Side-Channel Collision Attacks against ASCON

Hao Zhang[†], Yiwen Gao[†✉], Yongbin Zhou[†‡], Jingdian Ming[†]

[†]*School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*
[‡]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
zhanghao33105@njust.edu.cn, yw.gao@hotmail.com, zhouyongbin@njust.edu.cn

***Abstract*—Side-channel attack poses a significant threat to the security of electronic devices, particularly IoT/AIoT terminals. By leveraging side-channel leakages, collision attacks can efficiently extract the secret keys from cryptographic devices while requiring considerably less computational effort. In this paper, we investigate side-channel collision attacks against ASCON, a lightweight crypto designed for resource-constrained devices, which has been standardized by the NIST. For the first time, we propose a side-channel key recovery attack against ASCON by identifying the collisions in the linear diffusion layer. Using Pearson correlation coefficient and Euclidean distance for internal collision detections, our attack successfully recovers the secret key with approximately 5,000 power traces from an 8-bit software implementation on an AVR device. To further reduce attack complexity, we introduce a novel metric, Locally-Weighted Sum (LWS), which focuses on the most likely points of leakage, thereby decreasing the number of required power traces for successful attack. Our experiment on the same target demonstrates that the LWS-based collision attack can recover the full secret key with approximately 3,000 power traces, a reduction of 40 percent. Our study indicates that ASCON is susceptible to side-channel collision attacks, and bitslice implementations remain vulnerable to such threats.**

***Index Terms*—Side-Channel Attack, Collision Attack, ASCON, Lightweight Cryptography, IoT**

I. Introduction

Side-channel leakages, such as power consumption, electromagnetic emanation, timing, inherently provide exploitable information. Unlike techniques such as Differential Power Analysis (DPA) [1], Correlation Power Analysis (CPA) [2], and Template Attacks [3], side-channel collision attacks exploit the similarity of leakage patterns that arise when identical operations are performed on the same intermediate value. By examining these patterns, attackers can establish correlations between internal states and the sensitive information, such as secret keys, without the need for intricate leakage models. These attacks represent a serious security threat, with documented success in compromising numerous real-world encryption systems.

The earliest side-channel collision attacks were demonstrated against DES [4] and AES [5]. Bogdanov later extended these techniques to other encryption schemes [6], introducing refined collision detection methods based on binary and ternary voting mechanisms [7]. As these attacks depend on high-quality power traces, enhancing collision detection in noisy environments has become a critical focus of research. Notable advancements include the stochastic collision attack proposed by Bruneau et al. [8], which employs statistical methods to improve collision detection under noisy conditions. Glowacz and Grosso developed the Optimal Collision Attack (OCA), leveraging maximum likelihood techniques to distinguish collisions efficiently by optimizing the likelihood function for side-channel leakages [9]. More recently, Long et al. introduced the Collision-Paired Correlation Attack (CPCA), which effectively pairs side-channel samples to exploit low-noise environments [10]. Side-channel collision attacks have demonstrated effectiveness against both unprotected and certain masked implementations. For instance, Moradi, Guilley and Heuser applied these attacks to masked algorithms with residual first-order leakage [11], while Staib and Moradi utilized deep learning approaches to exploit side-channel collisions in non-profiled scenarios, enabling key recovery even in black-box settings [12].

Since the NIST confirmed ASCON as a new standard, the lightweight cipher has garnered extensive attention. Thanks to its efficiency in resource-constrained environments, ASCON is considered a key security solution for future IoT and embedded systems. Although there has been considerable research on side-channel collision attacks, most studies so far have focused on conventional encryption algorithms, such as DES and AES. Given the growing importance of ASCON in IoT and embedded systems, investigating its susceptibility to side-channel attacks—particularly side-channel collision attacks—has become a critical research priority.

In this paper, we provide a comprehensive analysis of ASCON's internal structure and operational mechanisms, identifying vulnerabilities that could be exploited by side-channel collision attacks. Our findings demonstrate the feasibility of the attacks and underscore the potential security threats they pose to ASCON.

Our contributions are summerized as follows:

- We present side-channel collision attacks to ASCON, identifying internal collision vulnerabilities through theoretical analysis. By exploiting these vulnerabilities, we analyze the resulting power consumption patterns, establish relationships between subkeys, and successfully recover ASCON's 128-bit secret key.
- We introduce the novel locally-weighted sum (LWS) metric. Experimental results show that, compared to traditional methods based on Euclidean distance and correlation coefficients, LWS achieves higher success rates, particularly in scenarios with limited power traces.

The remainder of this paper is organized as follows: In Section II, we provide background information on side-channel collision attacks and the ASCON algorithm. In Section III, we present a collision analysis of ASCON's algorithmic structure

978-3-9826741-0-0/DATE25/© 2025 EDAA

and algebraic properties and utilize the LWS metric. In Section IV, we detail the experimental results. Finally, we conclude the paper in Section V.

II. Background

A. Side-channel Collision Attack

In cryptographic algorithms, an internal collision occurs when an internal function produces identical outputs for different inputs. This phenomenon is common in many-to-one mapping functions, such as the S-box in the DES algorithm, which maps a 6-bit input to a 4-bit output, inevitably leading to collisions. Although the S-box in the AES algorithm is bijective, providing a one-to-one mapping, collisions can still occur across different S-box instances when they are fed the same plaintext byte value as input, regardless of whether they belong to the same encryption process. Internal collisions occur exclusively within the internal functions of encryption and decryption processes, making them undetectable at the input or output stages. However, these collisions can reveal partial information about the cryptographic key, making their detection and analysis crucial for key recovery.

Taking the side-channel collision attack on the AES S-box as an example, we search for collisions at the round key addition in the first round of the AES algorithm. Following this operation, the state after round key addition undergoes the byte substitution operation. The power trace of the S-box operation during this process can be expressed in the following form:

$$x_i = \varphi(P_i \oplus K_i) + N_i \tag{1}$$

where x_i represents the power trace at the i-th S-box position (i=1,2,3,...,16). The function φ is a deterministic but unknown function that models the power leakage resulting from the S-box lookup operation. It is deterministic because each S-box lookup operation is consistent, with specific inputs producing corresponding lookup results. However, its exact form remains unknown. P_i and K_i denote the plaintext byte value and subkey corresponding to the i-th S-box position, respectively. N_i represents additive Gaussian noise [13], which accompanies all power leakages. We mitigate this noise effect by averaging multiple measurements. When a collision occurs, the power traces at the i-th and j-th S-box positions follow the equations (j=1,2,3,...,16):

$$P_i \oplus K_i = P_j \oplus K_j \tag{2}$$

$$\delta^* = P_i \oplus P_j = K_i \oplus K_j \tag{3}$$

For convenience, we denote a potential collision scenario as $(p, p \oplus \delta)$, where p represents the plaintext and δ is the difference between plaintext pairs. We iterate through all possible values of δ, and a collision occurs only when δ matches the specific collision value δ^*, satisfying the following equation:

$$P_j \oplus K_j = (p \oplus \delta) \oplus (K_i \oplus \delta^*) = P_i \oplus K_i \tag{4}$$

When a collision occurs, the power traces at the two S-box positions become more similar, resulting in a smaller difference between the traces. In this context, δ represents the collision value we aim to identify. During the execution of the encryption algorithm, numerous internal collisions frequently arise. The key to a successful side-channel collision attack lies in identifying collisions that can reveal the relationships between subkeys; otherwise, merely detecting internal collisions is inconsequential.

B. ASCON

From the recommended parameters, we utilize ASCON-128; henceforth, when referring to ASCON, we specifically mean ASCON-128. This variant of ASCON employs 128-bit keys and nonces.

The authenticated encryption algorithm ASCON employs a duplex construction. Figure 1 illustrates the encryption process of ASCON, which takes four inputs: plaintext P_i, associated data A_i, nonce N, and a key K. The block cipher produces two outputs: ciphertext C_i and a tag T. The permutation is denoted by p^a and p^b, where $a = 12$ and $b = 8$, representing the number of rounds. The nonce is public. The tag is used during decryption to authenticate the ciphertext. During decryption, the algorithm uses the tag as an input and outputs the plaintext along with the result of the tag validation. If the tag is invalid, no output is returned.

The internal state of the ASCON algorithm consists of 320 bits, divided into five 64-bit registers named x_0 to x_4. During initialization, a 64-bit constant IV is loaded into register x_0. Registers x_1 and x_2 are assigned the key, while the variable nonce is stored in registers x_3 and x_4. The associated data and plaintext are padded to ensure their lengths are multiples of 64 bits. Once the internal state is initialized, permutation p^a is applied. Subsequently, the optional associated data is absorbed into the state, followed by the absorption of the plaintext. After processing each block of associated data and plaintext, p^b is applied, except for the final block of plaintext. During the algorithm's finalization phase, p^a is applied once more, and a tag is generated.

The round function, or permutation, in ASCON involves three steps. First, a round constant is added to register x_2 at the lower indices. The second step involves a nonlinear five-bit S-box with an algebraic degree of two, which takes one bit from each register, as illustrated in Figure 2, and replaces it with the S-box output. ASCON's S-box operation employs the bitslice method to achieve efficient parallel processing. The final step is the linear diffusion layer, where each register undergoes two rotations and is XORed with itself. The expressions for the linear diffusion layer are provided in Figure 3.

III. Our Attacks

A. Collision Points and Algebraic Analysis

Existing side-channel collision attacks against AES predominantly focus on analyzing and exploiting the S-box, given its role as a repeatedly executed nonlinear operation. Compared to linear operations, the power traces for different inputs in the S-box are more distinguishable. Furthermore, the repetitive nature

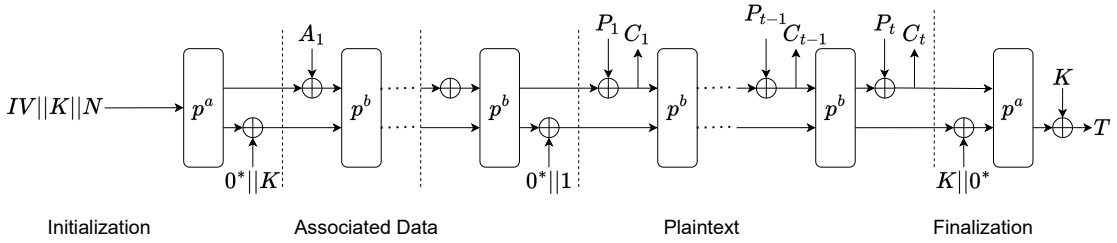

Fig. 1: An Overview of ASCON Encryption.

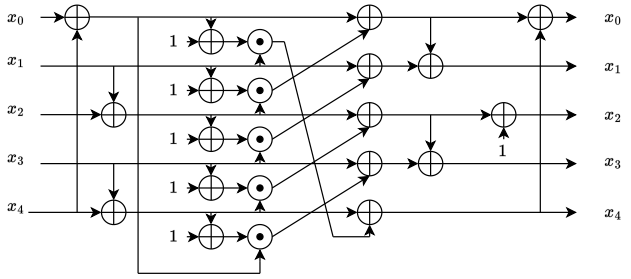

Fig. 2: Substitution Layer with 5-bit S-box in ASCON.

$$x_0 \leftarrow x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$$
$$x_1 \leftarrow x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$$
$$x_2 \leftarrow x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$$
$$x_3 \leftarrow x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$$
$$x_4 \leftarrow x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$$

Fig. 3: Linear Layer with 64-bit Diffusion in ASCON.

of the S-box allows for the effective utilization of internal collisions, thereby enhancing the efficiency and success rate of the attack. However, applying this method to the ASCON algorithm presents new challenges. The S-box operation in ASCON is not a traditional S-box but rather employs the bitslice method, achieving the effect of an S-box through parallel operations on several registers. This design results in a high degree of overlap between S-box operations, making it infeasible to apply traditional side-channel collision attacks to this S-box. Specifically, the ASCON algorithm utilizes 64 parallel S-box operations, rendering it unrealistic to find collisions in a vertical S-box manner. This parallelized design complicates the attack, as distinguishing between individual S-box operations becomes more challenging.

Given these challenges, this paper shifts the focus from S-box operations to the linear diffusion operations that follow the S-box in the analysis of side-channel collision attacks. These linear diffusion operations are implemented relatively independently within registers, making them theoretically exploitable.

In the ASCON algorithm, the elements of the initial state matrix have different origins: x_0 is a constant, x_1 and x_2 are parts of the key k, and x_0, x_1, and x_2 are fixed and invariant. In contrast, x_3 and x_4 are random numbers that can be adjusted. It is important to note that the subsequent analysis is based on individual bits, i.e., one bit in each 64-bit register. We attempt to find collisions in the linear diffusion operation (P_l) of the first round of the ASCON algorithm. Suppose two different inputs result in a collision in a certain X_i after the state matrix P_s, which means the two X_i values are equal. In that case, the power traces resulting from the diffusion operation on X_i should be highly similar. Figure 4 shows the Algebraic Normal Form (ANF) of the P_s operation, illustrating the involvement of both fixed and non-fixed parts. Here, y_i represents the result of applying the P_s operation to x_i. Take y_0 as an example: if the two inputs are unequal in the non-fixed part but result in equal y_0 values after the computation, it indicates an internal collision. In such cases, since y_0 values are equal, the diffusion operations performed on y_0 would be identical, and the power traces resulting from these diffusion operations would theoretically be highly similar.

$$y_{0,i} = x_{4,i}x_{1,i} \oplus x_{3,i} \oplus x_{2,i}x_{1,i} \oplus x_{2,i} \oplus x_{1,i}x_{0,i} \oplus x_{1,i} \oplus x_{0,i}$$
$$y_{1,i} = x_{4,i} \oplus x_{3,i}x_{2,i} \oplus x_{3,i}x_{1,i} \oplus x_{3,i} \oplus x_{2,i}x_{1,i} \oplus x_{2,i} \oplus x_{1,i} \oplus x_{0,i}$$
$$y_{2,i} = x_{4,i}x_{3,i} \oplus x_{4,i} \oplus x_{2,i} \oplus x_{1,i} \oplus 1$$
$$y_{3,i} = x_{4,i}x_{0,i} \oplus x_{4,i} \oplus x_{3,i}x_{0,i} \oplus x_{3,i} \oplus x_{2,i} \oplus x_{1,i} \oplus x_{0,i}$$
$$y_{4,i} = x_{4,i}x_{1,i} \oplus x_{4,i} \oplus x_{3,i} \oplus x_{1,i}x_{0,i} \oplus x_{1,i}$$

Fig. 4: The ANF of the S-box Layer.

TABLE I: Results of y_0^* for Different Values of x_3 and x_4.

x_3	x_4	y_0^*
0	0	0
0	1	$x_1 \oplus 0$
1	0	1
1	1	$x_1 \oplus 1$

To determine the specific requirements and conditions for collisions, we conduct a detailed analysis of several linear diffusion operations in the ASCON algorithm. Using the linear diffusion operation of the first y_0 as an example, we illustrate a specific collision scenario.

First, we isolate the fixed parts of y_0, which are x_0, x_1, and x_2, and retain the non-fixed parts related to x_3 and x_4. This yields the expression $y_0^* = x_4 \cdot x_1 \oplus x_3$. Here, y_i^* represents a new expression after removing the fixed parts. When y_i^* collides, y_i also collides. For the corresponding bit positions, each bit in x_3 and x_4 can take on the values 0 or 1. Based on this equation, a table can be constructed to analyze the results of these four scenarios, as shown in Table I.

By examining Table I, we observe that there are four possible input combinations, but the output consists of only one bit, resulting in two possible outcomes: 0 and 1. Among these four results, there are two instances of 0 and two instances of 1. Mathematically, by analyzing these four results and performing pairwise combinations, we obtain six possible pairing results. Theoretically, two of these pairing results will be identical, indicating a collision (i.e., pairs of 0 and 0, or 1 and 1), while the remaining four pairing results will be different, indicating no collision (i.e., one side is 0 and the other is 1). This is illustrated in Figure 5, where solid lines represent collisions and dashed lines represent no collisions.

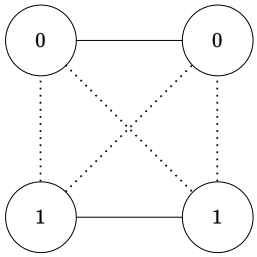

Fig. 5: The Collision of Four Results.

TABLE II: Results of $y_0^*, y_1^*, y_2^*, y_3^*, y_4^*$ for Different Values of x_3 and x_4.

x_3	x_4	y_0^*	y_1^*	y_2^*	y_3^*	y_4^*
0	0	0	0	0	0	0
0	1	$x_1 \oplus 0$	1	1	$x_0 \oplus 1$	$x_1 \oplus 1$
1	0	1	$x_2 \oplus x_1 \oplus 1$	0	$x_0 \oplus 1$	1
1	1	$x_1 \oplus 1$	$x_2 \oplus x_1 \oplus 0$	0	0	$x_1 \oplus 0$

To achieve a single-bit collision, let the colliding bit position be a (where a ranges from 1 to 64). We need to ensure that all bit positions except the a-th bit are identical, meaning that $x_{3,i}$ and $x_{4,i}$ are the same for both inputs (for $i \neq a$). Since the inputs are identical, y_0 will also be identical at these bit positions. Then, by changing the inputs at $x_{3,a}$ and $x_{4,a}$ so that y_0^* is equal in both cases, then y_0 will be equal for both inputs. This demonstrates that a collision has been found.

Set all bits of $x_{3,i}$ and $x_{4,i}$ to 0, except for the a-th bit.

Inputs:

- *Input-1*: $x_{3,a} = 0$, $x_{4,a} = 1$
- *Input-2*: $x_{3,a} = 1$, $x_{4,a} = 0$
- *Input-3*: $x_{3,a} = 1$, $x_{4,a} = 1$

Assuming that a collision occurs between *Input-1* and *Input-2*, and the correlation coefficient between *Input-1* and *Input-2* is higher than that between *Input-2* and *Input-3*, then according to Table I, $1 = x_1 \oplus 0$ implies $x_1 = 1$. This allows us to recover a specific bit of the key within the first 64 bits. By systematically performing bit-by-bit collisions in this manner, the entire 64-bit key can be recovered. The same approach can be applied to analyze collisions for other linear diffusion operations. Among these three inputs, selecting any two distinct inputs (for example, *Input-1* and *Input-2*, or *Input-1* and *Input-3*) will inevitably yield a colliding pair.

By analyzing the Table II, it is evident that collisions can be found at y_0 and y_4, allowing us to determine x_1. Subsequently, collisions at y_1 can reveal the relationship between x_1 and x_2, and using the previously determined x_1, we can deduce x_2. Collisions at y_2 and y_3 are unrelated to x_1 and x_2, and thus cannot be utilized.

B. Verification of Collision

Similar to how the sixteen S-box operations in the AES algorithm produce sixteen consecutive, clearly distinguishable patterns in the power traces, the five linear diffusion operations in the ASCON algorithm also yield five consecutive and obvious patterns in the power traces. This feature is helpful for side-channel collision attacks in precisely identifying the specific location to attack. As shown in Figure 6, the two differently colored boxes correspond to the first two linear diffusion operations. Consequently, manual segmentation of the power traces can be performed with relative ease.

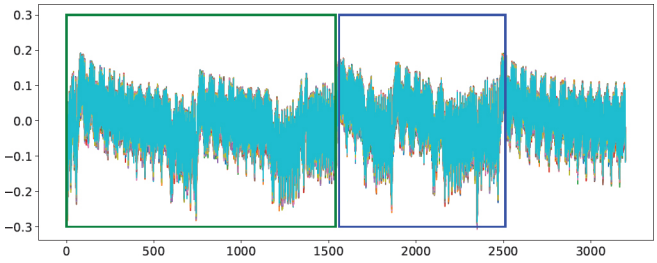

Fig. 6: Power Traces from ASCON Encryption on XMEGA.

It is important to clarify that collisions and non-collisions between power traces are determined by comparison. We must identify subtle variations among highly similar power traces. Therefore, the quality of the power traces is critical. The less noise in the traces, the fewer measurements are needed for a successful side-channel collision attack. Currently, averaging multiple power traces under the same input is a practical method for noise reduction.

Taking the first bit as an example, we input four different combinations of nonce values where the first bits of x_3 and x_4 are 00, 01, 10, and 11, respectively, while keeping the other bit positions unchanged. After averaging the power traces to reduce noise, we perform pairwise correlation analysis. Subsequently, four significantly lower correlation coefficients and two significantly higher correlation coefficients can be observed.

By subtracting each pair of noise-reduced power traces and taking the absolute value of the results, we can more intuitively

distinguish collisions from non-collisions. Power traces with collisions are more similar and thus yield results closer to zero, whereas non-colliding traces exhibit larger differences.

C. Collision Detection Based on Correlation Coefficient

Through analysis, each bit requires only three inputs to distinguish a collision. For the i-th bit, the input combinations for $x_{3,i}$ and $x_{4,i}$ are 01, 10, and 11, with all other bits set to 0, where i ranges from 1 to 64. We compare their correlation coefficients to determine whether a collision has occurred. The combination with a relatively high correlation coefficient is considered to collide. For the first 64 bits of the key, side-channel collisions in the first linear diffusion operation are sufficient to recover the key. For the second 64 bits, by conducting side-channel collisions in the second linear diffusion operation, we can obtain the XOR relationship between the first and second 64 bits of the key. Combining this relationship with the first 64 bits recovered in the previous step, we can deduce the entire 128-bit key.

D. Locally-Weighted Sum

In this paper, we propose a novel side-channel collision attack detection method based on LWS. This method subtracts corresponding points of two power traces, takes their absolute differences, and sorts them in descending order. A specific range of the largest differences, such as from the 5th to the 30th or from the 5th to the 50th largest values, is selected for weighted summation, where larger values receive greater weights. We exclude the first five points to mitigate the impact of extreme noise. Unlike the traditional Pearson correlation coefficient approach, LWS assigns different weights to each key point, giving greater emphasis to points with larger differences and thus increasing their influence on the final decision. This weighted strategy better highlights the critical portions of leaked information while ignoring noisy data points, thereby improving detection accuracy and robustness.

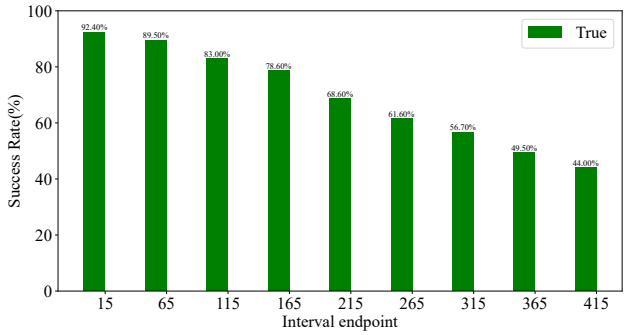

Fig. 7: Success Rate Variation Across Different Interval Endpoints.

Figure 7 shows how the interval size selection affects the success rate of key recovery. Noise was reduced by averaging 15 power traces for each input, and the attack was conducted on the first 64 bits of the key. We select the fifth-largest absolute difference as the starting point, and the x-axis values in Figure 7 represent the end of the range. For example, the bar corresponding to an x-axis value of 15 indicates a range from the 5th to the 15th largest absolute differences.

Based on these experimental results, we conclude that LWS effectively focuses on the most likely leakage points. Therefore, choosing an appropriate range of weighting points is crucial for practical applications. LWS performs better when the number of weighting points is selected judiciously. In some cases, excessively expanding the weighting range may hinder key recovery, as the increased influence of noise can reduce accuracy.

IV. Experiment

A. Setup

We employ the ChipWhisperer-Pro CW1200 as our testbed for collecting power traces, with the target board based on an AVR XMEGA microcontroller. The ASCON [14] implementation serves as the attack target, and our investigation focuses on the first two linear diffusion operations in the first round. The plaintexts used for encryption are described in Section III, where the bit under attack undergoes a combination change in both x_3 and x_4, while all other bits remain fixed.

B. Attack on Single Bit

We first perform side-channel collision attacks on individual bits of ASCON using the method described in Section III. We set the required plaintext input, use a specific number of power traces for averaging to reduce noise, and then compare the corresponding correlation coefficients. The success rate for each of the 128 bits is shown in Figure 8. Taking the first bit of the key as an example, we found that when the number of power traces used for averaging reaches 10, the success rate of the attack has already reached 90%. When the number reaches 20, the success rate reaches 100%. Due to the influence of random noise, the success rate of certain bits may not be as high as that of the first bit, and it is the bits with lower success rates that increase the difficulty of recovering the entire key.

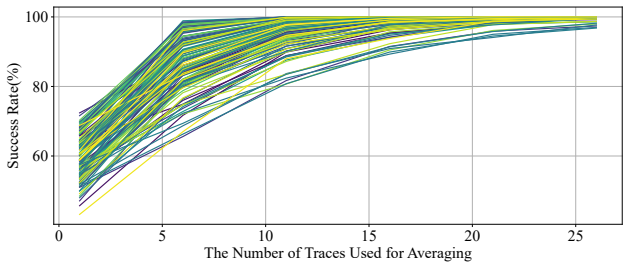

Fig. 8: Success Rate of Single-Bit Side-Channel Collision Attacks on Each of 128-bit ASCON Secret Key with Noise Reduction Using n Power Traces.

C. Attack on All Bits

Analysis of Table II reveals that inputs where both x_3 and x_4 are zero can be utilized in every comparison. Therefore, using these power traces can reduce the required number of power traces by approximately one-third. Figure 9 shows the success rate of recovering all keys. Subsequently, we conducted experiments with additional random keys, validating the effectiveness of the proposed method.

D. Comparison of Three Methods

Figure 10 compares the performance of LWS method with those based on Euclidean distance and correlation coefficients. The success rates of the methods based on correlation coefficients and Euclidean distance are comparable. However, when using fewer power traces, the success rate of LWS rises rapidly, approaching 100%, significantly outperforming the other two methods. The advantage of LWS is particularly evident when handling smaller numbers of power traces, indicating that LWS is more effective at capturing valuable leakage information, which confers a distinct advantage in practical attacks.

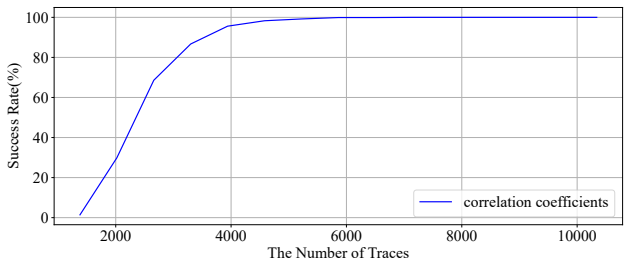

Fig. 9: Success Rate of Recovering All Subkeys.

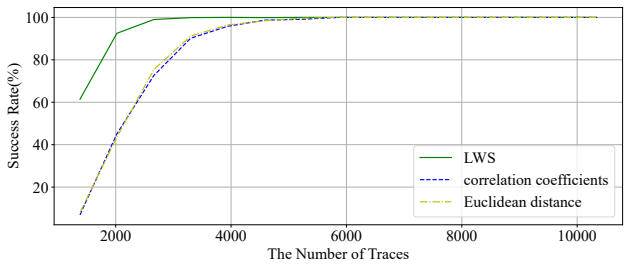

Fig. 10: Comparison of Success Rates Between LWS, Correlation Coefficients, and Euclidean Distance.

V. Conclusion

This paper presents the first application of side-channel collision attacks on the ASCON algorithm. Through theoretical analysis and experimental validation, we demonstrate the feasibility of conducting side-channel collision attacks on ASCON. By targeting the first two linear diffusion operations in the first round of ASCON, we successfully recovered the complete 128-bit key. Side-channel collision attacks require high-quality power traces and often necessitate averaging multiple traces for noise reduction, enabling better distinction between collision and non-collision traces. We introduce a novel metric for collision detection, called LWS. By selecting points with large differences after trace subtraction and applying weighted processing, LWS focuses on the most likely leakage points, thereby enhancing the efficiency of key recovery. In certain specific scenarios, LWS exhibits higher attack efficiency.

The S-box operations of ASCON utilize a bitslice approach, enhancing resistance to side-channel attacks while reducing the S-box size. Extending bitslice strategies to the linear diffusion phase could further strengthen ASCON's defense against side-channel collision attacks without necessitating masking strategies.

Currently, our work focuses on attacking the unprotected ASCON algorithm. The next phase will involve side-channel collision attacks on ASCON implementations with masking defenses. Given the success of such attacks on masked AES, this phase is highly promising. Additionally, ASCON plays a crucial role in the security of future IoT applications, making realistic side-channel collision attacks on ASCON a highly meaningful research endeavor.

Acknowledgments

This work is supported in part by National Key R & D Program of China (No. 2022YFB3103800), National Natural Science Foundation of China (No. 62202231, No. 62202230, No. 62302224, No. 62302226) and China Postdoctoral Science Foundation (No. 2023M741709).

References

[1] P. Kocher, "Differential power analysis," in *Proc. Advances in Cryptology (CRYPTO'99)*, 1999.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings 6*, pp. 16–29, Springer, 2004.

[3] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop Redwood Shores, CA, USA, August 13–15, 2002 Revised Papers 4*, pp. 13–28, Springer, 2003.

[4] K. Schramm, T. Wollinger, and C. Paar, "A new class of collision attacks and its application to des," in *Fast Software Encryption: 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers 10*, pp. 206–222, Springer, 2003.

[5] K. Schramm, G. Leander, P. Felke, and C. Paar, "A collision-attack on aes: Combining side channel-and differential-attack," in *Cryptographic Hardware and Embedded Systems-CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings 6*, pp. 163–175, Springer, 2004.

[6] A. Bogdanov, "Improved side-channel collision attacks on aes," in *Selected Areas in Cryptography: 14th International Workshop, SAC 2007, Ottawa, Canada, August 16-17, 2007, Revised Selected Papers 14*, pp. 84–95, Springer, 2007.

[7] A. Bogdanov, "Multiple-differential side-channel collision attacks on aes," in *Cryptographic Hardware and Embedded Systems–CHES 2008: 10th International Workshop, Washington, DC, USA, August 10-13, 2008. Proceedings 10*, pp. 30–44, Springer, 2008.

[8] N. Bruneau, C. Carlet, S. Guilley, A. Heuser, E. Prouff, and O. Rioul, "Stochastic collision attack," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, pp. 2090–2104, 2017.

[9] C. Glowacz and V. Grosso, "Optimal collision side-channel attacks," in *Smart Card Research and Advanced Applications: 18th International Conference, CARDIS 2019, Prague, Czech Republic, November 11–13, 2019, Revised Selected Papers 18*, pp. 126–140, Springer, 2020.

[10] J. Long, C. Ou, Y. Ma, Y. Fan, H. Chen, and S. Zheng, "How to launch a powerful side-channel collision attack?," *IEEE Transactions on Computers*, 2023.

[11] A. Moradi, S. Guilley, and A. Heuser, "Detecting hidden leakages," in *Applied Cryptography and Network Security: 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings 12*, pp. 324–342, Springer, 2014.

[12] M. Staib and A. Moradi, "Deep learning side-channel collision attack," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2023, no. 3, pp. 422–444, 2023.

[13] S. Mangard, E. Oswald, and T. Popp, *Power analysis attacks: Revealing the secrets of smart cards*, vol. 31. Springer Science & Business Media, 2008.

[14] Ascon128v12. https://github.com/ascon/ascon-c/tree/main/crypto_aead/ascon128v12/ref.