New Ascon Implementations

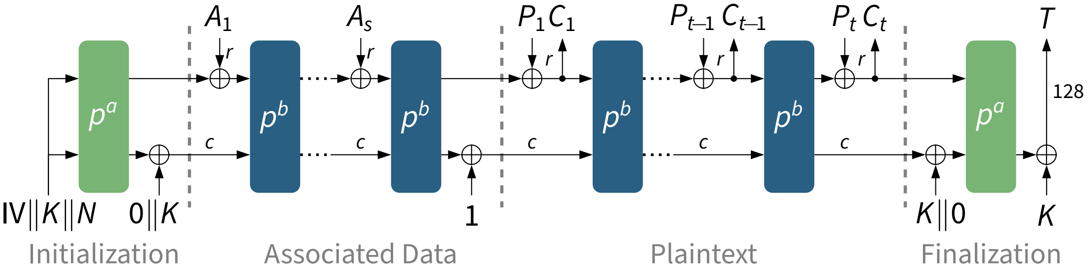
Christoph Dobraunig, Maria Eichlseder, Florian Mendel, **Robert Primas**, **Martin Schläffer**

NIST LWC Workshop 2022

Outline

Ascon Overview

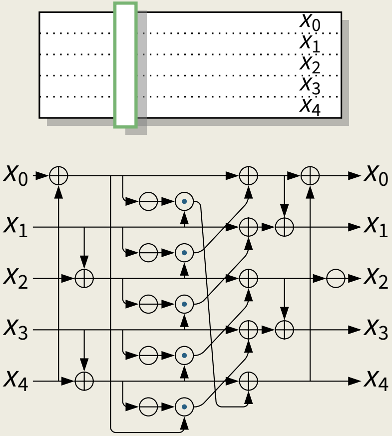
ASCON Mode for Authenticated Encryption



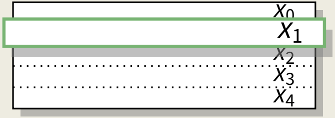
- Designed in 2014 [DEMS16], Journal of Cryptology in 2021 [DEMS21c]
- First choide for lightweight AEAD in CAESAR portfolio
- Extensive published cryptanalysis confirming its security margin
- Additional modes for Hash, XOF, MAC, PRF [DEMS21a; DEMS21b]

Ascon Permutation with $\{6, 8, 12\}$ Rounds

S-box layer



Linear layer



$$x_0 := x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$$

$$x_1 := x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$$

$$x_2 := x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$$

$$x_3 := x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$$

$$x_4 := x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$$

Ascon-128 vs Ascon-128a

- Ascon-128a: 33% more performance, more rounds, larger rate
- Same security, different trade-off (rate vs. number of rounds)
- Both scrutinized for 8 years in cryptographic competitions
- Most security analysis can be applied to both algorithms
- Similar security margin, no clear preference

Ascon Implementations

https://github.com/ascon/ascon-c (Ascon team)

- AEAD, Hash, XOF, MAC, PRF
- C: ref, speed/area optimized, combined
- ASM: esp32, armv6, armv6m, armv7m, rv32
- Masked C+ASM: 2-4 shares, leveled

https://github.com/rweather/ascon-suite (Rhys Weatherley)

- AEAD, Hash, HKDF, ISAP, KMAC, PBKDF2, PRNG, SIV, XOF
- 8/32/64-bit C, AVR, ARM, RISC-V, m68k, Xtensa (ESP32)
- Framework to generate C/ASM/masked implementations

Performance and Code Size

New Ascon Implementations

(Improvements in the Final Round)

- Fewer instructions for S-box [CJL+20]: -10%
- Improved 8-bit AVR [ascon-suite] (time/size): -11%/-44%
- Combined Ascon AEAD+Hash [ascon-c] (size): -17%
- Improved low-size [ascon-c] (size 128/128a/Hash): -7%/-30%/-20%
- Bit-interleaved interface [ascon-c] (time 128/128a/Hash): -17%/-23%/-5%
- ESP32 implementations [ascon-c][Bac22] (time/size): -66%/-64%
- RV32 implementations [ascon-c][Bac22] (RV32,RV32I,RV32B): New
- Masked ARMv6/RV32 [ascon-c] (leveled, 2-4 shares): New
- Ascon-Hasha, Ascon-Xofa [DEMS21b] (time): -33%
- Ascon-Mac, Ascon-Prf compared to Ascon-KMAC [DEMS21a] (time): -66%

Microcontroller Benchmarking

$\frac{ascon-nocrypt}{best-nocrypt}$ for primary submission @las3

Performance (time)		Code size (ROM)	
Uno:	1.34x	Uno:	3.22x
F1:	1.06x	F1:	1.62x
ESP:	1.92x	ESP:	1.31x
F7:	1.02x	F7:	1.10x
R5:	0.61x	R5:	1.07x

https://lwc.las3.de/ [2020/10/14]

Microcontroller Benchmarking

ASCON-128: best primary finalist in most categories

	Performance (time)	Code size (ROM)
Uno:	1.24x	1.59x
F1:	1.28x	0.80x
ESP:	0.58x	0.55x
F7:	0.89x	0.87x
R5:	0.55x	0.90x

https://lwc.las3.de/ [2022/05/05]

Microcontroller Benchmarking

ASCON-128: best primary finalist in most categories

	Performance (time)	Code size (ROM)
Uno:	1.24x	1.59x
F1:	1.28x	0.80x
ESP:	0.58x	0.55x
F7:	0.89x	0.87x
R5:	0.55x	0.90x

0-25% slower

https://lwc.las3.de/ [2022/05/05]

High-end Benchmarking

(Imagine Ascon hardware instructions)

AMD Ryzen 9:

- Ascon-128a: 5.1 c/b
- Ascon-128: 7.8 c/b
- Ascon-Hasha: 10.6 c/b*
- Ascon-Hash: 15.9 c/b

ARM Cortex-A72:

- Ascon-128a: 6.9 c/b
- Ascon-128: 10.4 c/b
- Ascon-Hasha: 13.5 c/b*
- Ascon-Hash: 20.2 c/b

https://bench.cr.yp.to/ [2022/05/03]

* estimated, not yet benchmarked

Implementation Techniques

Flexibility of Ascon Components

- Parallelism: S-box and linear layer support up to 5 ALUs
- Small state: 10 32-bit registers, 2 temporary, 1 for loop
- S-box: new description with fewer instructions
- Linear: 64-bit rotate or bit interleaving or funnel shift
- Modes: combine absorb, squeeze, insert (xor, read, write)
- Rate: loop for combined implementations (rate 64, 128)
- Short messages: only init and final needed

Ascon Hardware Extensions

- Fast, lightweight Ascon round instruction for 32-bit ARM/RV32 [SP20]
  - RI5CY Ascon-*p* with 4.7kGE: speedup factor 50x
  - Reuse 10 registers of CPU register file
- ARM Custom Datapath Extendion, RISC-V Bitmanip Extension, ...
  - 32-bit funnel shift instructions (RV32B: FSRI, ESP32: SRC)
  - 32-bit interleaving instructions (RV32B: ZIP/UNZIP, ARM CDE: CX3)
  - Fused AND/XOR, BIC/XOR instructions (ARM A64: BCAX, ARM CDE: CX3A)
  - SHA-2 like Sigma instructions (ARM CDE: CX3DA)

Bit-interleaved Interface

(ascon128bi32, ascon128abi32, asconhashbi32, asconhashabi32)

- Convention: data is stored/transmitted in bit interleaved format
- Communication parties need to agree, similar to endianess
- Improved performance on 32-bit ARM platforms:
  - Ascon-128/Ascon-128a: -17%/-23%
- Also demonstrates improvement of Ascon with
  - Bit-interleaving instructions (obvious)
  - Funnel shift instructions (same effect!)

Side-channel Protection

Designed with SCA in Mind

- Algebraic degree 2 of S-box
- Limited damage if state is recovered
- Leveled implementations [BBC+20]
  - Higher protection order for Init/Final (key)
  - Lower protection order for AD/PT/CT processing (data)
- Masking using Toffoli gate [DDE+20]

Masking using Toffoli Gate

- More efficient than masked AND gate
  - Fewer instructions, registers, randomness
- No fresh randomness needed during round computation
  - Randomness is not lost (invertible shared Toffoli gate)
  - Randomness of previous round can be reused
- Benefits of invertible shared function:
  - Uniform by design
  - SIFA: Reduced attack surface if used with redundancy [DDE+20]

1st-order Masked Keccak S-box

```
State:[a0,a1,b0,b1,c0,c1,d0,d1,e0,e1,r0]

(r1,r0) ← clone(r0)
toffoli_shared(r0,r1,e0,e1,a0,a1)
toffoli_shared(a0,a1,b0,b1,c0,c1)
toffoli_shared(c0,c1,d0,d1,e0,e1)
toffoli_shared(e0,e1,a0,a1,b0,b1)
toffoli_shared(b0,b1,c0,c1,d0,d1)
d0 ← xor(d0,r0)
d1 ← xor(d1,r1)
```

- Similar constructions for higher degree S-boxes may be less efficient [DDE+20]

Further SCA Optimizations

- Preliminary Goal: Achieve 1st-order protection with 2/3 shares in C[1]
  - Rotation offset between shares
  - Minimum number of ASM instructions (Toffoli gate)
  - Some register clears/NOPS needed
  - Extension to 3-shares with trick from [SM21]
- Performance in cycles/byte (green: evaluated)

impl/shares flags	armv6	C -O2	C -Os	2-1-2 -O2	2-1-2 -Os	2 -O2	2 -Os	3 -O2	3 -Os
ARM1176JZF	58	70	85	88	100	260	343	524	703
STM32F415	59	84	90	90	98	320	378	650	669

[1] Our implementations should be considered as a starting point to generate device specific C/ASM implementations

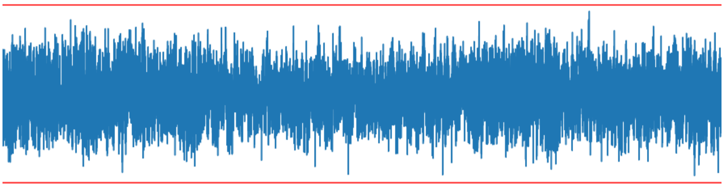
Evaluation and Verification

Testvector Leakage Assessment

- Goal: 1[st]-order protection with 2/3 shares
- Evaluation setup:
  - ChipWhisperer-Lite
  - UFO Board
  - `STM32F303`, `STM32F415`
  - We set $p^a, p^b = 2$ due to limited sample buffer
- We present decryption results of `protected_bi32_armv6`
- More implementations/results available at:
  https://github.com/ascon/simpleserial-ascon

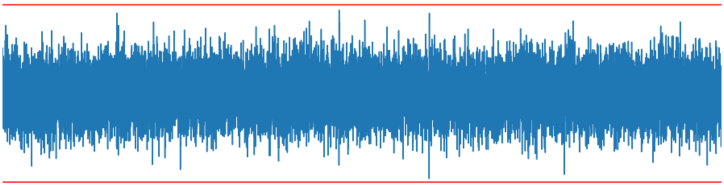
TVLA Results

- STM32F303
- 3 (rotated) shares
- No device-specific fixes
- 8m traces



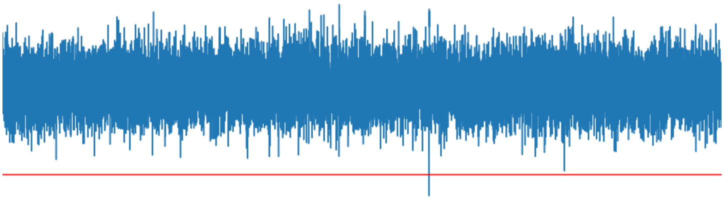
TVLA Results

- `STM32F415`
- 2 (rotated) shares
- Device-specific fixes
- 4m traces



TVLA Results

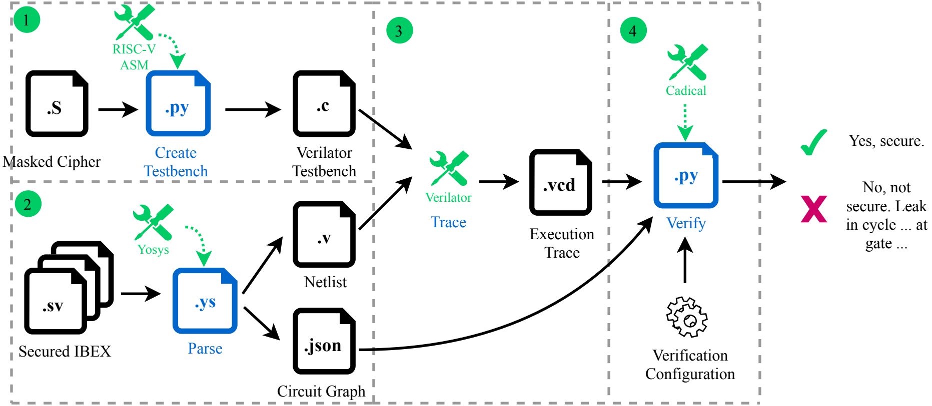
- `STM32F415`
- 2 (rotated) shares
- Device-specific fixes
- 5m traces



Formal Masking Verification

- Formal verification of masking in SW/HW using Coco [GHP+21]
  - Based on ideas of REBECCA [BGI+18]
- Verifies masked software in “hardware probing model” on CPU netlists
  - Considers stable signals, transitions, glitches
  - RISC-V IBEX core (comparable to ARM Cortex-M0)
- Also suitable for masked hardware circuits with/without state machines

Coco Verification Flow



Coco Verification Results

- Hardened RISC-V IBEX core from [GHP+21] as reference
- We mapped one round of 2-share ASCON-*p* round from to RISC-V ASM
- We verified 1st-order probing security (incl. transitions/glitches)
  - No online randomness
  - Performance of 260 c/b
  - Multi-round correctness due to uniformity of masking

Questions

Bibliography I

[Bac22] Ferdinand Bachmann. **Optimized C and Assembly Implementations for ESP32 and RISC-V**. Bachelor's Thesis (work in progress). 2022. URL: https://github.com/Ferdi265/ascon-c.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. **Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography - A Practical Guide Through the Leakage-Resistance Jungle**. Advances in Cryptology - CRYPTO 2020. Vol. 12170. Lecture Notes in Computer Science. Springer, 2020, pp. 369–400. DOI: 10.1007/978-3-030-56784-2_13. URL: https://doi.org/10.1007/978-3-030-56784-2%5C_13.

[BGI+18] Roderick Bloem, Hannes Groß, Rinat Iusupov, Bettina Könighofer, Stefan Mangard, and Johannes Winter. **Formal Verification of Masked Hardware Implementations in the Presence of Glitches**. EUROCRYPT (2). Vol. 10821. Lecture Notes in Computer Science. Springer, 2018, pp. 321–353.

Bibliography II

[CJL+20] Fabio Campos, Lars Jellema, Mauk Lemmen, Lars Müller, Daan Sprenkels, and Benoît Viguier. **Assembly or Optimized C for Lightweight Cryptography on RISC-V?** CANS 2020. Vol. 12579. LNCS. Springer, 2020, pp. 526–545. DOI: 10.1007/978-3-030-65411-5_26.

[DDE+20] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. **Protecting against Statistical Ineffective Fault Attacks**. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020.3 (2020), pp. 508–543.

[DEMS16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. **Ascon v1.2 (Submission to the CAESAR Competition)**. Final Portfolio of CAESAR: http://competitions.cr.yp.to/caesar-submissions.html. 2016.

[DEMS21a] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. **Ascon PRF, MAC, and Short-Input MAC**. IACR Cryptology ePrint Archive, Report 2021/1574. 2021. URL: https://ia.cr/2021/1574.

Bibliography III

[DEMS21b] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. **Ascon v1.2 (Submission to NIST).** NIST Finalists: https://csrc.nist.gov/Projects/Lightweight-Cryptography/Finalists. 2021.

[DEMS21c] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. **Ascon v1.2: Lightweight Authenticated Encryption and Hashing.** Journal of Cryptology 34.3 (2021), p. 33. DOI: 10.1007/s00145-021-09398-9.

[GHP+21] Barbara Gigerl, Vedad Hadzic, Robert Primas, Stefan Mangard, and Roderick Bloem. **Coco: Co-Design and Co-Verification of Masked Software Implementations on CPUs.** USENIX Security Symposium. USENIX Association, 2021, pp. 1469–1468.

[SM21] Aein Rezaei Shahmirzadi and Amir Moradi. **Second-Order SCA Security with almost no Fresh Randomness.** IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021.3 (2021), pp. 708–755.

Bibliography IV

[SP20] Stefan Steinegger and Robert Primas. **A Fast and Compact RISC-V Accelerator for Ascon and Friends.** CARDIS. Vol. 12609. Lecture Notes in Computer Science. Springer, 2020, pp. 53–67.