Side-channel attacks on Ascon's S-box

Alexane Boldo

ENS Rennes

OCIF, IRISA

IMT Atlantique

May-July 2025

Supervisor: Hélène Le Bouder

Introduction

Side-Channel Attacks (SCA): observation of computation time, power consumption, electromagnetic radiation, ... to discover a secret

Goal: Study the leaks from the winner for lightweight cryptography Ascon to theorize a SCA attack

Table of Contents

What is Ascon-AEAD?

Authenticated Encryption with Associated Data (AEAD): encrypt, check authentication of content and associated data

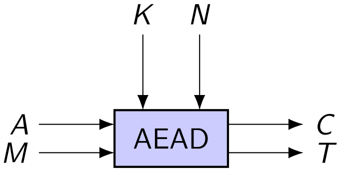

Figure: AEAD algorithm from [1]

Ascon's State

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	
IV								S_0
first half of K, K_0								S_1
second half of K, K_1								S_2
first half of N, N_0								S_3
second half of N, N_1								S_4

Encryption and decryption phases

4 phases: initialization, associated data process, plaintext/ciphertext process, finalization

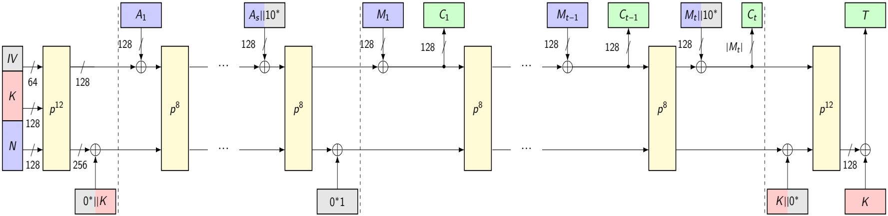

Figure: Ascon-AEAD mode, from [1]

Ascon's permutation

$p = p_L \circ \underbrace{p_S} \circ p_C$

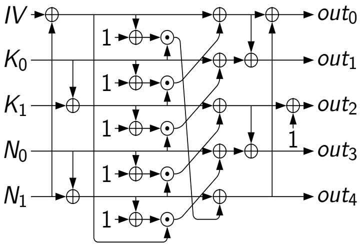

Figure: Circuit to compute the S-box, from [2], permutation of ⟦1;31⟧

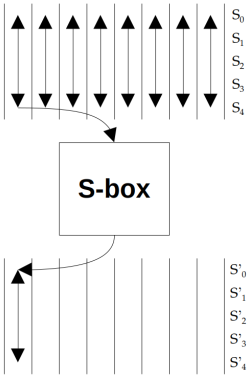

Figure: S-box computation for the first byte of each word

Table linking the output of the S-box and the key

(N_0^j, N_1^j, IV^j)	S_4^j
$(0,0,0)$	K_0^j
$(0,0,1)$	0
$(0,1,0)$	1
$(0,1,1)$	$1 \oplus K_0^j$
$(1,0,0)$	$1 \oplus K_0^j$
$(1,0,1)$	1
$(1,1,0)$	0
$(1,1,1)$	K_0^j

Figure: Link between K_0^j and S_4^j depending on IV and N, from [3]

ChipWhisperer-Lite

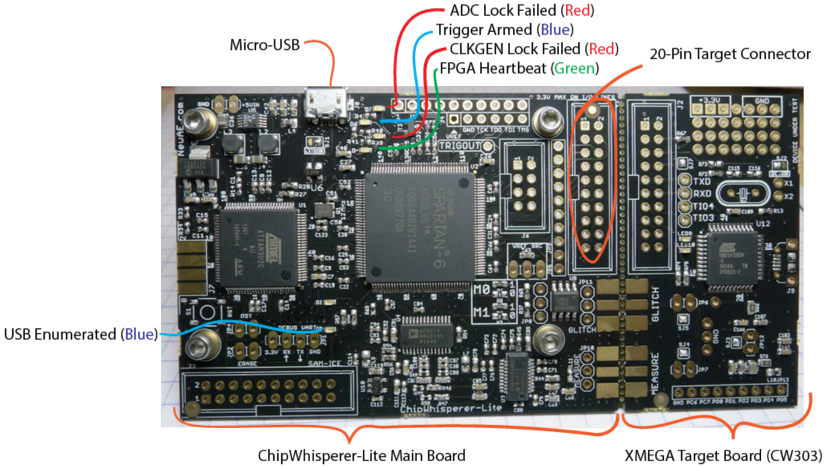

Figure: ChipWhisperer Lite board, from [4]

Analyses done

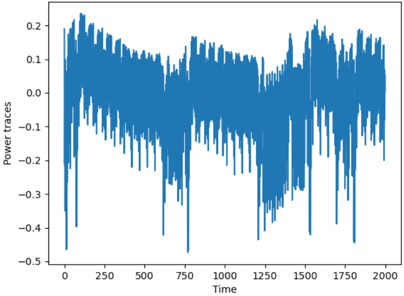

Figure: Power trace during Ascon's S-box

- Finding the best model
  - Vertical vs horizontal
  - HW vs value
- Attack: finding the vertical output and deduce the key

Results vertical vs horizontal and HW vs value

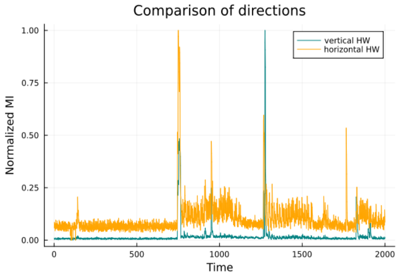

Figure: Mutual information for the horizontal and the vertical value

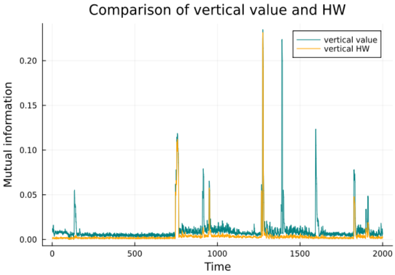

Figure: Mutual information between power consumption and HW or value

Results attack

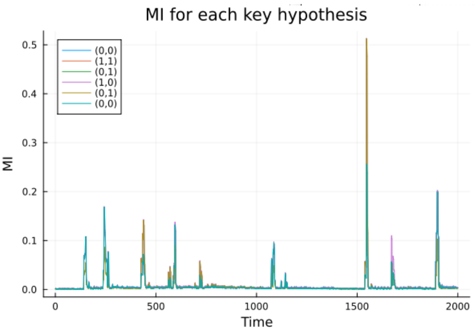

Figure: Mutual information between the HW of the outputs and the power consumption, for each of the possible outputs for the first nonce

Conclusion

- Good leaks compared to random values
- Though apparent weaknesses, unsuccessful attempts
- Not enough randomness with false key hypotheses
- Leads to follow: belief propagation

- L. B. H., T. G., and A. D., "Théorie de la cryptographie."
- Tikz. [Online]. Available: https://extgit.isec.tugraz.at/meichlseder/tikz
- S. M., "Side channel analysis against aead." [Online]. Available: https://theses.hal.science/tel-04816066v1
- Chipwhisperer documentation. [Online]. Available: https://chipwhisperer.readthedocs.io/en/latest/getting-started.html

Permutation (1), p_C

Constant for the round i: $const_{16-nb_{rounds}+i}$

i	const$_i$	i	const$_i$
0	0x000000000000003c	8	0x00000000000000b4
1	0x000000000000002d	9	0x00000000000000a5
2	0x000000000000001e	10	0x0000000000000096
3	0x000000000000000f	11	0x0000000000000087
4	0x00000000000000f0	12	0x0000000000000078
5	0x00000000000000e1	13	0x0000000000000069
6	0x00000000000000d2	14	0x000000000000005a
7	0x00000000000000c1	15	0x000000000000004b

Figure: Constant-addition layer, constants

Permutation (2), p_C

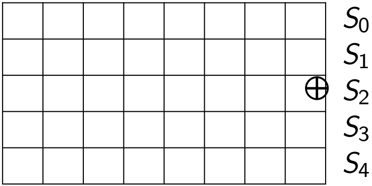

Figure: Constant-addition layer, each box representing a byte of one of the 64-bit words

Permutation (3), p_S

x	0	1	2	3	4	5	6	7
$S-box(x)$	4	b	1f	14	1a	15	9	2
x	8	9	a	b	c	d	e	f
$S-box(x)$	1b	5	8	12	1d	3	6	1c
x	10	11	12	13	14	15	16	17
$S-box(x)$	1e	13	7	e	0	d	11	18
x	18	19	1a	1b	1c	1d	1e	1f
$S-box(x)$	10	c	1	19	16	a	f	17

Figure: Lookup table for the 5-bit S-box

Permutation (4), p_S

```
state[0] ^= state[4];
state[4] ^= state[3];
state[2] ^= state[1];
uint64_t t0 = ~state[0];
uint64_t t1 = ~state[1];
uint64_t t2 = ~state[2];
uint64_t t3 = ~state[3];
uint64_t t4 = ~state[4];
t0 &= state[1];
t1 &= state[2];
t2 &= state[3];
t3 &= state[4];
t4 &= state[0];
state[0] ^= t1
; state[1] ^= t2;
state[2] ^= t3;
state[3] ^= t4;
state[4] ^= t0;
state[1] ^= state[0];
state[0] ^= state[4];
state[3] ^= state[2];
state[2] =~ state[2];
```

Figure: Equations to compute the S-box

Permutation (5), p_L

Diffusion: $S_i \leftarrow \Sigma_i(S_i)$:

$$\begin{aligned}
\Sigma_0(S_0) &= S_0 \oplus (S_0 >>> 19) \oplus (S_0 >>> 28) \\
\Sigma_1(S_1) &= S_1 \oplus (S_1 >>> 61) \oplus (S_1 >>> 39) \\
\Sigma_2(S_2) &= S_2 \oplus (S_2 >>> 1) \oplus (S_2 >>> 6) \\
\Sigma_3(S_3) &= S_3 \oplus (S_3 >>> 10) \oplus (S_3 >>> 17) \\
\Sigma_4(S_4) &= S_4 \oplus (S_4 >>> 7) \oplus (S_4 >>> 41)
\end{aligned}$$

Finding this table (1)

$$S_4^j = n_o^j \oplus n_1^j \oplus k_0^j \times (1 \oplus IV^j \oplus n_1^j)$$

$$S_4^j = \begin{cases} k_0^j \times (1 \oplus IV^j) & if\ (n_0^j, n_1^j) = (0,0) \\ k_0^j \times IV^j & if\ (n_0^j, n_1^j) = (1,1) \\ 1 \oplus k_0^j \times IV^j & if\ (n_0^j, n_1^j) = (0,1) \\ 1 \oplus k_0^j \times (1 \oplus IV^j) & if\ (n_0^j, n_1^j) = (1,0) \end{cases}$$

Finding this table (2)

Then if $IV^j = 0$:

$$S_4^j = \begin{cases} k_0^j & if \ (n_0^j, n_1^j) = (0,0) \\ 0 & if \ (n_0^j, n_1^j) = (1,1) \\ 1 & if \ (n_0^j, n_1^j) = (0,1) \\ 1 \oplus k_0^j & if \ (n_0^j, n_1^j) = (1,0) \end{cases}$$

Finding this table (3)

Otherwise, if $IV^j = 1$:

$$S_4^j = \begin{cases} 0 & \textit{if } (n_0^j, n_1^j) = (0,0) \\ k_0^j & \textit{if } (n_0^j, n_1^j) = (1,1) \\ 1 \oplus k_0^j & \textit{if } (n_0^j, n_1^j) = (0,1) \\ 1 & \textit{if } (n_0^j, n_1^j) = (1,0) \end{cases}$$

Complementary graph (1)

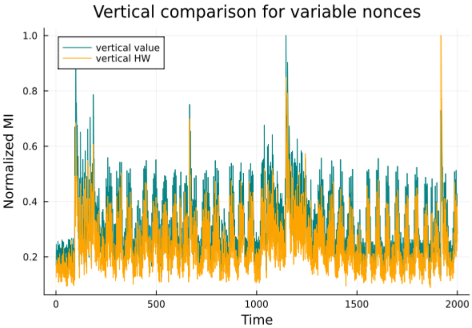

Figure: Mutual information between power consumption and Hamming weight of the concatenation of the first bit of each of the word of S and its value like 9 but for random nonces

Complementary graph (2)

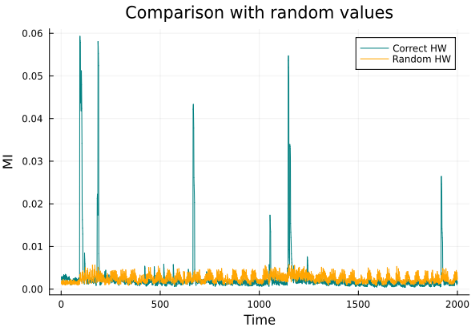

Figure: Mutual information between power consumption and vertical HW or random possible HW