Side-Channel Evaluation on Protected Implementations of Several NIST LWC Finalists

Dawu Gu, Pei Cao, Yuhang Ji, Xiangjun Lu, Shipei Qu, Tengfei Wang,
Chi Zhang, Hongyi Zhang, Xiaolin Zhang (sorted alphabetically by last name)
Cryptology and Computer Security Laboratory (LoCCS)

School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China

August 12, 2022

On the Side-channel Leakage Assessment of Ascon with Boolean Masking

Hongyi Zhang[1], Pei Cao[1]

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

1 Introduction

In the report, we will make a side-channel leakage assessment for Ascon with second-order boolean masking in software implementation(STM32F303) and first-order boolean masking in hardware implementation(Sakura-X). With collected power traces, the report will show the capability of counter side-channel analysis of Ascon through three tests, including Welch's t-test, χ^2-test, and correlation power analysis (CPA).

2 Our Work and Results Overview

In this report, our work and the assessment results of the side-channel leakage assessment on Ascon can be concluded as follows:

- We collected two trace sets from the given software and hardware implementations of Ascon on an STM32F303 MCU and a SAKURA-X evaluation board
- We performed Welch's t-test and χ^2-test to evaluate the power leakage condition of Ascon. Also, we tried to recover the private keys of Ascon by CPA
- Welch's t-test and χ^2-test did **not** show obvious leakage of intermediate value
- CPA **cannot** recover the private key bits in the software and hardware implementations when protection is applied to Ascon

3 Assessment Strategy

There are three phases in the assessment strategy on the given Ascon implementations:
Phase 1: Determine the intermediate value for analysis.
As far as we know, unprotected Ascon has been shown to be insecure under CPA [SD17]. Since we know the contents of the state at initialization, except for the key part. And we can vary the nonce each run, we pick the end of the first round of the initialization phase as our point of analysis (see figure 1). Specifically, we chose the output of the first round of permutation Keccak-p as the intermediate value:

$$y0 = k(m' + 1) + m \tag{1}$$

where $y0$ is the output of the non-linear S-box, k is one bit of the key, m' and m are related to the variable nonce.
Phase 2: Discern possible power leakage.

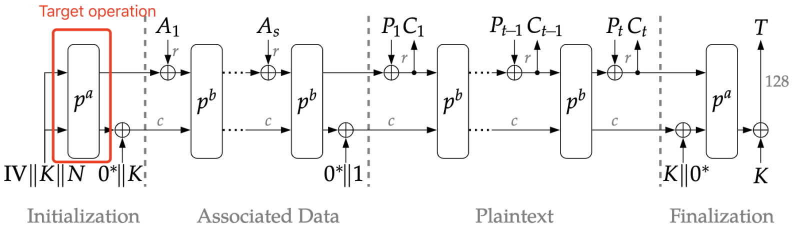


Figure 1: Encryption of Ascon

Next, we need to determine if there is power leakage in the process of Ascon encryption. As mentioned before, we assess the power traces with Welch's t-test and χ^2 test, which can help us discover the possible leakage points in the power trace. These leakage points in turn help us to find the corresponding operation that cause the leakage.

Phase 3: Recover the secret key bits.

If we detect the possible power leakage in Phase 2, then we can apply CPA to the Ascon to recover the key bits. In the process of CPA, we would foucs on the first three bits of the private key, which should be easily obtained if there is leakage in Ascon's encrpytion.

4 Assessment on the EM traces of software implementation

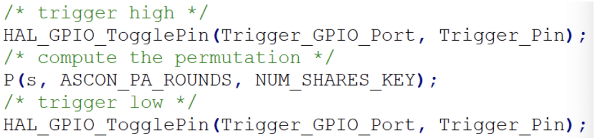
4.1 Experiment Procedure

First, we download the Ascon's firmware of the C implementation into the STM32F303's flash memory. Then we connect the STM32F303 chip to the host computer through a USB serial port to execute the algorithm and record the input and output data.At the same time, we collect the electromagnetic power traces of the chip with a high-precision electromagnetic probe. And the sampled traces are recorded by the Pico-3203D oscilloscope in the form of .trs files. After repeatedly collected a large number of traces, we can have a power trace set of the masked Ascon for the assessment. In the experiment, the software code applied to STM32F303 can be found on github(https://github.com/ascon/simpleserial-ascon/releases/tag/v1.2.6).

Table 1: Details of experimental environments for software implementation.

Items	Details
Target MCU	STM32F303RCT6
EM probe	Langer RF-U 5-2
Oscilloscope	Pico 3203D
Baud rate (USB Serial Port)	115200 bps
Sampling rate	62.5MS/s
Amplifier	Mini-Circuits ZKL-1R5+
Ascon Code Version	protected_bi32_armv6

Trigger location. When sampling traces, we use a trigger signal to locate the timing when the target operation (the permutation Keccak-p in the initialization phase) is executed. Therefore, we need to modify the original Ascon implementations so that they can control the corresponding pins of the device to send trigger signals at sepcific time point. To achieve this function, we insert the controlling soruce codes into the 'ascon_initaead' function, specifically, before and after the call of the 'P' function (see figure 2). To cover larger parts of the implementation, the number of rounds have been reduced to 2 rounds for PA and PB.



```
/* trigger high */
HAL_GPIO_TogglePin(Trigger_GPIO_Port, Trigger_Pin);
/* compute the permutation */
P(s, ASCON_PA_ROUNDS, NUM_SHARES_KEY);
/* trigger low */
HAL_GPIO_TogglePin(Trigger_GPIO_Port, Trigger_Pin);
```

Figure 2: Code snippet to set trigger in the software implementation

Input data of Ascon. During the experiments of EM trace collection, the input of Ascon encryption consists of three parts: a 16-byte nonce, 16-byte associated data and 16-byte plaintext. The output is 32-byte, including a 16-byte ciphertext and a 16-byte authenticated tag. The 16-byte encryption key is fixed throughout the collection. The detailed information about the fixed input is shown in table 2. Since changing solely the input nonce will change the intermediate values, we choose to alter the nonce in each encryption. Then the intermediate values will change under the same key, thereby generating different but related power consumption patterns. This allows us to perform CPA and other tests.

Table 2: Details of input for software Ascon implementation.

Items	Details
Key	000102030405060708090A0B0C0D0E0F
Plaintext	000102030405060708090A0B0C0D0E0F
Associated data	000102030405060708090A0B0C0D0E0F
Nonce	random

Trace information. The basic information of collected traces is presented in Table 3.

Table 3: Basic information of the collected EM traces for software Ascon implementation.

Items	Details
No. of traces	60000
No. of points per trace	80000
Precision	$-2^{15} \sim 2^{15}$
Sampling time	4 hours

4.2 Result of Welch's t-test

We use Welch's t-test, which is used to compare the means of two sample groups, to examine if there is any leakage points in the power trace. To do the t-test, we compart the power traces into two groups according to the difference in their intermediate values. As the private key is fixed, we can divide the power traces of Ascon by the intermediate value $y0$ in equation 1, which is determined by the first bit of the secret key. The test results are shown in figure 3.

4.3 Result of χ^2-test

χ^2-test is a statistical hypothesis test to determine whether there is a significant difference between the expected and observed frequencies. It can also test the null hypothesis of

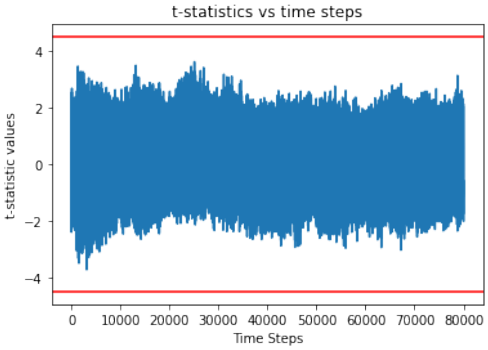


Figure 3: Results of Welch's t-test on Ascon software implementation

independence of a pair of random variables. Therefore, like t-test, we need to divide the power traces of Ascon by the intermediate value $y0$ in equation 1 and observe their statistical differences.

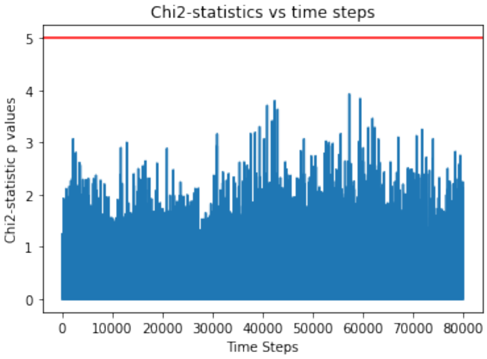


Figure 4: Result of χ^2-test on Ascon software implementation

4.4 CPA results for software power traces

We try to recover the first three bits of the key by using CPA(Correlation Power Analysis), which should be 000. In this way, we get the result as below, the figure shows the leakage situation at different sample points.

And best 8 keys guessed by CPA are shown in Table 4:

We can see the best key is 4 and 5, which are not the correct ones. Therefore, the CPA against Ascon power traces of software implementation is not successful, showing that the Ascon can resist the side-channel attacks.

Table 4: Best 8 keys guessed by CPA.

Key Rank	Key	Correlation Value	Leak Position
0	4(100)	0.014632203153923171	17218
1	5(101)	0.014632203153923171	17218
2	0(000)	0.014263695444477729	39050
3	1(001)	0.014263695444477729	39050
4	6(110)	0.013485215252643857	47599
5	7(111)	0.013485215252643857	47599
6	2(010)	0.012931953494060296	40501
7	3(011)	0.012931953494060296	40501

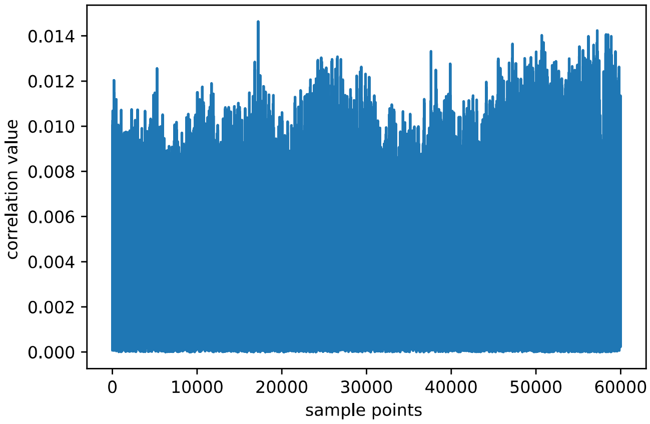


Figure 5: CPA result for Ascon using software power traces

5 Assessment on the power traces of hardware implementation

5.1 Experimental Setting

We first need to download the firmware of Ascon into the SAKURA-X. Then we connect the device to the host computer through a USB serial port so that we can execute the cipher and record its input and output. The captured power comsumption is then transmitted to the oscilloscope to generate and display the waveform of electronic signals. With the help of the oscilloscope, we can acquire enough raw power traces of protected Ascon in the host computer for later assessment. The source code of hardware implementation can be found online(https://cryptography.gmu.edu/athena/LWC/LWC_Finalists_protected_HW_implementations.html).

Input data of Ascon. During the experiments of power trace collection, the input of Ascon encryption consists of four parts (see table 6). Only nonce is variable, the other inputs (i.e., key, plaintext, and associated data) are fixed.

Trace information. The basic information of collected traces is presented in Table 7.

Table 5: Details of experimental environments for hardware implementation.

Items	Details
Target platform	SAKURA-X (with Xilinx Kintex-7 FPGA)
Oscilloscope	LeCroy 610Zi
Sampling rate	1GS/s
Ascon code version	ASCON_HPC2(first order boolean mask)

Table 6: Details of input for hardware Ascon implementation.

Items	Details
Key	D90E654D39818255180DD3DCA9FAEB4B
Plaintext	/
Associated data	81B2D700
Nonce	random

Table 7: Basic information of the collected traces for hardware Ascon implementation.

Items	Details
No. of traces	1000000
No. of points per trace	10000
Precision	$-2^7 \sim 2^7$
Sampling time	8 hours

5.2 Result of Welch's t-test

The result is shown in figure 6

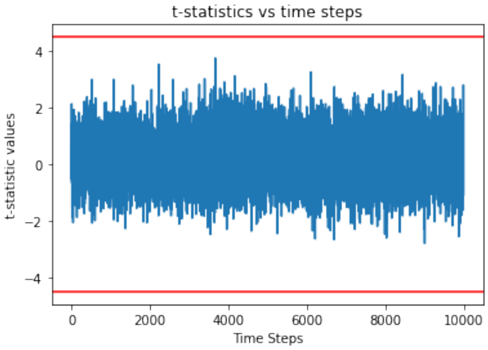


Figure 6: Results of Welch's t-test on Ascon hardware implementation

5.3 Result of χ^2-test

The result of χ^2-test is as below:

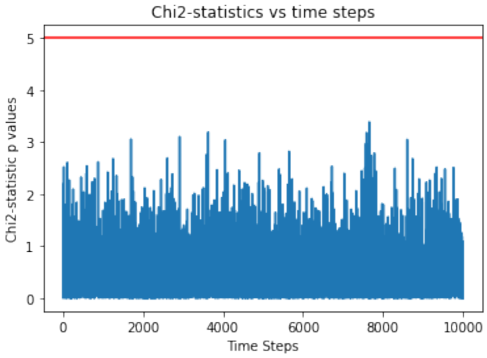


Figure 7: Results of χ^2-test on Ascon hardware implementation

5.4 CPA results for hardware power traces

We try to recover the first three bit of the key by using CPA(Correlation Power Analysis), which should be 000. And we get the result as below, the figure 8 shows the leakage situation at different sample points. We can see the best guessing is 110, which is far from being the right answer 000.

Best 8 guessed keys are shown in Table 8 below:

Table 8: Best 8 keys guessed by CPA.

Key Rank	Key	Correlation Value	Leak Position
Rank0	key 6(110)	0.049118043345022504	9823
Rank1	key 7(111)	0.049118043345022504	9823
Rank2	key 2(010)	0.04863842222881651	7496
Rank3	key 3(011)	0.04863842222881651	7496
Rank4	key 0(000)	0.045888110508008735	7497
Rank5	key 1(001)	0.045888110508008735	7497
Rank6	key 4(100)	0.04552392802497998	7496
Rank7	key 5(101)	0.04552392802497998	7496

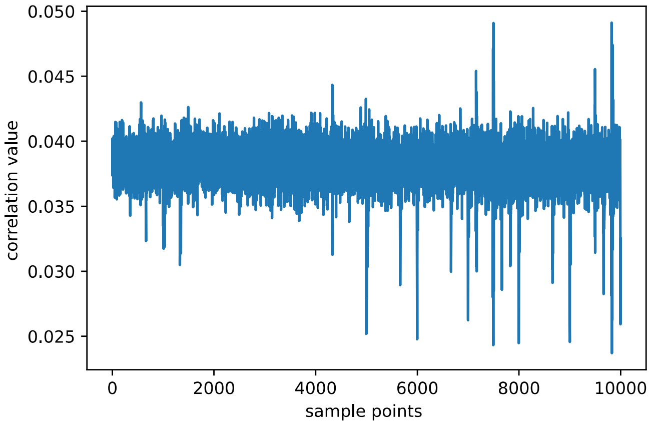


Figure 8: CPA result for Ascon using hardware power traces

References

[SD17] Niels Samwel and Joan Daemen. DPA on hardware implementations of ascon and keyak. In *Proceedings of the Computing Frontiers Conference, CF'17, Siena, Italy, May 15-17, 2017*, pages 415–424. ACM, 2017.