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

Side-channel Evaluation of ISAP

Yuhang Ji[1]

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

1 Introduction

1.1 Background

ISAP is a family of nonce-based authenticated ciphers with associated data (AEAD) designed with a focus on robustness against passive side-channel attacks. All ISAP family members are permutation-based designs that combine variants of the sponge-based ISAP mode with one of several published lightweight permutations.

Power side-channel analysis enables attackers to collect the power consumption of a cryptographic hardware device, which allows them to infer the secrets inside, e.g. private keys. More precisely, simple power analysis (SPA) refers to interpreting raw power traces visually to deduce the patterns of cryptographic operations. Dierential power analysis (DPA) is a more advanced technique based on statistical analysis, which helps attackers to reveal the original key through intermediate values of the cryptographic computations. Over the decade, deep learning (DL) has been developed as a powerful tool for side-channel attacks.

ISAP is known for its resilience against DPA attacks. Until the day of this report, there is no published work regarding ISAP's side-channel security. In this report, we perform a side-channel leakage assessment against ISAP. The collected power traces are going through several tests such as Welchs t-test and correlation power analysis (CPA) to demonstrate the actual performance of the side-channel resilience of ISAP.

1.2 Our Work and Results Overview

Our work in this report and the results of the side-channel leakage assessment on ISAP can be summarized as follows.

- We collected three trace sets from the given software and hardware implementations of ISAP-Ascon on an MCU and a side-channel attack evaluation board.

- We performed side-channel leakage assessment on ISAPRk procedure of an ISAP encryption. Welchs t-test and χ^2-test were used to evaluate the power leakage of ISAP-Ascon.

- CPA attack cannot recover the private key bytes under the given implementations.

The overall assessment reveals that power leakage mainly comes from associated data input of the ISAPRk procedure, while the actual private key and permutation does not induce any leakage.

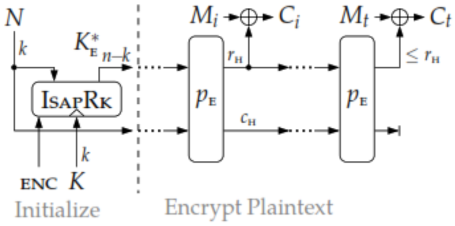

Figure 1: Authenticated Encryption of ISAP

2 Assessment Strategy

2.1 Specify the Targets

Side-channel analysis towards a cryptography algorithm normally requires three essential elements: a random variable x that's visible to attacker, a fixed secret value k the attacker wants to recover, and a non-linear operation f which takes x and k as inputs and generates sensitive values c. c is known as the sensitive intermediate values to k. Operation f on the crypto hardware generates side-channel leakages, which is know as traces. Normally the power leakage increases by the non-linearity of operation f. Side-channel analysis heavily relies on the selection of these three elements.

The ISAP algorithm provides several modes, such as authenticated encryption/decryption and MAC. All of the modes are designed to sponge-based constructions. The basic operation unit of ISAP is permutation. The states of the opted permutation are used to absorb data and squeeze data. Permutation used in ISAP can be either Ascon or Keccak. The selection of the permutation leverages on the environment the ISAP needs to be and does not pose any discrepancy in the scenarios of Side-Channel analysis. In this paper, Ascon is opted for the permutation as illustration.

Below we will analyze each component of ISAP and find out which component has a potential side-channel vulnerability. The ultimate goal is to recover the master key of Ascon based ISAP with 128 bit security parameter.

2.1.1 Authenticated Encryption

Authenticated encryption and decryption with associated data have the same structure in ISAP. In this section we focus on the encryption.

As illustrated in Figure 1, ISAP encryption takes three inputs: nonce N, master key K and message M. Nonce and master key is used to derive encryption key K_E^* using ISAP component ISAPRk. And K_E^* and nonce jointly forms the initial state of the Ascon permutation. Nonce N must vary from each encryption and is visible to the attacker. Any bit change on N will result in different K_E^* and different initial states thus affecting the ciphertexts C by avalanche effect.

What is visible to the attacker is N, M_i and C_i. Note that nonce should be assumed not to reuse in each encryption so initial states of permutation varies constantly, rendering the recovery of K_E^* impossible and meaningless. To achieve cipher text forgery, the attacker must be able to recover master key K to compute K_E^* using ISAPRk.

Due to the discarding of states variables in the end, the attacker will not be able to recover the input of final permutation p_E, thus K_E^* is invisible.

We'll dive into the ISAPRk component later.

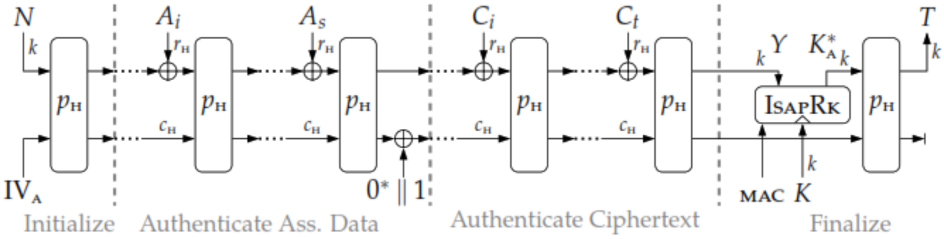

Figure 2: Message Authentication Code of ISAP

2.1.2 Message Authentication Code

As shown in Figure 2, the ISAPMAC takes nonce N, fixed public parameter IV_A, public associated data A, cipher C and master key K as inputs, generates tag T using part of the states and discards unused states in the end.

Due to the discarding of states variables in the end, the attacker will not be able to recover the input of final permutation p_H, thus K_A^* is invisible. By changing A and C, the attacker can change and compute ISAPRk parameter Y.

2.1.3 Rekey

Rekey component (ISAPRk) serves as a key derivation function using fixed master key. As we have discussed in 2.1.1 and 2.1.2, it is obvious that the attacker should target on ISAPRk to recover master key K.

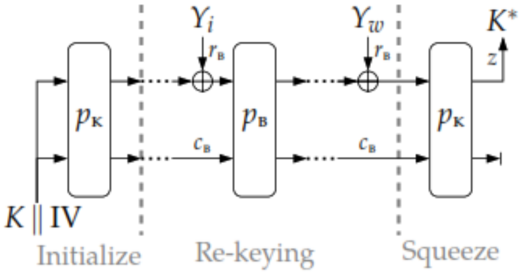

Figure 3: ISAPRk

The ISAPRk takes three inputs: the fixed master key K, the publicly visible initial vector IV and associated data Y. From both scenarios where ISAPRk is used, the attacker is always able to observe Y and IV, and always unable to observe K^*. Since the K and IV are both fixed, the state outputs by initialization stage are fixed as well. A recovery on initial states will also enable the attacker to compute K^*, thus compromising the security of ISAPEnc and ISAPMAC.

Y is split into blocks of size r_B, indicating only r_B bits of Y are absorbed by each permutation. In our scenario, $r_B = 1$ and the state bit length of 320.

Since the goal of the attacker is to recover K, she may manipulate Y and observe side-channel leakages on p_B and tries to recover the state variables.

2.2 Specify the Strategy

We will focus on the permutation p_B used in ISAPRk.

The non-linearity of Ascon is induced by the internal SBox construction. In the typical implementation, 320 bit state is split into 5 state variables of 64 bit, and data to be absorbed only influence the leading r_B bits.

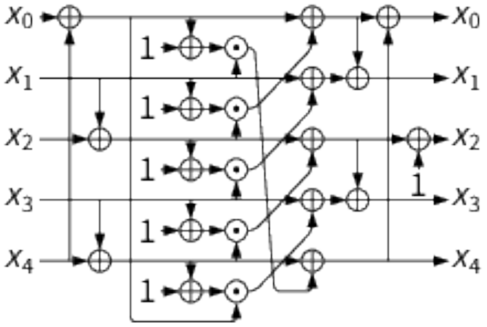

Figure 4: SBox of Ascon permutation

Before the SBox, the Ascon permutation only performs a linear constant addition to the state. And the constant is public so the attacker can calculate SBox input giving the data to be absorbed and initial state. As shown in Figure 4, state variables x_0 to x_4 to be passed to SBox are 64 bit each. All the operations are bit-wise, indicating the calculation is on bits of the same position of x_i.

For each absorption, only leading r_B bits of x_0 are randomizable. The attacker will not be able to recover trailing fixed $320 - r_B$ bits.

To recover the leading r_B bits of each x_i, the attacker can randomize leading r_B bits of Y and cause leading r_B bits of input x_0 to change and choose same bits of SBox output x_0 to x_5 as intermediate values.

Since SBox is a highly non-linear operation on input bits, any leakage will be amplified on hardware, which constitutes a target of side-channel analysis. Although only recover leading r_B bits of five state variables can be recovered, the attacker will gather sufficient information regarding of the permutation state after sufficient rounds of permutation. Having correctly recovered sufficient leading r_B bits of multiple intermediate states, the attacker can utilize a linear solver (like z3) to solve the initial state mathematically.

Our analysis strategy uses a r_B bit visible random value to recover $5 \times r_B$ fixed values by exploiting non-linear SBox which takes them as input, and the intermediate values are output leading r_B bits of five state variables. It is hard to achieve such 1 : 5 information ratio under side-channel analysis, but it's worth trying.

3 Experiments

3.1 Setting

We first need to download the firmware containing the C/ASM implementation of Ascon-ISAP into the devices flash memory. 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. Meanwhile, we use a high-precision electromagnetic probe to capture the electromagnetic power emitted from the device chip. The captured power 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 ISAP in the host computer for later assessment.

The hardware platform we use to flash code into is STM32F303RCT6 with Xilinx Kintex-7 FPGA. The power consumption traces are acquired by a high precision LeCroy

610Zi ossciliscope.

The assessment we have conducted for each implementation is listed in Table 1.

Table 1: Assessment scenarios

Implementation	Assessment Strategy
Software impl. by ISAP team	CPA
Hardware impl. by IAIK	CPA
Hardware impl. by Ruhr-University Bochum	CPA, t-test, χ^2-test

Since the associated data input in both ISAPEnc and ISAPMAC are visible to attacker, so the analysis on ISAPRk precedure are expected to present the same results on them. In our experiment, we decided to run ISAPEnc and collected nonce inputs and master key for further analysis.

The acquisition triggers were placed on entering ISAPRk for each implementation.

3.2 Software Implementation by ISAP team

CPA is a side-channel analysis technique to reveal the private keys using the power leakage of a cryptographic device. We conducted CPA on the ISAP software implementation.

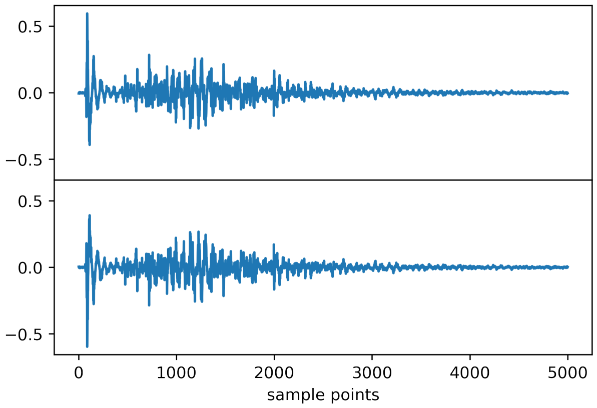

Figure 5: CPA on Software Implementation. The upper one is a random guess, and the lower one is the correct guess

The CPA result on random guess and correct guess of corresponding 5 bits of initial state does not present any notable discrepancy. So the software CPA can not effectively distinguish the correct initial state from wrong ones.

3.3 Hardware Implementation by IAIK

The results on figure 6 presents exactly the same phenomenon as software implementation. There is also no notable discrepancy between random guess and correct initial states.

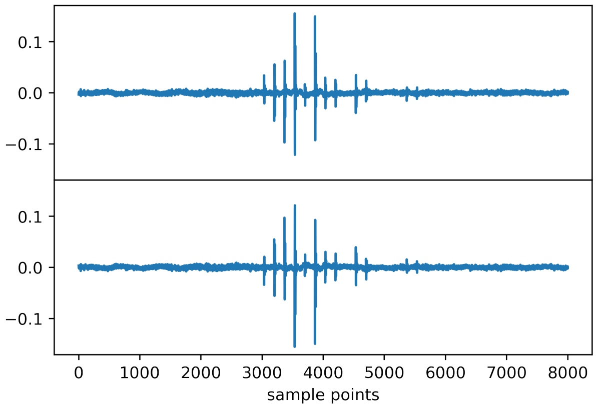

Figure 6: CPA on IAIK hardware implementation. The upper one is a random guess, and the lower one is the correct guess

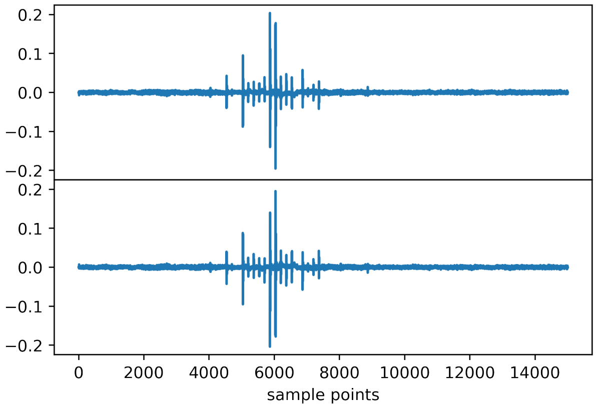

Figure 7: CPA on ISAP second order hardware implementation. The upper one is a random guess, and the lower one is the correct guess

3.4 Hardware implementation by Ruhr-University Bochum

3.4.1 CPA

The CPA result depicted in Figure 7 leads to the same conclusion that CPA can not make a correct guess value distinguishable from incorrect ones for ISAP ISAPRk procedure.

3.4.2 t-test and χ^2-test

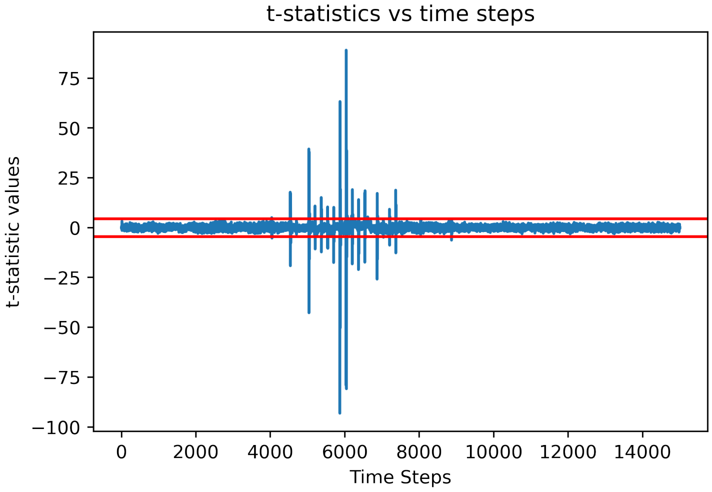

Figure 8: t-test result, the red line indicates the ±4.5 threshold

For a further leakage analysis, we conducted t-test and χ^2-test on collected hardware implementations.

Welchs t-test is a statistical hypothesis test used to compare the means of two groups, especially when the two groups have unequal sample sizes and variances. In terms of side-channel analysis, we can divide the power traces into two groups according to the dierence in intermediate values.

More precisely, when the private key is xed, we can divide the power traces of by a bit of selected intermediate SBox output bits. Here we select the second bit for illustration.

As shown in Figure 8, both t-test and χ^2-test are able to detect obvious leakage on ISAP hardware implementation.

The t-test and χ^2-test shows tremendous leakage but the CPA fails to distinguish correct state guesses. For CPA, the correct guess and incorrect guess plot are either exactly the same or symmetric against x-axis. And t-test result shown in Figure 8 looks like amplification of y-scale of figure 7. It is bizarre that CPA result and t-test result look the same shape. We leave the explanation of this to the next section.

4 Results

4.1 Analysis

The anomaly of CPA results and t-test drove us into reviewing the ISAP permutation construction. The data to be absorbed Y_i each permutation is a single bit, thus only affecting a single bit of total 320 bit state. The linearity of SBox resides in operations of bits of the same position in five state variables. When the five bits are all XORed with associate data Y, a side-channel analysis could take place because all the bits of Y will be spread through SBox output. But in our selected ISAP's scenario, only first bit is XORed. This greatly reduces the leakage induced by SBox. When the attacker chooses a five-bit state initial state to guess, the SBox output is only determined by the bit to be absorbed.

Let x_i be bits of initial state to be guessed, and y be the bit to be absorbed, b_i be the SBox output bits. When x_i is fixed, whether our guess is correct or not, the b_i is only determined by y. The $b_i = y \oplus c_i^1 + c_i^2$ where c_i^1 and c_i^2 are fixed combination of x_0 to x_5. When conducting side-channel analysis on b_i, we're actually conducting on single-bit XOR. The non-linearity of SBox is significantly reduced by only absorbing a single bit each permutation. Any leakage on b_i actually comes from bit y rather than initial state x_i. For a single bit absorption scenario, there is no exploitable non-linearity of ISAP.

CPA results of correct guessing value and incorrect ones are the same because the intermediate values of each guess, which only determined by nonce bit, are either the same or opposite. Traces divided by the selected intermediate bit are also actually divided by the nonce bit, thus showing tremendous leakage.

4.2 Conclusion

In this report we have shown that ISAP is resilient against DPA-based Side-Channel Analysis. The attacker is unable to find a proper attack target in all components of ISAP. We further show that the leakage-resilience critically depends on the choice of r_B in the ISAPRk component. r_B indicates how many bits of data to be absorbed in the permutation. A small choice of r_B will lead the power leakage of master key to be covered by the leakage of publicly visible data. As a conclusion, for Ascon-based ISAP, the data to be absorbed has to be longer than 64 bit to conduct side-channel analysis.As a result, the current ISAP algorithm does not present any side-channel leakage with regard to the master key.