

# SPAE

## A Single Pass Authenticated Encryption scheme

Philippe Elbaz-Vincent<sup>1</sup>, **Cyril Hugounenq<sup>1</sup>**, **Sébastien Riou<sup>2</sup>**

<sup>1</sup>Univ. Grenoble Alpes / Institut Fourier, philippe.elbaz-vincent@univ-grenoble-alpes.fr,  
cyril.hugounenq@univ-grenoble-alpes.fr

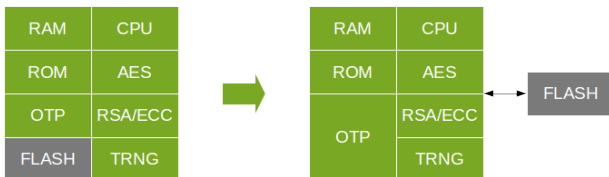
<sup>2</sup>Tiempo, France, sebastien.riou@tiempo-secure.com

This work is supported by SECURIOT-2-AAP FUI 23 and by ANR-15-IDEX-02.

WRACH, Roscoff, 18 April, 2019

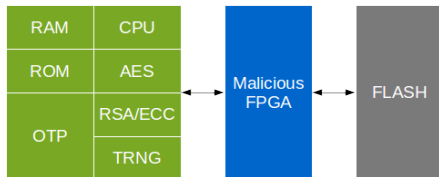
# Secure IC with external flash memory

- Typical secure element/smart card: internal flash memory (everything on single chip)
- Our goals:
  - Use external flash memory
  - Achieve same security level



# What could go wrong ?

- On the fly traffic analysis
- Replay attacks

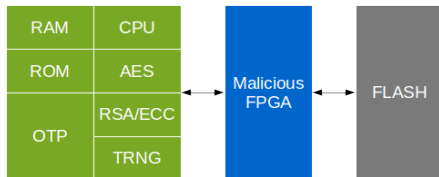


Clear need for:

- Confidentiality
- Authenticity
- Freshness

# What could go wrong ?

- On the fly traffic analysis
- Replay attacks



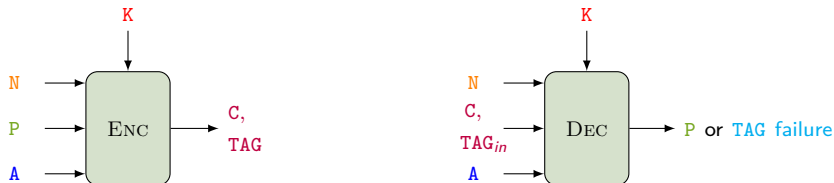
Clear need for:

- Confidentiality
- Authenticity
- Freshness

⇒ We need an Authenticated Encryption scheme.

# Authenticated Encryption (AE or AEAD)

Symmetric encrypt-sign and decrypt-verify in a single algorithm



Our use case:

- NONCE N generated and stored inside the secure element
- Cipher-text C and TAG stored outside

# Requirements of our scheme

Optimization goals:

- Silicon area,
- Performance, energy efficiency (small message size),
- Development effort.

In the context of a secure element/smart card, this means:

- Use AES (market constraint),
- Use simple linear operators (XOR, rotate...),
- Fast in single thread  $\Rightarrow$  Single Pass,
- Prevent DFA attacks at algorithm level.

## Existing AE schemes

- 2 Passes:
  - AES-GCM[MV04]
  - AES-CCM [Dwo04]
  - COLM [ABD<sup>+</sup>15] <sup>1</sup>
  - SIV [RS07]
- Not using AES:
  - NORX [AJN14]
  - ASCON [DEMS16]
  - CHACHA20-POLY1305 [Ber08], [Ber05], RFC7539
- Ideal but patented:
  - OCB[RBB03]

---

<sup>1</sup>Final portofolio members of CAESAR [Ber14] in green

## Existing AE schemes

- 2 Passes:
  - AES-GCM[MV04]
  - AES-CCM [Dwo04]
  - COLM [ABD<sup>+</sup>15]<sup>1</sup>
  - SIV [RS07]
- Not using AES:
  - NORX [AJN14]
  - ASCON [DEMS16]
  - CHACHA20-POLY1305 [Ber08], [Ber05], RFC7539
- Ideal but patented:
  - OCB[RBB03]

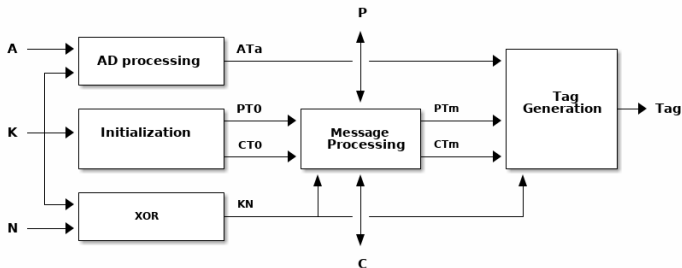
⇒ We need a new AE scheme.

---

<sup>1</sup>Final portofolio members of CAESAR [Ber14] in green



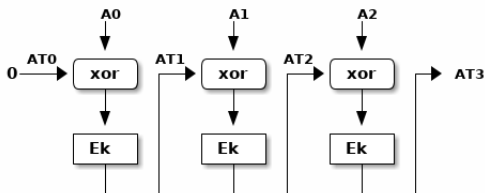
## SPAE overview



a: number of AD blocks  
m: number of message blocks  
AT<sub>a</sub>: tag over AD

KN: key derived from K and N  
PT<sub>0</sub>, CT<sub>0</sub>: initialization values  
PT<sub>m</sub>, CT<sub>m</sub>: message tag values

## SPAE Associated Data processing

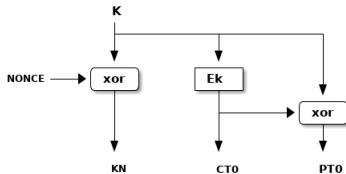


$E_k$ : block cipher call with key  $K$ , for example AES-128.

### Equations

$$AT_0 = 0 \quad AT_{i+1} = E_K(AT_i \oplus A_i) \quad A_i \text{ are blocks of associated data}$$

# SPAE Initialization and key derivation



## Equations

$$KN = \text{NONCE} \oplus K$$

$$CT_0 = E_K(K)$$

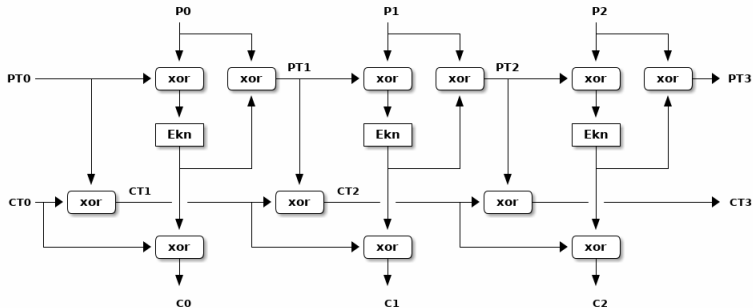
$$PT_0 = K \oplus CT_0$$

$PT_0$  and  $CT_0$  can be precomputed.

## Design Rationale

We choose those values to be strongly linked with the key since their secrecy is crucial to the security of the scheme.

# SPAE message processing



## Equations

$$C_i = E_{KN}(PT_i \oplus P_i) \oplus CT_i$$

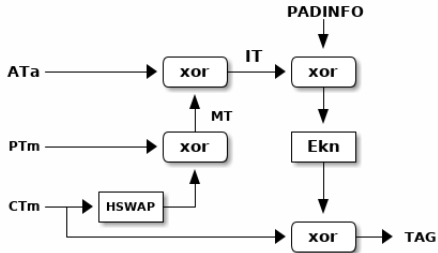
$$PT_{i+1} = E_{KN}(PT_i \oplus P_i) \oplus P_i$$

$$CT_{i+1} = PT_i \oplus CT_i$$

## Reminders

$KN = K \oplus \text{NONCE}$   
 $P_i(C_i)$  are blocks of plain(cipher)-text.  
 We aim to instantiate  $AES$  for  $E$ .

## SPAE TAG generation for $m > 0$



### Equations

$$MT = HSWAP(CT_m) \oplus PT_m \quad IT = AT_a \oplus MT$$
$$TAG = E_{KN}(IT \oplus PADINFO) \oplus CT_m$$

# Security of the scheme

## Setting of the attacker

The attacker is able to ask the encryption of any triple  $(N^i, A^i, M^i)$  but can ask only once an encryption with a same nonce  $N$ .

# Security of the scheme

## Setting of the attacker

The attacker is able to ask the encryption of any triple  $(N^i, A^i, M^i)$  but can ask only once an encryption with a same nonce  $N$ .

## Proposition

The attacker is not able to get a pair of values  $(X, E_{KN}(X))$  with some constant block  $X$ .

**Idea of the proof:** We look at all the relations between the variables and the reuse of outputs.

## Rationale Design

We choose to have two distincts internal variables to protect the knowledge of pairs of values  $(X, E_{KN}(X))$ .

# Differential analysis

## Proposition

The resilience of the scheme to differential attacks is as strong as the one of the encryption function  $E_K$  (which we aim to be *AES*).

**Idea of the proof:** To estimate the security, we upper bound the maximum probability of differential pairs  $(\delta X, \delta Y)$  we could get with the differential pair of the encryption function  $E_K$ .



# Differential Fault Analysis

The design of the scheme has been made with the aim to minimize the necessity to protect the use of  $E_K$ .

- For encryption and decryption we need only to protect the production of the TAG.

## Design Rationale

- Using a key  $KN = K \oplus \text{NONCE}$  dependant of the NONCE is a beneficial choice against DFA.
- Using *HSWAP* was motivated by DFA to avoid cancellation of non symmetrical faults in decryption.

# Privacy of the scheme

## Proposition

If the the adversary, "respecting the rules", asks  $q$  queries  $(N, A^i, M^i)$  that entails  $\sigma_n$  blockcipher calls of  $E_{KN}$  then

$$\text{Adv}_{\Pi}^{\text{priv}} \leq \frac{1.5\sigma_n(\sigma_n - 1)}{2^{\text{blocksize}}}.$$

For example with AES  $\text{blocksize} = 128$ .

**Idea of the proof:** We use a game playing argument measuring the distance to a perfect blockcipher (see lemma 3 of Krovetz and Rogaway [KR11] for details).

# Authenticity of the scheme

## Proposition

If the adversary asks  $q$  queries that entails  $\sigma$  blockcipher calls then

$$\text{Adv}_{\Pi}^{\text{auth}} \leq \frac{1}{\Gamma}$$

with  $\Gamma$  the size of the codomain of the function  $(x) \mapsto x \oplus E_K(x)$ .

**Idea of the proof** We make a strong supposition for the attacker and we conclude by the fact that the attacker does not know any couple of values  $X, E_K(X)$ .

## Benchmark: ARM-Cortex-M4

### ■ AES implementations:

- MMCAU: Flexible cryptographic accelerator,
- FAST: Software AES optimized for speed (use 8 Kbytes Tbox LUT),
- SMALL: Software AES optimized for size (use 256 bytes Sbox LUT).

**Table:** MbedTLS benchmark<sup>2</sup> on FRDM-K64F board, 1024 bytes messages

Algorithm	AES implementation	Kbytes/s	cycles/byte
AES-SPAE-128	MMCAU	3101	37.8
AES-SPAE-128	FAST	1141	102.9
AES-SPAE-128	SMALL	546	215.1
AES-GCM-128	FAST	401	293.0
AES-CCM-128	FAST	476	246.8

<sup>2</sup>Benchmarking code taken from <https://github.com/wolfeidau/mbedtls>

## Benchmark: ARM-Cortex-M0

STM32L011K4 is a low end device:

- no hardware AES,
- only 16KB FLASH, 2KB RAM.

**Table:** Benchmark on STM32L011 Nucleo board

	clock cycles	cycles/byte
SPAE	18.2K	1140
CCM	42.0K	2627
OCB	43.0K	2689
GCM	65.6K	4100

Scenario: encrypt and authenticate a 16 bytes message  
CCM,OCB and GCM implementations from CIFRA library<sup>3</sup>

<sup>3</sup><https://github.com/ctz/cifra>

# Conclusion

SPAE is a new AE algorithm:

- Single pass,
- Use only a block cipher and XOR,
- With AES, it is faster than AES-GCM and AES-CCM<sup>4</sup>,
- Not patented,
- Some security bounds,
- Some algorithmic level fault attack protections,
- Python and C code available at  
<https://github.com/TiempoSecure/SPAE>.

Further work:

- Adaptation to AES-256 (only about KN).
- Practical evaluation of fault attacks<sup>5</sup>.

---

<sup>4</sup>On typical low end MCUs where parallelization is not possible

<sup>5</sup>Feel free to ask us for a STM32 nucleo board to challenge our claims

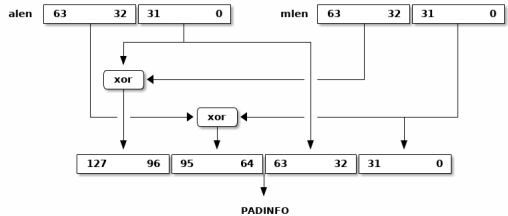
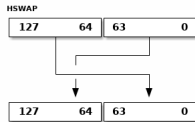
- [ABD<sup>+</sup>15] Elena Andreeva, Andrey Bogdanov, Nilanjan Datta, Atul Luykx, Bart Mennink, Mridul Nandi, Elmar Tischhauser, and Kan Yasuda.  
Submission to CAESAR competition: COLM v1, 2015.
- [AJN14] Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves.  
Norx: Parallel and scalable aead, 2014.  
<https://norx.io/>.
- [Ber05] Daniel J. Bernstein.  
The Poly1305-AES Message-Authentication Code.  
In *Fast Software Encryption*, pages 32–49. Springer Berlin Heidelberg, 2005.
- [Ber08] Daniel J. Bernstein.  
ChaCha, a variant of Salsa20, 2008.

- [Ber14] Daniel J. Bernstein.  
Caesar: Competition for authenticated encryption: Security, applicability, and robustness, 2014.
- [DEMS16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schl affer.  
Ascon v1.2.  
Submission to the CAESAR competition:  
<http://competitions.cr.yj.p.to/round3/asconv12.pdf>, 2016.
- [Dwo04] Morris Dworkin.  
Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality.  
Technical report, National Institute of Standards and Technology, 2004.

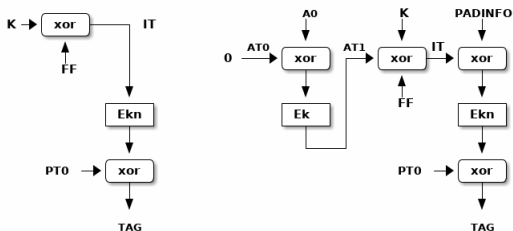


- [KR11] Ted Krovetz and Phillip Rogaway.  
The software performance of authenticated-encryption modes.  
*In International Workshop on Fast Software Encryption*, pages 306–327. Springer, 2011.
- [MV04] David McGrew and John Viega.  
The Galois/counter mode of operation (GCM).  
*Submission to NIST Modes of Operation Process*, 20, 2004.
- [RBB03] Phillip Rogaway, Mihir Bellare, and John Black.  
OCB: A Block-cipher Mode of Operation for Efficient Authenticated Encryption.  
*ACM Trans. Inf. Syst. Secur.*, 6(3):365–403, aug 2003.
- [RS07] Phillip Rogaway and Thomas Shrimpton.  
The SIV Mode of Operation for Deterministic Authenticated-Encryption (Key Wrap) and Misuse-Resistant Nonce-Based Authenticated-Encryption, 2007.

# Computation of PADINFO



## SPAE tag generation for $m=0$



$$TAG_{null} = PT_0 \oplus E_{KN}(K \oplus FF) = K \oplus E_K(K) \oplus E_{KN}(K \oplus FF)$$

FF in the formulae prevents  $TAG = K$  for  $NONCE = 0$

The present work has been partly funded by BPI France and FEDER (*Fonds Européen de Développement Économique Régional*) Rhône-Alpes Auvergne under national project "SecurIoT-2" supported by the four competitiveness clusters Minalogic, SCS, Systematic Paris-Région, and Derbi.

