for 5g iot network based on lightweight cryptography. *Applied Sciences*, 10(1):217, 2020.

[16] Rhys Weatherley. Lightweight cryptography primitives. https://rweather.github.io/lightweight-crypto/.