ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-202302.00099

Differential Power Analysis of the Lightweight Authenticated Encryption Algorithm ASCON: Postprint

Authors: Li Pan, Wei Yongzhuang, Yongzhuang Wei

Date: 2023-02-14T00:00:00+00:00

Abstract

A differential power analysis method is proposed for the structure of the lightweight authenticated encryption algorithm ASCON. This method leverages the implementation characteristics of the algorithm' s S-box, employs the Hamming weight model as the power distinguisher function, groups power traces, and recovers the encryption master key. Furthermore, to address the "ghost peaks" that appear in DPA attacks, a power trace preprocessing method is presented, which first groups traces according to plaintext and computes their mean values before launching a DPA attack on the preprocessed traces. By collecting 1,500 power traces leaked from the sa permutation, 44 bits of the master key can be rapidly recovered. Moreover, the time required for a direct attack on raw traces is 21,849.8889 ms, whereas after introducing the preprocessing technique, the time required to attack preprocessed traces is 198.9113 ms, approximately 1/109 of the time needed for direct attack on raw traces.

Full Text

Abstract

This paper proposes a differential power analysis (DPA) method targeting the lightweight authenticated encryption algorithm ASCON. The method combines algorithmic characteristics with the Hamming weight model as the power consumption discrimination function to group power traces and recover the master encryption key. A power trace preprocessing technique is introduced that first groups traces by plaintext and computes their averages before launching the attack. Experimental results demonstrate that using 10,000 power traces, the master key can be rapidly recovered. Direct attack on original traces requires approximately 109 seconds, while attacking preprocessed traces needs only about

1 second—roughly 1/109 of the direct attack time.

Keywords: ASCON; lightweight authenticated encryption algorithm; differential power analysis; Hamming weight model; preprocessing

1 Introduction

In emerging domains such as distributed control systems and resource-constrained devices—including sensor networks and cyber-physical systems—devices require interconnection to collaboratively perform tasks [1-3]. Since most encryption algorithms are designed for desktop and server environments, they prove unsuitable for resource-constrained devices. In 2014, experts from Graz University of Technology and Infineon Technologies jointly designed the ASCON lightweight encryption algorithm [4]. ASCON achieves high security while facilitating rapid hardware and software implementation. In 2019, NIST initiated a global competition for lightweight cryptographic algorithms, and ASCON was selected as a finalist. In 2021, ASCON became one of the winning algorithms in the CAESAR competition [5].

While traditional mathematical analysis and fault injection attacks on ASCON have advanced [6-15], research on side-channel attacks—particularly against software implementations—remains limited. This paper investigates ASCON' s vulnerability to DPA attacks and proposes a preprocessing technique to significantly accelerate the attack process.

2 ASCON Algorithm Description

ASCON is a lightweight authenticated encryption algorithm based on sponge construction [4]. The algorithm operates in two modes: authenticated encryption and hashing. Three variants were submitted to NIST: ASCON-128, ASCON-128a, and ASCON-80pq. ASCON-80pq increases key length to enhance resistance against quantum key search. This paper focuses on ASCON-128 and ASCON-128a.

The 320-bit internal state s is organized as five 64-bit register words. The encryption process comprises four phases: initialization, associated data absorption, plaintext processing, and tag generation. During initialization, key K and nonce v enter the s permutation. The initial state is constructed as $init = K \,||\, v \,||\, c$, where c is the initial vector. Data processing employs the s_b permutation (with fewer rounds than s), while tag generation uses s for enhanced security.

ASCON utilizes two permutations: s (12 rounds) and s_b (6 or 8 rounds). The S-box is implemented using a slicing technique with five parallel S-boxes. Each round consists of constant addition, S-box substitution, and a linear diffusion layer derived from SHA-2 [16].

3 DPA Attack Methodology

3.1 Power Consumption Model

Kocher et al. [7] first exploited the dependency between cryptographic chip power consumption and processed intermediate values. For software implementations, the Hamming weight model effectively characterizes dynamic energy consumption [7]. When using a pre-charged bus, all lines are set to 0 before operand transmission. The linear relationship between power consumption at time j ($T(j)$) and data Hamming weight is:

$$T_i(j) = \mu \cdot \mathrm{HW}(D_i(j)) + \epsilon + \eta$$

where is a scaling factor, $\mathrm{HW}(\cdot)$ is the Hamming weight function, is a constant offset, and represents noise.

3.2 Attack Point Selection

To reduce computational complexity, the attack targets the first round of the initialization phase. Before the diffusion layer propagates values, intermediate data remains localized, creating an optimal attack point. The power discrimination function uses Hamming weight to reduce dependency on device-specific leakage characteristics and improve success rates.

3.3 Trace Preprocessing Technique

A novel preprocessing technique accelerates DPA attacks by exploiting the observation that power consumption (excluding random noise) remains constant when encrypting identical plaintexts. The method groups power traces by plaintext and computes mean traces for each group.

Algorithm 1: Trace Preprocessing

```
Input: n original power traces T, n plaintexts P (each with k bytes)
Output: l preprocessed traces T', l plaintext groups
1. Group traces by plaintext values
2. For each group, compute the mean trace
3. Return l mean traces T' and corresponding plaintext indices
```

After preprocessing, trace count reduces from n to l (the number of distinct plaintext values), with each trace index directly representing a plaintext value.

3.4 Key Recovery Process

Single Plaintext Attack: 1. For each key guess g, compute the Hamming weight of the targeted intermediate value 2. Partition traces into two groups based on a threshold (e.g., HW $< k/2 + 0.5$) 3. Calculate within-group mean traces 4. Compute absolute difference between group means 5. The key guess producing the maximum difference is the recovered key

Multiple Plaintext Attack: The procedure iterates over each key byte position, processing l preprocessed traces to recover k key bytes.

4 Experimental Results

4.1 Experimental Setup

The experimental environment is summarized in . The target is an STM32F407 microcontroller (Cortex-M4F) at 168 MHz. Power traces were captured using a PicoScope 3206D oscilloscope at 500 MS/s. ASCON was implemented in software using the slicing technique.

Fixed Parameters: - Key K: 0x000102030405060708090a0b0c0d0e0f - Associated data A: 0x00···00 - Plaintext P: 0x00···00 - Nonce v: Varied per encryption

Dataset: 10,000 traces, each with 50,000 sample points

4.2 Attack Results

The attack successfully recovered the complete 128-bit master key: r = 0x00000000001000080819bc1c196a1950

Targeting parallel S-box operations, preprocessing reduced effective traces from 10,000 to 256 (distinct plaintext byte values), decreasing analysis time by two orders of magnitude.

4.3 Complexity Analysis

[Figure 1: see original paper] compares time complexity between direct and preprocessed attacks:

- **Direct attack**: Time scales linearly with trace count, requiring ~109 seconds for 10,000 traces
- **Preprocessed attack**: Time remains constant at ~1 second, as preprocessed trace count is fixed at 256

The preprocessing technique achieves a 109× speedup while maintaining equivalent success rates.

5 Comparison with Related Work

Comparison with prior ASCON DPA research [6,8] is shown in :

Work	Platform	Implementation	Traces Required	Attack Time
Samwel et al. [6]	SAKURA-G	Hardware	100,000	Not reported

Work	Platform	Implementation	Traces Required	Attack Time
Gross et al. [8]	Simulated	Hardware	50,000	Not reported
This work	STM32F407	Software (sliced)	10,000	~1s (preprocessed)

Our approach requires significantly fewer traces and provides concrete timing results. While hardware implementations offer clean leakage profiles, this software-based attack demonstrates practical threats to real-world ASCON deployments.

6 Conclusion

This paper demonstrates a practical DPA attack against ASCON' s software implementation. By targeting the first s permutation round and employing trace preprocessing, we recovered the master key using 10,000 traces in ~1 second —improving upon direct attacks by two orders of magnitude. Experimental results confirm that ASCON' s sliced S-box implementation leaks exploitable information.

Future work will investigate efficient masking countermeasures that maintain ASCON' s lightweight properties while resisting DPA attacks.

References

[1] KHAN A, SALAH K. IoT security: Review, blockchain solutions, and challenges[J]. Future Generation Computer Systems, 2018, 82: 395-411.

[2] MAHMUD R, KOTAGIRI R, BUYYA R. Internet of Things: Challenges and opportunities[M]//Internet of Everything. Berlin, Heidelberg: Springer, 2018: 103-130.

[3] DOBRAUNIG C, EICHLSEDER M, MENDEL F. Ascon[EB/OL]. (2021-04-14)[2021-11-23]. https://csrc.nist.gov/projects/lightweight-cryptography/finalists.

[4] DOBRAUNIG C, EICHLSEDER M, MENDEL F, et al. Cryptanalysis of Ascon[C]//Topics in Cryptology –CT-RSA 2015. Berlin, Heidelberg: Springer, 2015: 371-387.

[5] SAMWEL N, DAEMEN J. DPA on hardware implementations of Ascon[C]//Computing Frontiers 2017. New York: ACM Press, 2017: 415-424.

[6] MANGARD S, OSWALD E, POPP T. Power analysis attacks: Revealing the secrets of smart cards[M]. Berlin, Heidelberg: Springer, 2010.

[7] GROSS H, WENGER E, DOBRAUNIG C, et al. Hardware implementations of Ascon[J]. Microprocessors and Microsystems, 2017, 52: 1-11.

[8] RAMEZANPOUR K, AMPADU P, DIEHL W. Statistical fault methodology for Ascon authenticated cipher[C]//2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). Piscataway, NJ: IEEE Press, 2019: 41-50.

[9] BAR-ON A, DUNKELMAN O, KELLER N, et al. DLCT: A tool for differential-linear cryptanalysis[C]//Advances in Cryptology –CRYPTO 2019. Berlin, Heidelberg: Springer, 2019: 313-342.

[10] SURYA R, SARKAR S. Diving into the differential-linear connectivity of Ascon[J]. IACR Transactions on Symmetric Cryptology, 2021(4): 74-99.

[11] ROHIT R, SARKAR S. Misuse-free key recovery attacks on 7-round Ascon[J]. IACR Transactions on Symmetric Cryptology, 2021(3): 102-136.

[12] JOSHI P, MAZUMDAR B. SSFA: Subset fault analysis on cipher[J]. Microelectronics Reliability, 2021: 114-155.

[13] BASEL H, JORGE S. Hardware security[M]. Berlin, Heidelberg: Springer, 2021: 69-85.

[14] QUYNH D. Secure hash standard[EB/OL]. (2020-02-27)[2021-12-23]. http://dx.doi.org/10.6028/NIST.FIPS.180-4.

[15] SIM M, JAP D, BHASIN S. DAPA: Differential analysis aided power attack on (non) linear feedback shift registers[C]//Cryptographic Hardware and Embedded Systems (CHES) 2020. Berlin, Heidelberg: Springer, 2020: 169-188.

[16] CHEN J-S, KYAW K. Normalized differential power analysis for ghost peaks mitigation[C]//2021 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway, NJ: IEEE Press, 2021: 1-5.

[17] FEI G, CHENG S, GONG L, et al. Correlation power analysis of higher-order masking implementations of WAGE[C]//Selected Areas in Cryptography (SAC) 2020. Berlin, Heidelberg: Springer, 2020: 593-614.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.