Status Update on Ascon v1.2

Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer

September 18, 2020

1 New software and hardware implementations

Since Ascon was first published in 2014, there are already many hardware and software implementations. An overview can be found in the design document [6, Section 7]. This includes implementations that protect against side-channel attacks.

1.1 Software implementations

In the past year since the design document has been updated, several new implementations of Ascon have been published. We have updated our own suite of reference and optimized implementations at https://github.com/ascon/ascon-c/. The update includes implementations optimized for speed and size as well as (soon) implementations to protect against side-channel attacks.

The performance of many of these implementations has been benchmarked by https://bench.cr.yp.to/, https://lwc.las3.de/ and https://rweather.github.io/lightweight-crypto/. These benchmarks show an excellent performance of Ascon for a wide range of different platforms when compared versus the other schemes participating in the NIST Lightweight Crypto Standardization process. Compared to the current NIST standards AES-CCM [9] and AES-GCM [10], we expect a similar picture for platforms where no AES instruction is available, especially when considering short messages.

The second benchmarking effort also analyzes the size of optimized Ascon implementations and the third analyzes masked implementations. These results show that Ascon can be implemented at a very low size with only minimal performance penalties, and at a low overhead if protection against side-channel attacks matters.

Besides our own suite, there are several other software implementations. For instance, Campos et al. [4] provide, amongst others, implementations for Ascon on RISC-V, complete with benchmarks of various submissions to the NIST Lightweight Crypto Standardization process. Weatherley [12] published optimized Ascon implementations which perform very well on a wide range of embedded platforms.

1.2 Implementations that protect against side channel attacks

Ascon has been designed with protected implementations in mind. First of all, Ascon has a small state size and can be implemented at a low size. Ascon is also very flexible and several hardware and software options can be used to protect Ascon against side-channel attacks. For example, software shares can be stored and computed in a rotated form with limited performance impact since most operations in Ascon are rotation-symmetric. This reduces the side-channel leakage on real devices.

Furthermore, the non-linear Ascon S-box can be efficiently masked with fewer instructions and less randomness using the Toffoli gate, as discussed in [5]. Preliminary results of masked software implementations show that using the Toffoli gate, a 2-share implementation results in only a 2x performance decrease, while a 3-share implementation results in a 4x performance decrease on high-end 64-bit platforms.

Additionally, Ascon provides the option to use leveled implementations proposed by Adomnicai et al. [1] and recently discussed by Bellizia et al. [3]. In leveled implementations, a higher protection order is used for the initialization and finalization than for other parts of an algorithm. In particular, in scenarios where the number of decryption failures or queries can be limited or plaintext confidentiality is not critical, such implementations might be of interest. For long messages, such implementations can achieve performances similar to that of unprotected implementations.

1.3 Hardware implementations

A RISC-V instruction extension for Ascon's permutation has been developed [11]. This hardware accelerator can be realized needing only about 4.7 kGE of area, while having a performance of 4.2 cycles per byte for 64 byte plaintexts, and 2.2 cycles per byte for 1536 byte plaintexts for Ascon-128.

For hardware implementations, Ascon provides an excellent trade-off when considering throughput versus area, especially when considering implementations protected against side-channel attacks, see for instance Table 1.

Links to various hardware implementations of Ascon, including masked implementations, can be found at https://ascon.iaik.tugraz.at/implementations.

2 On Ascon's security

Ascon is based on the duplex/sponge constructions that are well understood and several proofs considering these constructions have been published. The relevant literature is already given in the Ascon submission document [6].

However, there is new work considering Ascon's properties with respect to side-channel attacks [3]. In particular, Ascon is shown to have similar properties as Spook [2], which can enable more efficient protection against side-channel attacks.

Table 1: DOM implementations for various protection orders [7, 8].

Protection Order	Pipelined		Parallel	
	[kGE]	[Mbps]	[kGE]	[Mbps]
1	10.86	108	28.89	2246
2	16.19	108	53.00	1896
3	21.59	110	81.21	1903
4	27.13	71	118.27	1786
5	32.76	95	161.87	1868
	...			
13	81.20	70	726.00	1833
14	87.75	71	828.19	1439
15	94.24	50	926.34	1480

With respect of third-party cryptanalysis, substantial work has already been published; we provide an overview in the Ascon submission document [6, Section 6]. New results that are not yet covered in the submission document include:

Collisions for Ascon-Hash reduced to 2 rounds with complexity 2^{125}:

🗎 R. Zong, X. Dong, and X. Wang. "Collision Attacks on Round-Reduced Gimli-Hash/Ascon-Xof/Ascon-Hash". Cryptology ePrint Archive, Report 2019/1115. https://eprint.iacr.org/2019/1115. 2019.

Improved 4-round differential-linear analysis and subspace trails:

🗎 C. Tezcan. "Distinguishers for Reduced Round Ascon, DryGASCON, and Shamash Permutations". NIST Lightweight Cryptography Workshop. 2019. URL: https://csrc.nist.gov/Presentations/2019/distinguishers-for-reduced-round-ascon-drygascon-a.

Integral distinguishers for the round-reduced inverse Ascon permutation:

🗎 H. Yan, X. Lai, L. Wang, Y. Yu, and Y. Xing. "New zero-sum distinguishers on full 24-round Keccak-f using the division property". In: IET Information Security 13.5 (2019), pp. 469–478.

All above-mentioned observations do not provide any threat for the security of Ascon. Instead, these new results confirm that all members of the Ascon family have a comfortable security margin.

3 Target applications and use cases for Ascon

Ascon follows a very balanced approach providing excellent performance/size trade-offs for a wide variety of software platforms and also for dedicated hardware designs. Furthermore, Ascon can keep its excellent performance even for short messages. In addition, Ascon has been designed with robustness and implementation attacks in mind. Hence, it allows for masking with a very low overhead [5, 8] and even leveled implementations [3]. Moreover, even if an attacker somehow manages to recover an internal state during data processing (e.g., due to side-channel attacks), this does not directly lead to the recovery of the secret key or to constructing trivial forgeries. These properties of the mode set Ascon apart from many other lightweight designs.

Taking all into account, Ascon is not only highly suited for scenarios where lightweight devices communicate with lightweight devices, but also for scenarios where many lightweight devices communicate with high-end devices (e.g., a back-end server), a typical use case in many applications including the Internet of Things (IoT). This is especially true in scenarios where protection against side-channel attacks is needed.

4 Planned tweak proposals

We do not plan any tweaks for Ascon.

Acknowledgments. The authors would like to thank all researchers contributing to the design, analysis and implementation of Ascon. In particular, we want to thank Hannes Gross and Robert Primas for all their support and various implementations of Ascon.

Part of this work has been supported by the Austrian Science Fund (FWF): P26494-N15 and J 4277-N38.

References

[1] A. Adomnicai, J. J. Fournier, and L. Masson. “Masking the Lightweight Authenticated Ciphers ACORN and Ascon in Software”. Cryptology ePrint Archive, Report 2018/708. https://eprint.iacr.org/2018/708. 2018.

[2] D. Bellizia, F. Berti, O. Bronchain, G. Cassiers, S. Duval, C. Guo, G. Leander, G. Leurent, I. Levi, C. Momin, O. Pereira, T. Peters, F.-X. Standaert, B. Udvarhelyi, and F. Wiemer. “Spook: Sponge-Based Leakage-Resistant Authenticated Encryption with a Masked Tweakable Block Cipher”. In: IACR Transactions on Symmetric Cryptology 2020.S1 (June 2020), pp. 295–349. URL: https://tosc.iacr.org/index.php/ToSC/article/view/8623.

[3] D. Bellizia, O. Bronchain, G. Cassiers, V. Grosso, C. Guo, C. Momin, O. Pereira, T. Peters, and F.-X. Standaert. “Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography – A Practical Guide Through the Leakage-Resistance Jungle”. In: Advances in Cryptology – CRYPTO 2020. Vol. 12170. LNCS. Springer, 2020, pp. 369–400. URL: https://doi.org/10.1007/978-3-030-56784-2_13.

[4] F. Campos, L. Jellema, M. Lemmen, L. Müller, D. Sprenkels, and B. Viguier. "Assembly or Optimized C for Lightweight Cryptography on RISC-V?" Cryptology ePrint Archive, Report 2020/836. https://eprint.iacr.org/2020/836. 2020.

[5] J. Daemen, C. Dobraunig, M. Eichlseder, H. Gross, F. Mendel, and R. Primas. "Protecting against Statistical Ineffective Fault Attacks". In: IACR Transactions on Cryptographic Hardware and Embedded Systems 2020.3 (2020), pp. 508–543.

[6] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. "Ascon v1.2 (Submission to NIST)". NIST Round 2 Candidate, https://csrc.nist.gov/Projects/Lightweight-Cryptography/Round-2-Candidates. 2019.

[7] H. Groß and S. Mangard. "Reconciling d+1 Masking in Hardware and Software". In: Cryptographic Hardware and Embedded Systems - CHES 2017. Vol. 10529. LNCS. Springer, 2017, pp. 115–136. URL: https://doi.org/10.1007/978-3-319-66787-4%5C_6.

[8] H. Gross and S. Mangard. "A unified masking approach". In: Journal of Cryptographic Engineering 8.2 (2018), pp. 109–124.

[9] National Institute of Standards and Technology. "NIST Special Publication 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality". May 2004. URL: https://doi.org/10.6028/NIST.SP.800-38C.

[10] National Institute of Standards and Technology. "NIST Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC". Nov. 2007. URL: https://doi.org/10.6028/NIST.SP.800-38D.

[11] S. Steinegger and R. Primas. "A Fast and Compact Accelerator for Ascon and Friends". Cryptology ePrint Archive, Report 2020/1083. https://eprint.iacr.org/2020/1083. To appear at CARDIS. 2020.

[12] R. Weatherley. "Lightweight Cryptography Primitives". GitHub, https://github.com/rweather/lightweight-crypto. 2020.