Ascon v1.2 – Analysis of Security and Efficiency

Proposal for Presentation

Christoph Dobraunig[1], Maria Eichlseder[2], Florian Mendel[3] and Martin Schläffer[3]

[1] Radboud University, The Netherlands
[2] Graz University of Technology, Austria
[3] Infineon Technologies AG, Germany

https://ascon.iaik.tugraz.at

Abstract. Ascon has been selected as the first choice for resource-constrained environments by CAESAR. Ascon-128 and Ascon-128a were also submitted to NIST's call for lightweight cryptography. The submission to NIST has been complemented with hashing functionality based on the well-studied permutation of Ascon and the sponge construction. In this note, we will give an overview of the existing analysis of the Ascon cipher suite and discuss the performance of Ascon in both software and hardware.

Highlights. The Ascon suite supports authenticated encryption and hashing with the same lightweight permutation. It achieves high security and robustness in practice with a very low area footprint in hardware, while providing good performance in both software and hardware implementations, in particular for short messages. Cryptographic primitives that operate efficiently on resource-constrained devices, on modern high-end systems, and also in the area between these two extremes are of rising importance. A typical example of such environments is the Internet of Things (IoT), where a large number of very constrained devices need to communicate efficiently with high-performance back-end servers. In the following, we discuss the security and performance of Ascon in more detail and show why the cipher suite is a perfect fit for such applications.

Security. Ascon-128 and Ascon-128a have been selected as the *primary choice* for lightweight authenticated encryption in the final portfolio of the CAESAR competition after five years of evaluation. During this competition, Ascon and its permutation have undergone a thorough public analysis. So far, this has resulted in numerous publications that give insight into the security of Ascon and in a total of almost 100 publications that discuss Ascon more generally. All existing analysis confirms a comfortable security margin for Ascon. An overview of the best-known analysis of the authenticated encryption schemes is summarized in Table 1.

Table 1: Best known analysis of Ascon (⊘ = misuse).

Type	Target	Rounds	Time	Method	Reference
Key recovery	Ascon initialization	7 / 12	2^{104}	Cube-like	[LDW17]
	Ascon initialization	5 / 12	2^{36}	Diff.-linear	[DEMS15]
	Ascon initialization	7 / 12	2^{97} ⊘	Cube-like	[LZWW17]
Forgery	Ascon finalization	4 / 12	2^{101}	Differential	[DEMS15]
	Ascon finalization	6 / 12	2^{33} ⊘	Cube tester	[LZWW17]
State recovery	Ascon-128a iteration	2 / 8	–	Sat-Solver	[Dwi+17]
	Ascon-128 iteration	5 / 6	2^{66} ⊘	Cube-like	[LZWW17]

Software Performance. The Ascon suite and in particular the underlying permutation have been designed for high security and robustness in practice with a low area footprint in hardware and software while providing good performance on various platforms. Ascon's permutation is defined on 64-bit words using only bitwise Boolean functions (AND, NOT, XOR) and rotations within words. Hence, the permutation lends itself well to fast bitsliced implementations on 64-bit platforms, while bit interleaving allows for fast bitsliced implementations on 32-, 16-, and 8-bit platforms. Moreover, implementing the Ascon permutation once is enough to get authenticated encryption as well as decryption with a very small overhead, since decryption does not require the inverse of the permutation (that is, Ascon is inverse-free). Together with Ascon-Hash and Ascon-Xof, it also provides hashing functionality using the same permutation. Thus, Ascon is an excellent choice in scenarios where lightweight devices carry out cryptographic operations. Due to the good performance in software, Ascon is a perfect fit in scenarios where lightweight devices communicate with high-end servers.

The simplicity of the design and the small state play also a crucial role in the efficiency of Ascon's authenticated encryption for short messages. For instance, if no associated data is present, Ascon-128 can encrypt plaintexts strictly smaller than 8 bytes and Ascon-128a can encrypt plaintexts strictly smaller than 16 bytes with just two calls to the permutation. Software performance results for short messages on several platforms are shown in Table 2a and Table 2b.

Table 2: Ascon-128 and Ascon-128a software performance in cycles per byte. Message length is the length of the encrypted plaintext in bytes with empty associated data.

(a) Ascon-128

Message Length	1	8	16	32	64	1536	long
AMD Ryzen 7 1700					14.5	8.8	8.6
Intel Xeon E5-2609 v4					17.3	10.8	10.5
Cortex-A53 (ARMv8)					18.3	11.3	11.0
Intel Core i5-6300U	367	58	35	23	17.6	11.9	11.4
Intel Core i5-4200U	521	81	49	32	23.9	16.2	15.8
Cortex-A15 (ARMv7)					69.8	36.2	34.6
Cortex-A7 (NEON)	2705	250	150	99	73.2	48.8	47.9
Cortex-A7 (ARMv7)	1871	292	175	115	86.6	58.3	57.2
ARM1176JZF-S (ARMv6)	2189	340	202	133	97.9	64.4	65.3

Results taken from eBACS [BL].

(b) Ascon-128a

Message Length	1	8	16	32	64	1536	long
AMD Ryzen 7 1700					12.0	6.0	5.7
Intel Xeon E5-2609 v4					14.1	7.3	6.9
Cortex-A57 (ARMv8)					15.1	7.6	7.3
Intel Core i5-6300U	365	47	31	19	13.5	8.0	7.8
Intel Core i5-4200U	519	67	44	27	18.8	11.0	10.6
Cortex-A15 (ARMv7)					60.3	25.3	23.8
Cortex-A7 (NEON)	2805	274	133	83	57.6	33.5	32.6
Cortex-A7 (ARMv7)	1911	255	161	102	71.3	42.3	41.2
ARM1176JZF-S (ARMv6)	2267	303	191	120	84.4	50.0	50.2

Results taken from eBACS [BL].

Hardware Implementations. As shown in Table 3, Ascon's small state and simple round function are well-suited for small hardware implementations, without compromising on the full security of 128 bits. Existing lightweight hardware implementations of Ascon's authenticated encryption modes are as small as 2.6 kGE [GWDE15]. The round-based implementations are smaller than 10 kGE and still offer a throughput of 4.9–7.3 Gbps, which is already sufficient to encrypt a Gigabit Ethernet connection.

Table 3: Ascon-128 hardware implementations taken from [GWDE15]

Design	Chip Area w/o interface	Chip Area w/ interface	Throughput	Power at 1 MHz	Energy
	[kGE]	[kGE]	[Mbps]	[µW]	[µJ/byte]
Ascon-fast					
1 round	7.08	7.95	5,524	43	33
2 rounds	10.61	11.48	8,425	72	27
3 rounds	14.26	15.13	10,407	102	25
6 rounds	24.93	25.80	**13,218**	184	**23**
Ascon64-bit	4.99	5.86	72	32	1,397
Ascon-x-low-area	**2.57**	**3.75**	14	**15**	5,706

Side-Channel Resistance. Moreover, Ascon can be implemented efficiently for platforms and applications where side-channel resistance is important. The very efficient bitsliced implementation of the S-boxes prevents cache-timing attacks, since no lookup tables are required. Furthermore, the low algebraic degree of the S-box facilitates both first- and higher-order protection using masking or sharing-based side-channel countermeasures. For instance, Gross, Wenger, Dobraunig, and Ehrenhöfer [GWDE15] provide threshold implementations of Ascon-128 as small as 7.97 kGE. Besides this, many other state-of-the-art masking approaches have been applied on Ascon, like UMA [GM17] and DOM [GMK16], even for a high protection order (see Table 4). Recent results also show that the S-box is well-suited for efficient countermeasures against statistical ineffective fault attacks [Dae+19].

Table 4: DOM implementations for various protection orders [GM17; GM18].

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

Robustness. Finally, we want to address the fact that ciphers are not used in an ideal world. Therefore, Ascon's authenticated encryption has been designed to provide robustness against certain implementation mistakes and attacks: For example, 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.

Acknowledgments. Part of this work has been supported by the Austrian Science Fund (FWF): P26494-N15 and J 4277-N38, by the European Union's Horizon 2020 research and innovation programme (H2020 ICT 644052: HECTOR), and by the Austrian Government (FFG/SFG COMET 836628: SeCoS and FIT-IT 835919: SePAG).

References

[BL] Daniel J. Bernstein and Tanja Lange, eds. *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. URL: https://bench.cr.yp.to (visited on 02/14/2019).

[Dae+19] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. *Protecting against Statistical Ineffective Fault Attacks*. IACR Cryptology ePrint Archive, Report 2019/536. 2019. URL: https://eprint.iacr.org/2019/536.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. "Cryptanalysis of Ascon". In: *CT-RSA 2015*. Ed. by Kaisa Nyberg. Vol. 9048. LNCS. Springer, 2015, pp. 371–387.

[Dwi+17] Ashutosh Dhar Dwivedi, Miloš Klouček, Pawel Morawiecki, Ivica Nikolič, Josef Pieprzyk, and Sebastian Wójtowicz. "SAT-based Cryptanalysis of Authenticated Ciphers from the CAESAR Competition". In: *SECRYPT ICETE 2017*. Ed. by Pierangela Samarati, Mohammad S. Obaidat, and Enrique Cabello. SciTePress, 2017, pp. 237–246.

[GM17] Hannes Gross and Stefan Mangard. "Reconciling d+1 Masking in Hardware and Software". In: *CHES 2017*. Ed. by Wieland Fischer and Naofumi Homma. Vol. 10529. LNCS. Springer, 2017, pp. 115–136.

[GM18] Hannes Gross and Stefan Mangard. "A unified masking approach". In: *Journal of Cryptographic Engineering* 8.2 (2018), pp. 109–124.

[GMK16] Hannes Gross, Stefan Mangard, and Thomas Korak. *Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order*. Cryptology ePrint Archive, Report 2016/486. 2016. URL: https://eprint.iacr.org/2016/486.

[GWDE15] Hannes Gross, Erich Wenger, Christoph Dobraunig, and Christoph Ehrenhöfer. "Suit up! – Made-to-Measure Hardware Implementations of Ascon". In: *DSD 2015*. IEEE Computer Society, 2015, pp. 645–652.

[LZWW17] Yanbin Li, Guoyan Zhang, Wei Wang, and Meiqin Wang. "Cryptanalysis of round-reduced ASCON". In: *SCIENCE CHINA Information Sciences* 60.3 (2017), p. 38102.

[LDW17] Zheng Li, Xiaoyang Dong, and Xiaoyun Wang. "Conditional Cube Attack on Round-Reduced ASCON". In: *IACR Transactions on Symmetric Cryptology* 2017.1 (2017), pp. 175–202. URL: https://github.com/lizhengcn/Ascon_test.