Motivations
Design of SPAE
Security of the scheme
Performances

SPAE
A Single Pass Authenticated Encryption scheme

Philippe Elbaz-Vincent$^1$, Cyril Hugounenq$^1$, Sébastien Riou$^2$

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

$^2$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

<table>
<thead>
<tr>
<th>RAM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>AES</td>
</tr>
<tr>
<td>OTP</td>
<td>RSA/ECC</td>
</tr>
<tr>
<td>FLASH</td>
<td>TRNG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RAM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>AES</td>
</tr>
<tr>
<td>OTP</td>
<td>RSA/ECC</td>
</tr>
<tr>
<td>FLASH</td>
<td>TRNG</td>
</tr>
</tbody>
</table>

FLASH
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

<table>
<thead>
<tr>
<th>RAM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>AES</td>
</tr>
</tbody>
</table>
| OTP  | RSA/ECC
     | TRNG |

Malicious FPGA

FLAS

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 ⇒ Single Pass,
- Prevent DFA attacks at algorithm level.
Existing AE schemes

- 2 Passes:
  - AES-GCM [MV04]
  - AES-CCM [Dwo04]
  - COLM [ABD+15] \(^1\)
  - SIV [RS07]

- Not using AES:
  - NORX [AJN14]
  - ASCON [DEMS16]
  - CHACHA20-POLY1305 [Ber08], [Ber05], RFC7539

- Ideal but patented:
  - OCB [RBB03]

\(^1\)Final portfolio members of CAESAR [Ber14] in green
Existing AE schemes

- **2 Passes:**
  - AES-GCM[MV04]
  - AES-CCM [Dwo04]
  - COLM [ABD+15] \(^1\)
  - 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.

\(^1\)Final portofolio members of CAESAR [Ber14] in green
SPAЕ overview

- **a**: number of AD blocks
- **m**: number of message blocks
- **AT_a**: tag over AD
- **KN**: key derived from K and N
- **PT_0, CT_0**: initialization values
- **PT_m, CT_m**: message tag values
**Ek:** block cipher call with key $K$, for example AES-128.

**Equations**

\[
\begin{align*}
A_{T0} &= 0 \\
A_{T_{i+1}} &= E_K(A_{T_i} \oplus A_i) \\
\end{align*}
\]

$A_i$ are blocks of associated data.
Motivations

Design of SPAE

Security of the scheme

Performances

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.
Motivations
Design of SPAE
Security of the scheme
Performances

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

\[ KN = K \oplus NONCE \]

\( P_i(C_i) \) are blocks of plain(cipher)-text.

We aim to instantiate AES for \( E \).
Motivations
Design of SPAE
Security of the scheme
Performances

SPAE TAG generation for \( m > 0 \)

Equations

\[ MT = HSWAP(CT_m) \oplus PT_m \]
\[ IT = AT_a \oplus MT \]
\[ TAG = E_{KN}(IT \oplus PADINFO) \oplus CT_m \]
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 \( E_K \).
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 $KN = K \oplus NONCE$ 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 adversary, "respecting the rules", asks $q$ queries $(N, A^i, M^i)$ that entails $\sigma_n$ blockcipher calls of $E_{K_N}$ then

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

For example with AES 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

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

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\(^2\) on FRDM-K64F board, 1024 bytes messages

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AES implementation</th>
<th>Kbytes/s</th>
<th>cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-SPAE-128</td>
<td>MMCAU</td>
<td>3101</td>
<td>37.8</td>
</tr>
<tr>
<td>AES-SPAE-128</td>
<td>FAST</td>
<td>1141</td>
<td>102.9</td>
</tr>
<tr>
<td>AES-SPAE-128</td>
<td>SMALL</td>
<td>546</td>
<td>215.1</td>
</tr>
<tr>
<td>AES-GCM-128</td>
<td>FAST</td>
<td>401</td>
<td>293.0</td>
</tr>
<tr>
<td>AES-CCM-128</td>
<td>FAST</td>
<td>476</td>
<td>246.8</td>
</tr>
</tbody>
</table>

\(^2\)Benchmarking code taken from [https://github.com/wolfeidau/mbedtls](https://github.com/wolfeidau/mbedtls)
BMbenchmark: ARM-Cortex-M0

STM32L011K4 is a low end device:
- no hardware AES,
- only 16KB FLASH, 2KB RAM.

Table: Benchmark on STM32L011 Nucleo board

<table>
<thead>
<tr>
<th></th>
<th>clock cycles</th>
<th>cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPAE</td>
<td>18.2K</td>
<td>1140</td>
</tr>
<tr>
<td>CCM</td>
<td>42.0K</td>
<td>2627</td>
</tr>
<tr>
<td>OCB</td>
<td>43.0K</td>
<td>2689</td>
</tr>
<tr>
<td>GCM</td>
<td>65.6K</td>
<td>4100</td>
</tr>
</tbody>
</table>

Scenario: encrypt and authenticate a 16 bytes message
CCM, OCB and GCM implementations from CIFRA library³

³https://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\(^4\),
- 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\(^5\).

\(^4\)On typical low end MCUs where parallelization is not possible
\(^5\)Feel free to ask us for a STM32 nucleo board to challenge our claims
Motivations
Design of SPAE
Security of the scheme
Performances

Submission to CAESAR competition: COLM v1, 2015.


[Ber05] Daniel J. Bernstein.

[Ber08] Daniel J. Bernstein.
ChaCha, a variant of Salsa20, 2008.

[DEMS16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer.
Ascon v1.2.
Submission to the CAESAR competition:

[Dwo04] Morris Dworkin.
Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality.

The software performance of authenticated-encryption modes.

[MV04] David McGrew and John Viega.
The Galois/counter mode of operation (GCM).

[RBB03] Phillip Rogaway, Mihir Bellare, and John Black.
OCB: A Block-cipher Mode of Operation for Efficient
Authenticated Encryption.

[RS07] Phillip Rogaway and Thomas Shrimpton.
The SIV Mode of Operation for Deterministic
Authenticated-Encryption (Key Wrap) and Misuse-Resistant
Computation of PADINFO
Motivations  
Design of SPAE  
Security of the scheme  
Performances

**SPAE tag generation for m=0**

\[ \text{TAG}_{\text{null}} = PT_0 \oplus E_{KN}(K \oplus FF) = K \oplus E_K(K) \oplus E_{KN}(K \oplus FF) \]

FF in the formulae prevents \( \text{TAG} = K \) for NONCE = 0