IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2024, No. 2, pp. 118–140. DOI:10.46586/tosc.v2024.i2.118-140

Improved Conditional Cube Attacks on Ascon AEADs in Nonce-Respecting Settings

with a Break-Fix Strategy

Kai Hu[1,2,3]

[1] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, China
kai.hu@sdu.edu.cn

[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore

[3] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China

Abstract. The best-known distinguisher on 7-round Ascon-128 and Ascon-128a AEAD uses a 60-dimensional cube where the nonce bits are set to be equal in the third and fourth rows of the Ascon state during initialization (Rohit et al. ToSC 2021/1). It was not known how to use this distinguisher to mount key-recovery attacks. In this paper, we investigate this problem using a new strategy called *break-fix* for the conditional cube attack. The idea is to introduce slightly-modified cubes which increase the degrees of 7-round output bits to be more than 59 (break phase) and then find key conditions which can bring the degree back to 59 (fix phase). Using this idea, key-recovery attacks on 7-round Ascon-128, Ascon-128a and Ascon-80pq are proposed. The attacks have better time/memory complexities than the existing attacks, and in some cases improve the weak-key attacks as well.

Keywords: Ascon · AEAD · Conditional Cube Attack

1 Introduction

Ascon [DEMS21], designed by Dobraunig, Eichlseder, Mendel, and Schläffer, is a family of lightweight Authenticated Encryptions with Associated Data (AEAD) and hash functions. Ascon has been selected as the NIST lightweight cryptography (LWC) standard recently [Ann], so it is crucial to understand its security in more depth.

This paper studies the security of the Ascon-AEADs against conditional cube attacks. The cube attack was originally proposed by Dinur and Shamir on stream ciphers [DS09]. Huang *et al.* adapted it to analyze the security of the permutation-based ciphers such as the Keccak keyed mode [HWX+17]. The technique chooses a set of input variables called the cube variables and observes the value of the coefficient of the maximal-degree term of these cube variables, a.k.a. superpoly. By carefully selecting cube variables, the value of a superpoly can be controlled by some specific key equations, so it can reveal some key information. This technique has been widely used in analyzing the sponge-based AEADs such as [LBDW17, DLWQ17, SGSL18, SG18].

The original version of Ascon has two AEADs: the Ascon-128 (the primary recommendation) and Ascon-128a. In its version 1.2, a new variant Ascon-80pq was added to increase the resistance against the quantum key-search. Most attacks were developed for Ascon-128 but are also applicable to the other members. Analyses of Ascon-AEADs can be divided into several categories according to which part is attacked. For the initialization phase composed of 12 rounds, the most effective attack is the cube-like attack. In [DEMS15],

Received: 2024-03-01 Accepted: 2024-05-01 Published: 2024-06-18

the designers proposed a cube key-recovery attack on up to 6 rounds with the *borderline cube* technique where the superpoly of a specific cube involves only a corresponding subpart of the key bits. In [LDW17], Li *et al.* improved the attacking record to 7 rounds using the conditional cube attack. This attack is based on a 65-dimensional cube, which means its data complexity cannot be smaller than 2^{64}, which is the security claim set by the designers [DEMS21, Chapter 2, Page 9]. The first attack with 2^{64} data complexity was proposed by Rohit *et al.* [RHSS21] which was also a cube attack where a *partial polynomial multiplication* and the borderline cube technique were leveraged to efficiently compute the superpoly. In the same paper, the authors also gave a cube distinguisher whose dimension is only 60. In [RS21], Rohit and Sarkar further studied the weak-key properties of the Ascon initialization. More efficient distinguishing and key-recovery attacks were found up to 7 rounds (when the weak-key conditions are satisfied). Besides the cube-like attacks, there are also (higher-order) differential-linear attacks on the Ascon-AEAD initialization up to 5 rounds with better complexities [Tez20, DEMS15, LLL21, HPTY22]. All these attacks on the initialization phase are nonce-respecting.

When it comes to the encryption phase, most attacks are nonce-misuse, i.e., they assume that a single nonce will be used for several initializations with the same key (which is forbidden by the designers' security claims). The state recovery is usually the first target of attacks on the encryption phase such as in [LZWW17, BCP22, CHKT23]. There are also some attacks on the finalization phase, such as the forgery attack in [GPT21]. A complete enumeration of attacks on Ascon can be found in the latest NIST report [TMC+23].

Of these attacks, the most relevant to this paper is the cube distinguisher for the 7-round Ascon initialization in [RHSS21]. This distinguisher uses a special construction that requires the first 64 bits and the last 64 bits of the nonce to be equal. Then, a division property model [HLM+20, Tod15] finds that the algebraic degrees of the 7-round output bits are at most 59. Unfortunately, it was not known how to utilize this distinguisher to mount a key-recovery attack. On the one hand, its specific structure does not satisfy the borderline cube property [DEMS15], thus the superpoly (according to the current theory) involves all key variables, which makes a cube attack based on it impossible. On the other hand, the cube variables in this structure spread fast, such that it is difficult to detect a set of key conditions that can separate one variable from the others to stop the maximal-degree (59-degree) cube term from appearing, as in other conditional cube attacks.

Our contributions. The previous conditional cube attacks [LBDW17, DLWQ17, SGSL18, SG18, LDW17] tend to use an *elimination* strategy: there should be a d-degree term in the superpoly, then by carefully selecting a set of cube variables, the d-degree term can be eliminated when some (simple) key conditions are satisfied. This strategy does not work for the 60-dimensional cube distinguisher given in [RHSS21]. To address this problem, we introduce a new strategy called *break-fix*. In the break phase, we break the special structure of the 60-dimensional cube (by introducing a slightly modified cube structure), to make the algebraic degrees of the output bits to be greater than 59. Then, in the fix phase, we identify a set of key conditions that fixes the changes caused by the "broken" cube structure and the algebraic degrees can return to 59 again. During the break-fix process, by observing if the algebraic degrees go back to 59, corresponding key information can be recovered.

Our first new conditional cube attack focuses on Ascon-128, but it also applies to Ascon-128a. We introduce 64 proper structures each of which breaks the original 60-dimensional cube structure. By identifying the key conditions that can fix the changes, we manage to perform the key-recovery attacks. For about $2^{127.97}$ (out of 2^{128}) keys, our attack can recover the 128-bit key with 2^{70} data and $2^{72.4}$ time complexities. For the remaining keys, our attack works with 2^{70} data complexity and at worst $2^{104.7}$ time complexity.

Considering the data limit set by the designers that one key can at most protect 2^{64} blocks of data, we also adapt our attack to a weak-key version to meet the requirement. The first attack works for a $2^{127.3}$-size weak-key space and can recover all key bits with a 2^{64} data complexity and 2^{120} time complexity. The second attack works for a 2^{125}-size weak-key space and can recover all key bits with a $2^{63.32}$ data complexity and 2^{115} time complexity. The memory cost of our attacks is negligible.

Our second conditional cube attack targets the Ascon-80pq version only. By introducing another set of 32 cube structures that break the 60-dimensional cube structure, we can recover the 32-bit key in the first word of the Ascon-80pq initialization with 2^{65} data and time complexity. This process is conducted independently of the above attack on Ascon-128 and Ascon-128a. In other words, after recovering the 32-bit key in the first word, we continue to recover the remaining 128-bit key in the same way as the previous attack on Ascon-128 and Ascon-128a. Consequently, for about $2^{127.97+32} = 2^{159.97}$ (out of 2^{160}) keys, our attack can recover the 160-bit key with 2^{70} data complexity and $2^{72.4}$ time complexity. For remaining keys, our attack can work with 2^{70} data complexity and at worst $2^{104.7}$ time complexity to recover all the 160 bits of the key. Again, our attacks require negligible memory cost.

Complexity comparison with previous attacks. Table 1 gives a comprehensive comparison between our attacks and previous ones on Ascon-AEADs. As we mentioned, the key-recovery attack on Ascon-80pq is the first in the nonce-respecting setting, so the following comparison focuses on the case of Ascon-128/Ascon-128a. Compared to Li *et al.*'s conditional cube attacks [LDW17], our techniques require a lower data complexity (2^{70} versus $2^{77.2}$). The time complexity is $2^{72.4}$ for $2^{127.97}$ keys while theirs is 2^{77} for 2^{117} keys. While in the worst case the time complexity of our attacks ($2^{104.7}$) is slightly larger than theirs ($2^{103.9}$), this occurs only with a small probability of about $(2^{128} - 2^{127.97})/2^{128} \approx 2^{-6}$. Hence, we believe our attacks represent an important improvement of [LDW17].

Compared to Rohit *et al.*'s cube attacks [RHSS21], our techniques require a larger data complexity (2^{70} versus 2^{64}), but with a significantly lower time complexity ($2^{72.4}$ versus 2^{123} for almost all keys). Compared to Rohit and Sarkar's weak-key recovery attacks [RS21], our techniques under the weak-key setting can work for a larger fraction of weak keys and require negligible memory cost, with comparable data/time complexity (considering attacks with a data complexity lower than 2^{64}).

Outline of the paper. In Section 2, we introduce the notations and some necessary background knowledge. Section 3 gives a brief specification of the Ascon-AEADs and discusses some of their useful properties. We describe our new conditional cube attacks on Ascon-128/Ascon-128a in Section 4 and on Ascon-80pq in Section 5. In Section 6, we discuss the assumption that our attacks rely on. Section 7 concludes this paper.

2 Preliminaries

Notations. Let I be a set, we use $|I|$ to represent the size of I. Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field with two elements and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function whose algebraic normal form (ANF) is $f(x) = \sum_{u \in \mathbb{F}_2^n} a_u x^u$, where $x = (x_0, \ldots, x_{n-1})$, $a_u \in \mathbb{F}_2$, and $x^u = \prod_{i=0}^{n-1} x_i{}^{u_i}$. Given a set $I \subseteq \{0, \ldots, n-1\}$ of indexes, $x[I]$ denotes the set of variables $\{x_i : i \in I\}$ and x^I denotes the monomial $\prod_{i \in I} x_i$. $wt(x)$ is the Hamming weight of x, which is the number of 1 in all bits of x. We use "+" to denote all kinds of additions (of integers, field elements, Boolean functions, etc.), the actual meaning of a specific use instance should be clear from the context.

Table 1: Summary of attacks on Ascon-AEAD. The column Var. represents the Ascon variant, including Ascon-128, Ascon-128a and Ascon-80pq. The column Valid N and Valid D describe if the attack violates the nonce-respecting and data limitation ($\leq 2^{64}$) requirements, respectively. The memory complexity is measured with a unit of 128-bit blocks and "0" means the memory cost is negligible. In the Method column, "(H)DL" means (higher-order) differential-linear, "Con." is short for conditional.

Phase	Var.	Type	#R	Complexity D/T/M	#Key	Method	Valid N D	Source
Init.	all	key-rec.	4	$2^{18}/2^{18}/0$	2^{128}	DL	✓ ✓	[DEMS15]
	all	key-rec.	5	$2^{36}/2^{36}/0$	2^{128}	DL	✓ ✓	[DEMS15]
	all	key-rec.	5	$2^{35}/2^{35}/2^{17}$	2^{128}	Cube	✓ ✓	[DEMS15]
	all	key-rec.	5	$2^{26}/2^{26}/0$	2^{128}	Con.DL	✓ ✓	[LLL21]
	all	key-rec.	5	$2^{24}/2^{36}/0$	2^{128}	Con.cube	✓ ✓	[LDW17]
	all	key-rec.	5	$2^{22}/2^{22}/0$	2^{128}	Con.HDL	✓ ✓	[HPTY22]
	all	key-rec.	6	$2^{66}/2^{66}/2^{33}$	2^{128}	Cube	✓ ✓	[DEMS15]
	all	key-rec.	6	$2^{40}/2^{40}/0$	2^{128}	Con.cube	✓ ✓	[LDW17]
	128(a)	key-rec.	7	$2^{77.2}/2^{103.9}/0$	2^{128}	Con.cube	✓ ✗	[LDW17]
	128(a)	key-rec.	7	$2^{77.2}/2^{77}/0$	2^{117}	Con.cube	✓ ✗	[LDW17]
	all	key-rec.	7	$2^{64}/2^{123}/2^{94}$	2^{128}	Cube	✓ ✓	[RHSS21]
	all	key-rec.	7	$2^{64}/2^{97}/2^{63}$	$2^{116.34}$	Cube	✓ ✓	[RS21]
	all	key-rec.	7	$2^{63}/2^{115.2}/2^{62}$	$2^{116.34}$	Cube	✓ ✓	[RS21]
	128(a)	**key-rec.**	**7**	$\mathbf{2^{70}/2^{72.4}/0}$	$\mathbf{2^{127.97}}$	**Con.cube**	✓ ✗	**Sect. 4**
	128(a)	**key-rec.**	**7**	$\mathbf{2^{70}/2^{104.7}/0}$	$\mathbf{2^{128}}$	**Con.cube**	✓ ✗	**Sect. 4**
	128(a)	**key-rec.**	**7**	$\mathbf{2^{64}/2^{120}/0}$	$\mathbf{2^{127.3}}$	**Con.cube**	✓ ✓	**Sect. 4**
	128(a)	**key-rec.**	**7**	$\mathbf{2^{63.32}/2^{115}/0}$	$\mathbf{2^{125}}$	**Con.cube**	✓ ✓	**Sect. 4**
	80pq	**key-rec.**	**7**	$\mathbf{2^{70}/2^{72.4}/0}$	$\mathbf{2^{159.97}}$	**Con.cube**	✓ ✗	**Sect. 5**
	80pq	**key-rec.**	**7**	$\mathbf{2^{70}/2^{104.7}/0}$	$\mathbf{2^{160}}$	**Con.cube**	✓ ✗	**Sect. 5**
Enc.	128a	Forgery	3	$2^{117}/2^{117}/0$	–	Diff.	✓ ✗	[GPT21]
	all	State-rec.	5	$2^{66}/2^{66}/2^{49}$	–	Cube-like	✗✓	[LZWW17]
	all	key-rec.	5	$2^{33}/2^{97}/2^{49}$	2^{128}	Cube-like	✗✗	[LZWW17]
	all	State-rec.	6	$2^{44.8}/2^{128}/?$	–	Con.cube	✗✓	[CHKT23]
	80pq	key-rec.	6	$2^{39.6}/2^{128}/2^{32}$	2^{128}	Cube	✗✓	[CHKT23]
	all	State-rec.	7	$2^{39.6}/2^{39.6}/0$	–	Cube-like	✗✓	[BCP22]
	all	key-rec.	7	$2^{117}/2^{116.2}/2^{32}$	2^{128}	Cube	✗✗	[CHKT23]
Fin.	all	Forgery	3	$2^{34}/2^{34}/0$	–	Diff.	✓ ✓	[DEMS15]
	all	Forgery	3	$2^{32.67}/2^{32.67}/0$	–	Diff.	✓ ✓	[GPT21]
	128a	Forgery	3	$2^{20}/2^{20}/0$	–	Diff.	✓ ✓	[GPT21]
	all	Forgery	4	$2^{102}/2^{102}/9$	–	Diff.	✓ ✓	[DEMS15]
	all	Forgery	4	$2^{96.61}/2^{96.61}/0$	–	Diff.	✓ ✓	[GPT21]
	all	Forgery	4	$2^{9}/2^{9}/0$	–	Cube-like	✗✓	[LZWW17]
	all	Forgery	5	$2^{17}/2^{17}/0$	–	Cube-like	✗✓	[LZWW17]
	all	Forgery	6	$2^{33}/2^{33}/0$	–	Cube-like	✗✓	[LZWW17]

Cube attack and division property. The cube attack was proposed at EUROCRYPT 2009 by Dinur and Shamir to analyze black-box tweakable polynomials [DS09]. Given a keyed Boolean function $f(x,k)$ with n-bit public input $x \in \mathbb{F}_2^n$ and m-bit secret input $k \in \mathbb{F}_2^m$, for a set $I \subseteq \{0,\ldots,n-1\}$ with its complementary set $\bar{I} = \{0,\ldots,n-1\} - I$, we have

$$f(x,k) = x^I \cdot p_I(x[\bar{I}],k) + q(x,k),$$

where each term of $q(x,k)$ misses some variables in $x[I]$. We call x^I the cube term and $p_I(x[\bar{I}],k)$ the superpoly of x^I in $f(x,k)$. If we set the variables in $x[\bar{I}]$ to some fixed constants, the superpoly $p_I(x[\bar{I}],k)$ is a Boolean function of k. Concerning the superpoly, we have the following lemma.

Lemma 1 ([DS09])**.** *For a set $I \subseteq \{0,\ldots,n-1\}$ and a keyed Boolean function*

$$f(x,k) = \sum_{u\in\mathbb{F}_2^n} a_u(k)x^u = x^I \cdot p_I(x[\bar{I}],k) + q(x,k),$$

we have $p_I(x[\bar{I}],k) = \sum_{x[I]\in\mathbb{F}_2^{|I|}} f(x,k)$.

According to Lemma 1, when $x[\bar{I}]$ is fixed to a constant, the value of $p_I(x[\bar{I}],k)$ can be derived by performing $2^{|I|}$ evaluations of f. Usually, a key-recovery attack based on the superpoly is called a cube attack. When the superpoly is zero, the cube sum is always zero, then it can serve as a distinguisher which is called a cube tester [ADMS09].

Algebraic degree. Given a polynomial $f(x,k) = \sum_{u\in\mathbb{F}_2^n} a_u(k)x^u$ where k is secret and x is public. Since k is an unknown constant, $a_u(k) \in \mathbb{F}_2$, we only consider the algebraic degree of f in the public variable, i.e.,

$$\deg(f) = \max_u\{wt(u) \mid a_u(k) \not\equiv 0\}.$$

Division property. A critical cryptanalytic technique in cube attacks is Todo's division property proposed at EUROCRYPT 2015 [Tod15] as a generalization of integral cryptanalysis. Its bit-based variants [TM16] as well as their automatic search methods [XZBL16] have been found to have great potential in probing the structure of a Boolean function described as a sequence of composition of Boolean functions whose overall ANF is too complicated to compute [TIHM17, WHT+18, WHG+19, HLM+20]. In particular, bit-based division property can detect the presence or absence of a monomial in the target Boolean function, and therefore can be used to (partially) determine the algebraic structures of superpolies in cube attacks [TIHM17, WHT+18, WHG+19, HLM+20, HLLT20, HSWW20]. In fact, the division property has become a quite standard tool in assisting cube attacks (as well as integral attacks). In this work, we take the MILP (Mixed Integer Linear Programming) based approach [XZBL16] to search for division properties and calculate the degree upper bounds.

3 Specification and Useful Properties of Ascon

In a high level, the Ascon AEAD algorithm takes as input a nonce N, a secret key K, an associated data A and a plaintext or message M, and produces a ciphertext C and a tag T. The authenticity of the associated data and message can be verified against the tag T. Ascon adopts a MonkeyDuplex [Dae12] mode with a stronger keyed initialization and keyed finalization phases as illustrated in Figure 1.[1] The underlying permutations p^a and p^b are iterative designs, whose round function p is based on the substitution permutation network (SPN) design paradigm and consists of three simple steps p_C, p_S, and p_L. The round function $p = p_L \circ p_S \circ p_C$ operates on a 320-bit state arranged into five 64-bit words. The steps p_C, p_S, and p_L are visualized in Figure 2 and described as follows.

Addition of constants (p_C). An 8-bit constant is XORed to the bit positions $56,\ldots,63$ of the 64-bit word x_2 at each round.

[1]Thanks to TikZ for Cryptographers [Jea16]. All figures in this paper are created by TiKZ.

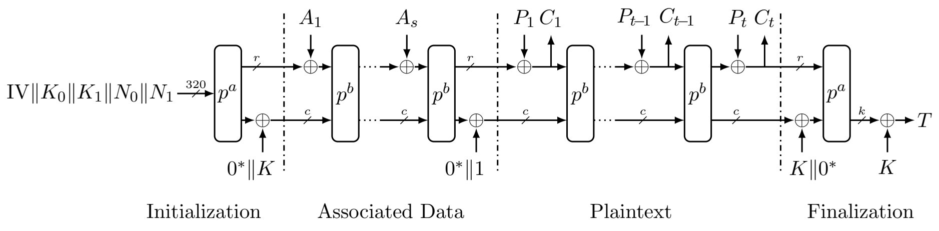

Figure 1: The high-level structure of the Ascon-AEAD.

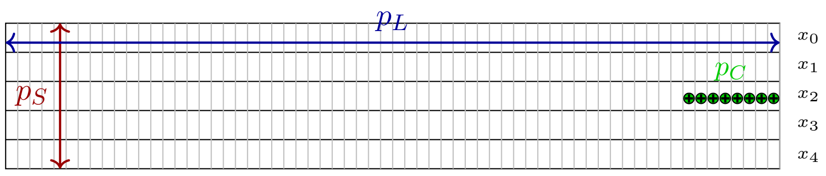

Figure 2: The demonstration of p_C, p_S and p_L.

Substitution layer (p_S). Update each slice of the 320-bit state by applying the 5-bit Sbox defined by the algebraic normal forms in Figure 3.

$$\begin{cases} y_0 = x_4x_1 + x_3 + x_2x_1 + x_2 + x_1x_0 + x_1 + x_0 \\ y_1 = x_4 + x_3x_2 + x_3x_1 + x_3 + x_2x_1 + x_2 + x_1 + x_0 \\ y_2 = x_4x_3 + x_4 + x_2 + x_1 + 1 \\ y_3 = x_4x_0 + x_4 + x_3x_0 + x_3 + x_2 + x_1 + x_0 \\ y_4 = x_4x_1 + x_4 + x_3 + x_1x_0 + x_1 \end{cases} \qquad \begin{cases} y_0 \leftarrow \Sigma_0(x_0) = x_0 + (x_0 \ggg 19) + (x_0 \ggg 28) \\ y_1 \leftarrow \Sigma_1(x_1) = x_1 + (x_1 \ggg 61) + (x_1 \ggg 39) \\ y_2 \leftarrow \Sigma_2(x_2) = x_2 + (x_2 \ggg 1) + (x_2 \ggg 6) \\ y_3 \leftarrow \Sigma_3(x_3) = x_3 + (x_3 \ggg 10) + (x_3 \ggg 17) \\ y_4 \leftarrow \Sigma_4(x_4) = x_4 + (x_4 \ggg 7) + (x_4 \ggg 41) \end{cases}$$

Figure 3: ANF of the Sbox (left) and the linear layer (right) of Ascon.

Linear diffusion layer (p_L). Apply a linear transformation Σ_i to each 64-bit word x_i with $0 \le i < 5$, where Σ_i is defined in Figure 3.

Notations for describing the Ascon permutation states. The 320-bit output state of the Ascon permutation after r rounds is denoted by $S^{(r)} = (p_L \circ p_S \circ p_C)^r(S^{(0)})$, where $S^{(0)}$ is the input into the permutation. We also use $S^{(r.5)}$ to represent a half-round $p_S \circ p_C(S^{(r)})$. Every state consists of 5 words as

$$S^{(r)} = S_0^{(r)} \| S_1^{(r)} \| S_2^{(r)} \| S_3^{(r)} \| S_4^{(r)},$$

where $S_i^{(r)}$ is the i^{th} word (the i^{th} row) of $S^{(r)}$; the j^{th} bit of $S_i^{(r)}$ is denoted by $S_{i,j}^{(r)}$ where $0 \le i < 5, 0 \le j < 64$, and $S^{(r)}_{i,\{j_0,j_1,\ldots,j_{s-1}\}}$ represents the s bits $S^{(r)}_{i,j_0}, S^{(r)}_{i,j_1}, \ldots, S^{(r)}_{i,j_{s-1}}$. $S^{(r)}_{0,0}$ is the leftmost bit of the first row of the state matrix $S^{(r)}$.

Ascon-128, Ascon-128a, and Ascon-80pq. The Ascon AEAD family consists of three members, Ascon-128, Ascon-128a, and Ascon-80pq where Ascon-128 is the primary recommendation. Both Ascon-128 and Ascon-128a use a 128-bit key while Ascon-80pq takes a 160-bit one. For all the three variants, $S_3^{(0)}||S_4^{(0)}$ is loaded with the 128-bit nonce. For Ascon-128 and Ascon-128a, $S_1^{(0)}||S_2^{(0)}$ is loaded with the 128-bit key and $S_0^{(0)}$ is loaded with the 64-bit initial value (IV). Ascon-80pq takes a half of the IV positions to allow 32 more key bits. Thus, $S^{(0)}_{0,\{32,33,\ldots,63\}}||S_1^{(0)}||S_2^{(0)}$ is initialized as a 160-bit key, and $S^{(0)}_{0,\{0,1,\ldots,31\}}$ is the 32-bit IV. The rates of the Ascon-128 and Ascon-80pq are 64 bits

Table 2: Ascon-AEAD variants and their recommended parameters

Name	State size	Rate r	Size of			Rounds		IV
			Key	Nonce	Tag	p^a	p^b	
Ascon-128	320	64	128	128	128	12	6	80400c0600000000
Ascon-128a	320	128	128	128	128	12	8	80800c0800000000
Ascon-80pq	320	64	160	128	128	12	6	a0400c06

while 128 bits for Ascon-128a. The parameters are summarized in Table 2 and all three variants provide 128-bit security.

The adversary can only access the rate part for Ascon-AEADs. Our paper considers only the case where the first 64 bits of output are accessed and thus our attack works for all three versions. Since the linear layer is applied to each row, we do not consider the linear layer of the last round. In other words, for 7-round Ascon-AEAD, the output state we consider is actually $S^{(6.5)}$ which is totally equivalent to $S^{(7)}$ in this paper.

Degree matrix of the Ascon permutation. In [HPTY22], Hu *et al.* introduced the degree matrix to describe and trace the changes of algebraic degrees of the Ascon permutation states.

Definition 1 (Degree Matrix of $S^{(r)}$ [HPTY22])**.** The algebraic degrees or their upper bounds of the bits in the state $S^{(r)}$ are called a degree matrix of $S^{(r)}$, denoted by

$$\mathrm{DM}(S^{(r)}) = \Big(\deg(S^{(r)}[i][j]), 0 \le i < 5, 0 \le j < 64\Big).$$

Considering the ANF of the Sbox and diffusion layer of the Ascon permutation, given the degree matrix of $S^{(r)}$, we can quickly calculate the degree matrix of $S^{(r+1)}$ according to the following Lemmas 2 and 3.

Lemma 2 (Degree Matrix Transition over p_S [HPTY22])**.** *With the knowledge of* $\mathrm{DM}(S) = (d_{i,j}, 0 \le i < 5, 0 \le j < 64)$*, we have* $\mathrm{DM}(p_S(S)) = (d'_{i,j}, 0 \le i < 5, 0 \le j < 64)$*, where* $d'_{i,j}, 0 \le i < 5, 0 \le j < 64$ *are computed as*

$$\left.\begin{aligned}
d'_{0,j} &= \max(d_{4,j}+d_{1,j}, d_{3,j}, d_{2,j}+d_{1,j}, d_{2,j}, d_{0,j}+d_{1,j}, d_{1,j}, d_{0,j})\\
d'_{1,j} &= \max(d_{4,j}, d_{3,j}+d_{2,j}, d_{3,j}+d_{1,j}, d_{3,j}, d_{2,j}+d_{1,j}, d_{2,j}, d_{1,j}, d_{0,j})\\
d'_{2,j} &= \max(d_{4,j}+d_{3,j}, d_{4,j}, d_{2,j}, d_{1,j}, 0)\\
d'_{3,j} &= \max(d_{4,j}+d_{0,j}, d_{4,j}, d_{3,j}+d_{0,j}, d_{3,j}, d_{2,j}, d_{1,j}, d_{0,j})\\
d'_{4,j} &= \max(d_{4,j}+d_{1,j}, d_{4,j}, d_{3,j}, d_{1,j}+d_{0,j}, d_{1,j})
\end{aligned}\right., \quad 0 \le j < 64$$

Lemma 3 (Degree Matrix Transition over p_L [HPTY22])**.** *With the knowledge of* $\mathrm{DM}(S) = (d'_{i,j}, 0 \le i < 5, 0 \le j < 64)$*, we have* $\mathrm{DM}(p_L(S)) = (d''_{i,j}, 0 \le i < 5, 0 \le j < 64)$*, where* $d''_{i,j}, 0 \le i < 5, 0 \le j < 64$ *are computed as*

$$\begin{aligned}
d''_{0,j} &= \max(d'_{0,j+0}, d'_{0,j-19 \bmod 64}, d'_{0,j-28 \bmod 64})\\
d''_{1,j} &= \max(d'_{1,j+0}, d'_{1,j-61 \bmod 64}, d'_{1,j-39 \bmod 64})\\
d''_{2,j} &= \max(d'_{2,j+0}, d'_{2,j-1 \bmod 64}, d'_{2,j-6 \bmod 64}), \quad 0 \le j < 64\\
d''_{3,j} &= \max(d'_{3,j+0}, d'_{3,j-10 \bmod 64}, d'_{3,j-17 \bmod 64})\\
d''_{4,j} &= \max(d'_{4,j+0}, d'_{4,j-7 \bmod 64}, d'_{4,j-41 \bmod 64})
\end{aligned}$$

4 New Conditional Cube Attack on Ascon-128(a)

In [RHSS21], new distinguishers for Ascon initialization were proposed by Rohit *et al.* The inputs to these distinguishers have a special initial structure (for convenience, we

Table 3: Algebraic degrees or their upper bounds of Ascon in cube variables with the structure IS_{-1} derived from the division properties [RHSS21]. Note in [RHSS21], the degrees are given for $S^{(r)}$ $(2 \le r \le 7)$ while here we focus on $S^{(r.5)}$ $(1 \le r \le 6)$.

Round r	**Degrees of words**				
	$S_0^{(r)}$	$S_1^{(r)}$	$S_2^{(r)}$	$S_3^{(r)}$	$S_4^{(r)}$
0.5	1	1	0	0	1
1.5	2	1	1	2	2
2.5	3	3	4	4	3
3.5	7	8	7	7	6
4.5	15	15	13	14	15
5.5	30	29	29	30	30
6.5	59	59	60	60	59

will use IS_{-1} as its shorthand): the first and second 64-bit nonce are always equal and regarded as 64-bit cube variables, i.e.,

$$IS_{-1}: \ (S_3^{(0)}[i], S_4^{(0)}[i]) = (v_i, v_i), \ 0 \le i < 64. \tag{1}$$

With IS_{-1}, the upper bounds on the algebraic degrees of $S^{(r)}$ are calculated by the division properties [HLM+20, Tod15]. These upper bounds are given in Table 3. Therefore, the 7-round Ascon initialization can be distinguished with 2^{60} nonces, which is the best distinguisher for 7-round Ascon-AEAD so far. As we mentioned in the introduction, this distinguisher, unfortunately, is difficult to be utilized in a key-recovery attack as it does not satisfy the borderline cube properties [DEMS15]. To use it in a key-recovery attack, new techniques are necessary.

We notice that the degrees in Table 3 heavily rely on the input structure of Equation 1, i.e., IS_{-1}. Therefore, we take a *break-fix* strategy to transpose this distinguisher into a key-recovery attack on 7-round Ascon.

4.1 Phase 1: Break

We introduce 64 new initial structures $IS_j, 0 \le j < 64$ for the nonce, each of which is slightly adapted from Equation 1,

$$IS_j : \left(S_3^{(0)}[i], S_4^{(0)}[i]\right) = \begin{cases} (v_i + 1, v_i) & \text{if } i = j \\ (v_i, v_i) & \text{if } i \neq j \end{cases}, \quad 0 \le i < 64 \tag{2}$$

Note that for a specific IS_j where $0 \le j < 64$, the two bits of the nonce $S_3^{(0)}[j]$ and $S_4^{(0)}[j]$ are set to different forms, which breaks IS_{-1}. Such a break will have an effect on the degrees of the Ascon initializations. With a similar modeling strategy as [RHSS21], we use the MILP-based division properties [XZBL16] to calculate the degrees of the states after $r.5$ $(0 \le r \le 6)$ rounds. The results are provided in Table 4.

Comparing Table 3 and Table 4, we can find that the degrees resulted from IS_j $(0 \le j < 64)$ are in general larger than those from IS_{-1}. The differences between the two tables are actually all due to the differences of the first two rounds (this will be clearly stated by Proposition 2 later). We analyze the reason for the difference in the first and second rounds. According to the ANFs of the Ascon Sbox, the IS_{-1} that requires $S_{3,i}^{(0)} = S_{4,i}^{(0)}$ will make $S_{2,i}^{(1)}$ and $S_{3,i}^{(1)}$ be independent of any nonce bits, for $0 \le i < 64$. In other words, the algebraic degrees of the 128 bits of $S_2^{(1)}$ and $S_3^{(1)}$ are all zero. Note that all quadratic terms of the second ANF of the Sbox (y_1 in the left part of Figure 3) are related to x_2 and x_3, thus, for the IS_{-1}, the degrees of all 64 bits of $S_1^{(1.5)}$ are still 1.

Table 4: Algebraic degrees of Ascon in cube variables with the structure IS_j $(0 \leq j < 64)$ derived from the division properties. "x/y" means that the degrees of some bits of that word are x and others are y.

Round r	**Degrees of words**				
	$S_0^{(r)}$	$S_1^{(r)}$	$S_2^{(r)}$	$S_3^{(r)}$	$S_4^{(r)}$
0.5	1	1	1/0	0	1
1.5	2	2/1	1	2	2
2.5	4/3	4/3	4	4	4/3
3.5	8/7	8	8/7	8/7	7/6
4.5	16/15	16/15	14	15/14	16/15
5.5	31	30	30	31	31
6.5	60	60	61	61	60

However, for a specific $IS_j, (0 \leq j < 64)$, we have $S_{3,j}^{(0)} = S_{4,j}^{(0)} + 1$, thus there will be one nonce bit remaining in $S_{2,j}^{(0.5)}$ making the degree of $S_{2,j}^{(0.5)}$ be 1. After the diffusion layer, the 1-degree bit spreads to 3 positions. Therefore, the p_S of the second round will multiply these 3 1-degree bits with the existing ones, leading to three 2-degree bits for $S_1^{(1.5)}$. An illustration for the case of IS_0 is given in Figure 4.

As a result, the degree upper bounds after 7 rounds are 59 for IS_{-1} and 60 for IS_j $(0 \leq j < 64)$, so we can use the gap to mount a conditional cube attack.

Remark. There are more kinds of methods to break IS_{-1}, e.g.,

$$IS'_j : \left(S_3^{(0)}[i], S_4^{(0)}[i]\right) = \begin{cases} (x, y) & \text{if } i = j \\ (v_i, v_i) & \text{if } i \neq j \end{cases}, \quad 0 \leq i < 64$$

where $(x, y) \in \{(v_i, 0), (0, v_i), (v_i, v_i + 1)\}$. We tried all of them and found Equation 2 is the one that requires the least key conditions for the attack.

4.2 Phase 2: Fix

In this phase, our task is to identify key conditions that can reset the degrees of $S^{(6.5)}$ back to 59. The differences between the two tables first appear in the first round (see the top part of Figure 4). For IS_j $(0 \leq j < 64)$, the degree of $S_{2,0}^{(0.5)}$ is 1. Denote the 128-bit key loaded into $S_1^{(0)} || S_2^{(0)}$ by $k_0, k_1, \ldots, k_{127}$, the ANF of $S_{2,j}^{(0.5)}$ is

$$S_{2,j}^{(0.5)} = v_j + k_j + k_{j+64} + 1,$$

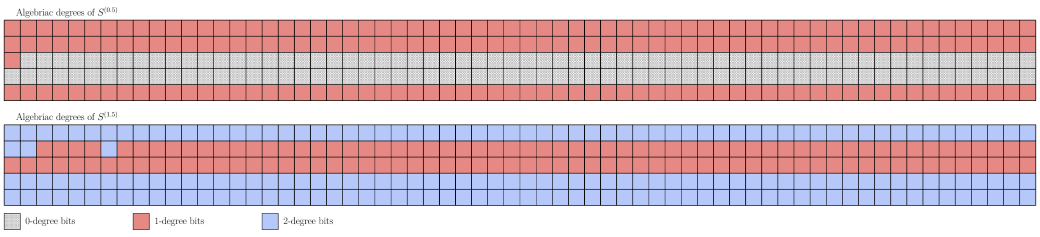
Algebriac degrees of $S^{(0.5)}$
Algebriac degrees of $S^{(1.5)}$
0-degree bits
1-degree bits
2-degree bits

Figure 4: Algebraic degrees of bits in $S^{(0.5)}$ and $S^{(1.5)}$ resulted from IS_0. For the other IS_j $(1 \leq j < 64)$, similar results can be obtained by rotating the bits.

which means the degree of $S_{2,j}^{(0.5)}$ is impossible to be controlled by any key conditions. We have to handle the second round. At the second round the algebraic degrees of $S_1^{(1.5)}$ can be 2 for IS_j $(0 \le j < 64)$ whereas they are always 1 for IS_{-1}, so if we can reset the algebraic degrees of $S_2^{(1.5)}$ to 1, the degrees of $S_0^{(6.5)}$ will return to 59. This is guaranteed by the following proposition.

Proposition 1. *If the algebraic degrees of the five words of $S^{(1.5)}$ are $(2, 1, 1, 2, 2)$, the upper bounds on the algebraic degrees of the five words of $S^{(6.5)}$ are $(59, 59, 60, 60, 59)$.*

Proof. Note that the p_C operation does not influence the degree matrix of a state in this case. The proposition can be proved by a direct application of Lemmas 2 and 3. □

We take IS_0 as an example. The bottom part of Figure 4 gives the concrete algebraic degrees of $S^{(1.5)}$ bits resulted from IS_0. It can be seen that there are three 2-degree bits in $S_1^{(1.5)}$. The ANFs of these three 2-degree bits when $j = 0$ are

$$\begin{aligned} S_{1,0}^{(1.5)} &= v_0v_3(k_3 + k_{67}) + v_0v_{25}(k_{25} + k_{89}) + \mathcal{L}_0(v_0, v_1, \ldots, v_{63}) \\ S_{1,1}^{(1.5)} &= v_0v_1(k_1 + k_{65}) + v_0v_4(k_4 + k_{68}) + v_0v_{26}(k_{26} + k_{90}) + \mathcal{L}_1(v_0, v_1, \ldots, v_{63}) \\ S_{1,6}^{(1.5)} &= v_0v_6(k_6 + k_{70}) + v_0v_9(k_9 + k_{73}) + v_0v_{31}(k_{31} + k_{95}) + \mathcal{L}_6(v_0, v_1, \ldots, v_{63}) \end{aligned} \tag{3}$$

where $\mathcal{L}_i(\cdot)$ for $i = 0, 1, 6$ are three linear functions. To reset the algebraic degrees of the three 2-degree bits to 1, we need to eliminate all the quadratic terms in $S_{1,0}^{(1.5)}$, $S_{1,1}^{(1.5)}$ and $S_{1,6}^{(1.5)}$. For other $j \in \{0, \ldots, 64\}$, the situation are similar. We list all their quadratic terms in Table 5. Note that due to the p_C in the first round, 4 key coefficients have constant 1 terms (the constant for the first round is `0xf0`, so there are 4 key coefficients that are affected), they have no influence on our analysis (actually, we can even completely ignore the p_C in the first round with an equivalent key technique).

According to Tables 3 and 4, if all these quadratic terms are canceled, the degrees of the 7-round Ascon are 59; otherwise, the degrees are 60. Therefore, the dimension of the cube we choose in our attack should be 60. In other words, 4 cube variables will be set as constants. Concretely, if we set any four out of the 8 cube variables (the indices are modulo 64)

$$v_{j+3}, v_{j+25}, v_{j+1}, v_{j+4}, v_{j+26}, v_{j+6}, v_{j+9}, v_{j+31}$$

to be zero and construct a 60-dimensional cube, we can use this cube to test if the corresponding four key coefficients are *all* zero. Taking the above $j = 0$ as an example again, if we set $v_3 = v_{25} = v_1 = v_4 = 0$, we derive a 60-dimensional cube with the remaining cube variables v_i $(i \in \{0, \ldots, 63\} \backslash \{3, 25, 1, 4\})$. Then if the cube tester on the 7-round Ascon is zero, we determine that $k_{26} + k_{90} = k_6 + k_{70} = k_9 + k_{73} = k_{31} + k_{95} = 0$.

For the sake of a clear description, we introduce the definitions of *key set* and *good/bad key set*.

Definition 2 (Key Set and Good/Bad Key Set)**.** Given a 128-bit key, we call the 8 key coefficients in Table 5 derived from a specific IS_j $(0 \le j < 64)$ the j-th key set (KS). A KS that contains at least four zero key coefficients is called a good key set (GKS), otherwise a bad key set (BKS).

When a KS is good, we can detect the four zero key coefficients with at most testing $\binom{8}{4} = 70$ different 60-dimensional cube testers as described above. Once we detect a GKS and have known the four zero key coefficients, it is easy to use the four known key coefficients to recover the remaining four key coefficients. Taking Equation 3 as an instance, if $k_3 + k_{67}$, $k_{25} + k_{89}$, $k_1 + k_{65}$ and $k_4 + k_{68}$ are zero and we have detected them with a cube that satisfies $v_{26} = v_6 = v_9 = v_{31} = 0$, we can observe the cube sum of the cube tester

Table 5: Quadratic terms in the ANFs of bits in $S_1^{(1.5)}$ resulted from IS_j $(0 \le j < 64)$. Note that some key coefficients have a constant 1 term which is caused by the p_C of the first round, but they do not influence our analysis.

Settings	Quad. bits	Quad. terms	Key Coef.
IS_j	$S_{1,j}^{(1.5)}$	$v_j v_{j+3}$	$k_{j+3} + k_{j+67}$ ♣
		$v_j v_{j+25}$	$k_{j+25} + k_{j+89}$ $\flat$
	$S_{1,j+1}^{(1.5)}$	$v_j v_{j+1}$	$k_{j+1} + k_{j+65}$ $\diamondsuit$
		$v_j v_{j+4}$	$k_{j+4} + k_{j+68}$ $\angle$
		$v_j v_{j+26}$	$k_{j+26} + k_{j+90}$ $\ddagger$
	$S_{1,j+6}^{(1.5)}$	$v_j v_{j+6}$	$k_{j+6} + k_{j+70}$ $\sharp$
		$v_j v_{j+9}$	$k_{j+9} + k_{j+73}$ $\natural$
		$v_j v_{j+31}$	$k_{j+31} + k_{j+95}$ $\dagger$

† when $j \in \{25, 26, 27, 28\}$, the key coefficient is $k_{j+31} + k_{j+95} + 1$
‡ when $j \in \{30, 31, 32, 33\}$, the key coefficient is $k_{j+26} + k_{j+90} + 1$
$^\flat$ when $j \in \{31, 32, 33, 34\}$, the key coefficient is $k_{j+25} + k_{j+89} + 1$
$^\natural$ when $j \in \{47, 48, 49, 50\}$, the key coefficient is $k_{j+9} + k_{j+73} + 1$
$^\sharp$ when $j \in \{50, 51, 52, 53\}$, the key coefficient is $k_{j+6} + k_{j+70} + 1$
$^\angle$ when $j \in \{52, 53, 54, 55\}$, the key coefficient is $k_{j+4} + k_{j+68} + 1$
$^♣$ when $j \in \{53, 54, 55, 56\}$, the key coefficient is $k_{j+3} + k_{j+67} + 1$
$^\diamondsuit$ when $j \in \{55, 56, 57, 58\}$, the key coefficient is $k_{j+1} + k_{j+65} + 1$

that satisfies $v_1 = v_6 = v_9 = v_{31} = 0$ to determine the value of $k_{26} + k_{90}$. Actually, when we set $v_1 = v_6 = v_9 = v_{31} = 0$, the key coefficients of the quadratic terms in Equation 3 are $k_3 + k_{67}$, $k_{25} + k_{89}$, $k_4 + k_{68}$ and $k_{26} + k_{90}$. Since we have known the first three key coefficients are zero, the cube sum of this cube tester will tell us the value of $k_{26} + k_{90}$ directly. Note that all these operations are already done when we check the 70 different 60-dimensional cube testers, so no additional complexities are required.

To conclude, if a KS is good, we can do (at most) 70 60-dimensional cube testers to recover all its 8 key coefficients. In our attack, we will try 70 cube testers for each of the 64 KS, to recover the keys in those good ones. Thus, in total we need to conduct $64 \times 70 = 4,480$ different cube testers.

The whole attack process is summarized in Algorithm 1. In this algorithm, we first compute all $64 \times \binom{8}{4} = 4,480$ cubes and derive their cube sums. If the cube sum is zero, then we know the corresponding 4 key coefficients are all zero (Line 9). Otherwise, $(0, 0, 0, 0)$ is not considered as their candidates. What's more, if we have known four key coefficients are not all zero but three of them have been known as zero, then we determine the remaining one as 1 (Line 15). In the exhaustive search phase, for those undetermined key coefficient tuples, we already know they cannot be all zero. Since for each $IS_j, 0 \le j < 64$, the data complexity is at most 2^{64}, the whole data complexity is $2^{64} \times 64 = 2^{70}$ chosen nonces. The time complexity is influenced by the time complexity of the exhaustive search phase (denoted by $\mathcal{T}_e$) as

$$\mathcal{T} = 64 \times 70 \times 2^{60} + \mathcal{T}_e \approx 2^{72.1} + \mathcal{T}_e.$$

The memory complexity is negligible.

Note that this attack works under an assumption that before the fixing process the algebraic degree of the 7-round Ascon is really 60 and the cube sum is not highly biased. The assumption is concluded as Assumption 1.

Assumption 1. *The cube sum of a 60-dimensional cube that satisfies* IS_j $(0 \le j < 64)$ *is not zero when the corresponding four coefficients are not all zero.*

Algorithm 1 Conditional cube attack on Ascon-128/Ascon-128a

```
1: procedure ConCubeAttack-128(IV)
2:     Initialize two sets $\mathbb{K}$ and $\mathbb{N}$
3:     for $0 \le j < 64$ do
4:         Construct cube with the structure $IS_j$ (Equation 2)
5:         for 4 variables from $\{v_{j+3}, v_{j+25}, v_{j+1}, v_{j+4}, v_{j+26}, v_{j+6}, v_{j+9}, v_{j+31}\}$ do ▷ $\binom{8}{4}$ choices
6:             Derive the 60-dimensional cube by setting the 4 selected variables as 0
7:             Compute the cube sum of this cube
8:             if the cube sum is 0 then
9:                 Add the corresponding 4 key coefficients into $\mathbb{K}$
10:            else
11:                Add the corresponding 4 key coefficients into $\mathbb{N}$
12:            end if
13:        end for
14:    end for
15:    Check every 4-tuple in $\mathbb{N}$ to see if there are 3 elements that have appeared in a certain
       4-tuple in $\mathbb{K}$. If so, determine the value of the last key coefficient as 1 and remove the 4 tuple
       from $\mathbb{N}$
16:    Brute-force search the unknown key information, for those 4 tuples in $\mathbb{N}$, $(0, 0, 0, 0)$ should
       not be considered                                                    ▷ exhaustive search
17:    return 128-bit key
18: end procedure
```

A detailed discussion on this assumption will be given in Section 6.

4.3 Time Complexity Analysis

Obviously, we want the complexity of the exhaustive search to be significantly smaller than $2^{72.1}$. Equivalently, we need to recover more than $128 - 72.1 = 55.9$ bits of key information from Algorithm 1. For the sake of simplification, in the following we study the probability of such an event that we can recover 58 bits of key information from Algorithm 1 (in this case, the whole time complexity is $\mathcal{T} = 2^{72.1} + 2^{128-58} \approx 2^{72.4}$).

To this end, we need to study: (1) The distribution of the GKS as the number of GKS directly influence the key information we will get. (2) Notice that some KS may contain linearly-dependent key coefficients. For example, both two KS from IS_0 and IS_2 contain the same key coefficient $k_3 + k_{67}$. Thus, we also need to estimate how many GKS we need to accumulate sufficient *independent* key coefficients.

The distribution of the number of GKS. Let x_i for $0 \le i < 64$ be a random variable satisfying

$$x_i = \begin{cases} 1 & \text{the } i\text{-th KS is good} \\ 0 & \text{the } i\text{-th KS is bad} \end{cases}$$

Define $X = \sum_{i=0}^{63} x_i$ be a random variable representing the number of GKS. Our task is to find a proper distribution of X.

From the plot in Figure 5, we observe that the points form a curve that is very similar to the normal distribution but with a small skewness. Thus, we use a skew-normal distribution to describe X, denoted by $\mathcal{SN}(\xi, \eta, \lambda)$.

The skew-normal distribution is a relatively new distribution. In [Azz85], Azzalini introduces the skew-normal class as one being able to reflect varying degrees of skewness, which is mathematically tractable and which includes the normal distribution as a special case. A random variable Z follows a standard skew-normal distribution with the parameter

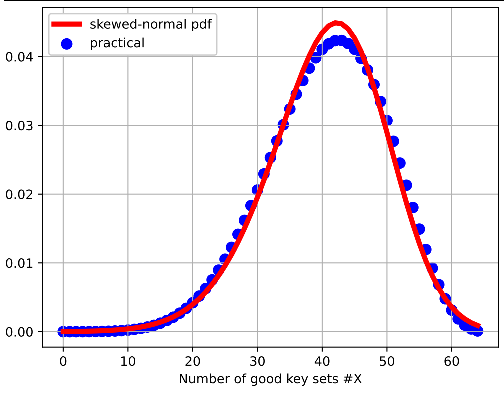

Figure 5: The skew-normal distribution for the number of GKS. The blue dots are the experimental values, whereas the red line is the skewed-normal distribution $\mathcal{SN}(48.88204620, 12.19456124, -1.52220805)$.

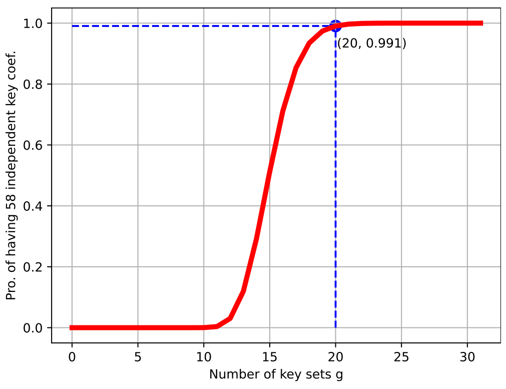

Figure 6: The probability for the number of KS we need for 58 independent key coefficients. When $g = 20$, the probability has been 0.991.

λ (called the shape parameter) is denoted by $Z \sim \mathcal{SN}(\lambda)$. The density is of the form,

$$\varphi(x;\lambda) = 2\phi(x)\Phi(\lambda x), \quad -\infty < x < \infty, -\infty \leq \lambda < \infty. \tag{4}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal density and distribution functions, respectively. The case $\lambda = 0$ corresponds to the standard normal distribution. The standard skew-normal contribution can be generalized by the inclusion of location and scale parameters denoted by ξ and η, respectively. Thus, if $Z \sim \mathcal{SN}(\lambda)$, $Y = \xi + \eta Z$ is a random variable that follows $\mathcal{SN}(\xi, \eta, \lambda)$.

Next, we estimate the parameters ξ, η, λ for the skew-normal distribution according to our case. In the one hand, according to [Azz85], for a random variable $Y \sim \mathcal{SN}(\xi, \eta, \lambda)$, the first three moments of Y are:

$$E(Y) = \xi + b\eta\delta \tag{5}$$
$$E(Y^2) = \xi^2 + 2b\xi\eta\delta + \eta^2 \tag{6}$$
$$E(Y^3) = \xi^3 + 3b\xi^2\eta\delta + 3\xi\eta^2 + 3b\eta^3\delta - b\eta^3 \tag{7}$$
$$D(Y) = \eta^2(1 - b^2\delta^2) \tag{8}$$

where $b = \sqrt{\frac{2}{\pi}}$ and $\delta = \frac{\lambda}{\sqrt{1+\lambda^2}}, -1 \leq \delta < 1$. The coefficient of skewness for Y is the same as that for Z, namely,

$$\gamma_1 = \frac{b\delta^3(2b^2 - 1)}{\sqrt[3]{(1 - b^2\delta^2)^2}}, \quad -0.99527 < \gamma_1 < 0.99527.$$

On the other hand, since $X = \sum_{i=0}^{63} x_i$ and the expectation of x_i is

$$E(x_i) = \frac{\sum_{j \geq 4}\binom{8}{j}}{256} = \frac{163}{256},$$

thus, the mean of X is

$$E(X) = \sum_{i=0}^{63} E(x_i) = 64 \times \frac{163}{256} = 40.75. \tag{9}$$

The second moment of X, i.e., $E(X^2)$ can be computed as follows,

$$E(X^2) = E\left(\sum_{i=0}^{63} x_i\right)^2 = \sum_{0\le i,j<64} E(x_i x_j).$$

Due to the rotational property of the Ascon state, for a fixed $0 \le d < 64$, $E(x_i x_{i+d})$ are all the same for any $0 \le i < 64$. Thus,

$$E(X^2) = 64 \times \sum_{0\le j<64} E(x_0 x_j) \approx 0.63671875. \tag{10}$$

$E(X^3)$ can be computed in a similar way,

$$E(X^3) = E\left(\sum_{i=0}^{63}(x_i)\right)^3 = 64 \times \sum_{0\le j,k<64} E(x_0 x_j x_k) \approx 77532.16384887. \tag{11}$$

The variance of X is then

$$D(X) = E(X^2) - (EX)^2 \approx 82.57714843. \tag{12}$$

The skewness coefficient is usually computed by[2]

$$\gamma_X = \frac{E(X - E(X))^2}{\sqrt[2]{D(X)^3}} \approx -0.30759064.$$

By equating the above corresponding statistical properties of $\mathcal{SN}(\xi,\delta,\lambda)$ and X (i.e., Equations 5 and 9, Equations 6 and 10, Equations 7 and 11, Equations 8 and 12), we can derive the values of the parameters of $\mathcal{SN}$,

$$\xi \approx 48.88204620,\ \eta \approx 12.19456124,\ \lambda \approx -1.52220805.$$

Thus, we derive the corresponding $\mathcal{SN}(\xi,\eta,\lambda)$. The density curve of this distribution is shown in Figure 5, it can be seen it fits the scatted points well.

The number of GKS for 58 independent key coefficients. We denote by P_g the probability that g KS can generate at least 58 independent key coefficients. For each g, we randomly choose 2^{20} g KS and calculate the number of independent key coefficients. The curve for P_g is shown in Figure 6, according to which we find $P_{20} \approx 0.991$. Thus, if a key has 20 GKS, it can accumulate at least 58 independent key coefficients with a probability about 0.991.

We use the skew-normal distribution $\mathcal{SN}(\xi,\eta,\lambda)$ specified above to predict the key space with equal or more than 20 GKS. With the help of the `scipy.stats.skewnorm` class in the `scipy` library for `python`[3], we have

$$\Pr[X \ge 20] = 1 - \Pr[X \le 19] \approx 0.986.$$

We use 2^{20} randomly-generated keys to practically test $\Pr[X \ge 20]$, and the experimental probability is about 0.986 which is extremely close to our theoretical value.

Consequently, with a probability about $\Pr[X \ge 20] \cdot P_{20} \approx 0.977$, we can accumulate at least 58 independent key coefficients. Thus, the exhaustive search would cost at most $2^{128-58} = 2^{70}$ time complexity. Together with the time complexity of Algorithm 1, the whole time complexity is about $2^{72.4}$. In other words, we can recover the 128-bit key with a time complexity of $2^{72.4}$ Ascon initializations for about $2^{128} \times 0.977 \approx 2^{127.97}$ keys.

[2]http://brownmath.com/stat/shape.htm

[3]https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.skewnorm.html

The worst case. When we cannot obtain 58 bits of key information from Algorithm 1, the time complexity will increase. For sake of simplification, we directly calculate the time complexity for the worst case, i.e., there is no GKS in the secret key. When a KS is not good, then we know any four key coefficients cannot be zero simultaneously. Thus, $\sum_{i\geq 4}\binom{8}{i} = 163$ out of all 256 possibilities can be excluded. We only need to search for the remaining key values by brute force. With testing 2^{32} randomly-chosen 128-bit keys, we detect 420 keys that lead to such worst case. Thus, the percentage of such keys is about $2^{-23.3}$. In other words, if none of the KS are good, we know that only $2^{128-23.3} = 2^{104.7}$ keys that are possible to be candidates. Therefore, the time complexity of the worst case is about $2^{104.7}$.

4.4 Weak-Key Attacks Satisfying the Data Limitation

The designers of Ascon have established a security claim as follows,

> "*The number of processed plaintext and associated data blocks protected by the encryption algorithm is limited to a total of 2^{64} blocks per key ...*" ([DEMS21], Chapter 2, Page 9])

Consequently, the maximal number of nonces we can use should be limited within 2^{64}. While at the present stage we should encourage attacks even they do not comply with this restriction to gain a better understanding of Ascon's security features, it is also valuable to explore whether an attack can be adapted to meet this requirement whenever possible. We provide two kinds of attacks on Ascon both of which satisfy the data limit.

Weak-key attack 1. Let us focus on only one certain IS_j. If four coefficients of its key set are zero, we can recover values of all these 8 key coefficients. For one IS_j, the data complexity is 2^{64}, so this satisfies the data limit. The time complexity of this attack is dominated by the exhaustive search process which is 2^{120} Ascon initializations. The above event happens with a probability $\sum_{i\geq 4}\binom{8}{4}/256 \approx 0.63$, thus the size of the weak-key space is about $2^{128} \times 0.63 \approx 2^{127.3}$.

Weak-key attack 2. When we have known that three key coefficients in a key set are zero, we can combine the three with another one in the same key set to obtain the value of the latter by conducting a corresponding cube tester.

When $j = 0$, the 8 key coefficients in the key set are

$$k_3 + k_{67}, k_{25} + k_{89}, k_1 + k_{65}, k_4 + k_{68}, k_{26} + k_{90}, k_6 + k_{70}, k_9 + k_{73}, k_{11} + k_{75}.$$

When $j = 2$, the 8 key coefficients are

$$k_5 + k_{69}, k_{27} + k_{91}, k_3 + k_{67}, k_6 + k_{70}, k_{28} + k_{92}, k_8 + k_{72}, k_{11} + k_{75}, k_{13} + k_{77},$$

Hence, there are three common key coefficients in the two key sets.

As a result, for the following 2^{125}-size key space,

$$\mathbb{K} = \left\{k \in \mathbb{F}_2^{128} : k_3 + k_{67} = 0, k_6 + k_{70} = 0, k_{11} + k_{75} = 0\right\},$$

we can do 10 cube testers to recover the other 10 key coefficients in the two key sets for $j \in \{0, 2\}$. The remaining 115 bits of key information can be recovered by an exhaustive search. The data and time complexities of this attack are $10 \times 2^{60} \approx 2^{63.2}$ which are less than the data limitation 2^{64}.

Comparing with Rohit and Sarkar's key-recovery attacks under the weak-key setting [RS21], our attack works for a significantly larger weak-key space ($2^{127.3}/2^{125}$ versus $2^{116.34}$).

Table 6: Algebraic degrees of Ascon-80pq in cube variables with the structure IS'_j ($32 \leq j < 64$) derived from the division properties. "x/y" means that the degrees of some bits of that word are x and others are y.

Round r	**Degrees of words**				
	$S_0^{(r)}$	$S_1^{(r)}$	$S_2^{(r)}$	$S_3^{(r)}$	$S_4^{(r)}$
0.5	1	1	0	1	1
1.5	2	2/1	2/1	2	2
2.5	4/3	4/3	4	4	4/3
3.5	8/7	8	8/7	8/7	7/6
4.5	16/15	16/15	14	15/14	16/15
5.5	31	30	30	31	31
6.5	60	60	61	61	60

5 Conditional Cube Attack on Ascon-80pq

Ascon-80pq is a relatively new member of Ascon-AEAD family. The main difference between Ascon-80pq and Ascon-128/Ascon-128a is that it uses the second 32 bits of IV positions for another 32 bits of key. Our attack in this section is designed to recover the 32-bit key. For the sake of convenience, we denote the 32-bit key by $k'_{32}, k'_{33}, \dots, k'_{63}$ (the indices starts from 32 because they are loaded into $S^{(0)}_{0,32}, \dots, S^{(0)}_{0,63}$). Once the 32-bit key is recovered, the previous conditional cube attack on Ascon-128 and Ascon-128a in Section 4 can be used to recover the remaining 128 key bits $k_0, \dots, k_{127}$.

To attack Ascon-80pq, we also take advantage of the 60-dimensional cube distinguisher of IS_{-1} with the break-fix strategy.

5.1 Phase 1: Break

32 new structures IS'_j ($32 \leq j < 64$) are introduced to break IS_{-1} as follows (we prefer $32 \leq j < 64$ rather than $0 \leq j < 32$ to match the positions of the 32-bit key in the first word),

$$IS'_j : \left(S_3^{(0)}[i], S_4^{(0)}[i]\right) = \begin{cases} (v_i, 0) & \text{if } i = j \\ (v_i, v_i) & \text{if } i \neq j \end{cases}, \quad 32 \leq i < 64$$

According to the ANFs of Ascon's Sbox, the ANF of the 4th output bit is

$$y_3 = x_4x_0 + x_4 + x_3x_0 + x_3 + x_2 + x_1 + x_0.$$

For IS_{-1} where $x_3 = x_4$, both x_3 and x_4 will be canceled, always leaving the degree of y_3 be 0. However, for IS'_j ($32 \leq j < 64$) where x_4 is set as 0 for the j^{th} Sbox, there will be a term $x_3(x_0 + 1)$ in its 4^{th} output bit of the j^{th} Sbox. The ANF of this output bit of the j^{th} Sbox is then

$$S^{(0.5)}_{3,j} = v_j(k'_j + 1) + k_j + k_{j+64} + k'_j. \tag{13}$$

Thus, the value of k'_j will directly control the degree of $S^{(0.5)}_{3,j}$. After the diffusion layer, this bit spreads to another 3 positions, and is multiplied by the p_S of the second round with other 1-degree bits to cause some 2-degree bits. With a similar division property model as [RHSS21], we calculate the degrees for Ascon-80pq with IS'_j which are shown in Table 6.

5.2 Phase 2: Fix

For IS'_j ($32 \leq j < 64$), if we reset the degree of $S^{(0.5)}_{3,j}$ to 0, the degrees of $S^{(6.5)}_0$ will return to 59. This can be guaranteed by the following proposition,

Proposition 2. *If the algebraic degrees of the five words of $S^{(0.5)}$ are $(1, 1, 0, 0, 1)$, the upper bounds on the algebraic degrees of the five words of $S^{(6.5)}$ are $(59, 59, 60, 60, 59)$.*

Proof. Note that the p_C operation does not influence the degree matrix of a state in this case. Therefore, the proposition can be proved by a direct application of Lemmas 2 and 3. □

According to Equation 13, if $k'_j + 1 = 0$, i.e., $k'_j = 1$, $\deg(S^{(0)}_{3,0}) = 0$, otherwise, $\deg(S^{(0)}_{3,0}) = 1$. Thus, we can observe the cube sum derived from IS'_j $(32 \le j < 64)$ to determine the value of k'_j. After trying all IS'_j $(32 \le j < 64)$, the 32-bit key $k'_{32}, k'_{33}, \ldots, k'_{64}$ are recovered.

Note that the concrete IV bits only contribute to the constant terms and do not appear in the coefficients of the maximal-degree cube variables in the first and second rounds, so they do not affect our attack in Section 4 on Ascon-128/Ascon-128a. Consequently, after recovering the 32-bit key in the first word, we apply Algorithm 1 to recover the remaining 128-bit key in the second and third words of the initial state.

Complexity. Since the recovery of the 32-bit key in the first word is completely an independent process which costs only $32 \times 2^{60} = 2^{65}$ data and time complexities, the main part of the complexities is determined by the recovery phase of the remaining 128-bit key. Thus, the data and time complexities are as the same as the attacks on Ascon-128/Ascon-128a in Section 4. In other words, for $2^{32+127.97} = 2^{159.97}$ keys, the data and time complexities are respectively 2^{70} and $2^{72.4}$. At the worst case, the time complexity can be as large as $2^{104.2}$. The memory cost is negligible.

6 Discussions on Assumptions in Our Attacks

Our attacks in Sections 4 and 5, like all previous conditional cube attacks, rely on some common assumptions. In our case, the assumption for 7-round Ascon initialization has been concluded in Assumption 1.

This assumption relies on how complex the superpolies of the cubes are. In our attacks on 7-round Ascon, the superpolies are the coefficients of the corresponding 60-degree cube terms, so, unfortunately, this assumption is hard to practically verify due to the huge complexity (practically verifying it needs to perform several 2^{60} Ascon initializations). Therefore, we test the validity of Assumption 1 for 5 and 6 rounds of Ascon. Although the experiments show that the assumption is not perfect for 5- and 6-round Ascon, we observe a clear trend that the likelihood of Assumption 1 grows sharply with increasing the number of rounds, which gives us the confidence that it still has a decent chance to be valid for 7 rounds.

6.1 Experiments on 5 and 6 Rounds of Ascon

According to Tables 3 and 4, the degree upper bounds for 5 rounds are 15 and 16 (some bits of $S^{(4.5)}_1$ have degrees of 16), for 6 rounds are 30 and 31, for IS_{-1} and $IS_j, 0 \le j < 64$, respectively. According to the degree details of $S^{(1.5)}$ which have been shown in Figure 4 and the ANFs of the three 2-degree bits given in Equation 3, if we set

$$v_{26}, v_6, v_9, v_{31},$$

as constants whereas $v_0, v_1, v_3, v_{25}, v_4$ as variables, the degrees of 5- and 6-round outputs are determined by the values of

$$k_1 + k_{65}, k_3 + k_{67}, k_{25} + k_{89}, \text{ and } k_4 + k_{68}.$$

If the four key coefficients are all zero, the 5-round degree is 15 and the 6-round degree is 30; otherwise, the degree can be 16 and 31 for 5 and 6 rounds, respectively.

To check the validity of Assumption 1 for 5- and 6-round Ascon, we randomly generate 100 keys where the four values

$$k_1 + k_{65}, k_3 + k_{67}, k_{25} + k_{89}, \text{ and } k_4 + k_{68}$$

are not zero. Then, we generate the cube by taking all different values of

$$v_0, v_1, v_3, v_{25}, v_4, v_{j_1}, v_{j_2}, \ldots, v_{j_n},$$

where $j_n = 16 - 5$ for 5-round Ascon and $j_n = 31 - 5$ for 6-round, and letting the remaining variables be constants for each of the 100 keys. If Assumption 1 works well, the cube sums for the 100 experiments will be non-zero.

For 5-round Ascon, unfortunately, all the 100 cube sums are zero. Therefore, Assumption 1 is definitely not valid for 5-round Ascon. The cube distinguisher cannot be used in a key-recovery attack by our break-fix strategy for 5 rounds. For 6-round Ascon, 54 experiments lead to non-zero cube sums. This means Assumption 1 holds with some probability for 6-round Ascon.

What about the case of 7-round Ascon? Although Assumption 1 is still not good for 6-round Ascon, it has become much better than the 5-round cases: Indeed, in the 5-round experiments, all 100 experiments lead to zero cube sums, while in the 6-round experiments, more than a half of experiments succeed. This is not a coincidence. The cube sum is the value of the superpolies of the cube term, i.e., the coefficients of the product of the cube variables. Thus, it is natural that as the number of rounds gets larger, the superpolies become more and more random. As the dimension of cubes used in our attacks is 60, we cannot practically verify Assumption 1 for 7-round Ascon. However, the results of our experiments on 5 and 6 rounds of Ascon show a clear trend that Assumption 1 is becoming more and more promising as the number of rounds increases.

The validity of Assumption 1 is crucial for the success of our attack on 7-round attack, while all cube-like attacks rely on similar assumptions (except for some with small cubes that can be verified by experiments). In the extremest case where there is no GKS, we need all the 4,480 cubes in our attacks to lead to non-zero cube sums. To this end, suppose that the probability for a cube sum to be zero when the four corresponding key coefficients are not all zero is smaller than p, then all the 4,480 cube sums are non-zero with a probability of about $(1-p)^{4,480}$. For $p = 2^{-20}$, $(1-p)^{4,480} \approx 0.996$ which shows the successful probability has been close to 1. Note that if we assume the 7-round Ascon is ideal, p should be 2^{-64}, so we believe that $p = 2^{-20}$ is not an overly harsh condition.

6.2 Key-Recovery Attack on 6-Round Ascon

Note that in our 100 experiments for 6-round Ascon, more than a half of them lead to a non-zero cubes. Thus, our break-fix strategy can be used to perform a key-recovery attack on 6-round Ascon. This experiments show that our break-fix strategy is useful as long as Assumption 1 is not entirely wrong. In this subsection, we describe this attack and practically verify it.

According to Tables 3 and 4, the degree upper bounds for 6 rounds are 30 and 31 for IS_{-1} and $IS_j, 0 \leq j < 64$, respectively. We take IS_0 as an example. The degree details of $S^{(1.5)}$ has been shown in Figure 4 and the ANFs of three 2-degree bits given in Equation 3. Therefore, if we set

$$v_3, v_{25}, v_4, v_{26}, v_6, v_9, v_{31},$$

as constants whereas v_0 and v_1 as variables, the degree of the 6-round output is determined by the value of $k_1 + k_{65}$. If $k_1 + k_{65} = 0$, then the 6-round degree is 30, otherwise, the degree is 31. As a result, we perform the following experiments to recover $k_1 + k_{65}$.

1. Choose v_0, v_1 and $v_{j_0}, v_{j_1}, \ldots, v_{j_{28}}, v_{j_i} \notin \{v_3, v_{25}, v_4, v_{26}, v_6, v_9, v_{31}\}$ for any $0 \leq i < 29$ as the 31 cube variables.

2. Prepare a cube where v_0, v_1 and $v_{j_0}, v_{j_1}, \ldots, v_{j_{28}}$ take all possible values and the remaining cube variables are set as random constants.

3. Apply the 6-round Ascon initialization to the cube, and summarize the results to observe the cube sum.

If the cube sum is non-zero, $k_1 + k_{65} = 1$; otherwise $k_1 + k_{65} = 0$. Due to the rotational property of the Ascon state, we can perform the above process for all IS_j to recover $k_{j+1} + k_{64+j+1}$.

What's more, for IS_j, we can recover keys by setting another set of 7 cube variables from

$$v_{j+1}, v_{j+3}, v_{j+25}, v_{j+4}, v_{j+26}, v_{j+6}, v_{j+9}, v_{j+31}$$

to be constants. For example, if we choose v_j, v_{j+3} together with 29 variables chosen from $\{v_0, \ldots, v_{63}\} \backslash \{v_j, v_{j+1}, v_{j+3}, v_{j+25}, v_{j+4}, v_{j+26}, v_{j+6}, v_{j+9}, v_{j+31}\}$ as cube variables and the remaining variables as constants, the cube sums will reflect the values of $k_{j+3} + k_{j+67}$.

Obviously, in total we have 8 opportunities to recover $k_j + k_{j+64}$ by setting different variables as constants. Thus, we repeat the above attack process 8 times, each of which uses $v_0, v_j, v_{j_0}, \ldots, v_{j_{29}}$ as cube variables and the remaining as constants. Also, we can repeat trying s different $v_{j_0}, \ldots, v_{j_{29}}$ to increase the precision of the key-recovery attack. If $k_j + k_{j+64}$ is found as 1 in any of the $8s$ times of experiments, $k_j + k_{j+64}$ is recognized as 1; otherwise, it is recognized as 0.

The whole time complexity (to recover the 64 key coefficients) of this attack $2^{31} \times 64 \times 8 \times s = 2^{40}s$ 6-round Ascon initializations. We practically perform this attack for 100 random keys. When setting $s = 2^3$, the 64 bits of information $k_j + k_{j+64}$ were recovered successfully for all the 100 keys. The code for the experiments is given in the git repository:

https://github.com/hukaisdu/Ascon_6R_Experiments.git

Discussion on 7-round attack. Similar to the 6-round case, whenever Assumption 1 is not entirely wrong, it will be useful for the key-recovery attack on 7-round Ascon. For 7-round attack, when we choose 60 cube variables, the remaining 4 variables are set as constants. The 4 varaibles are actually related to 8 nonce bits where different values of these 8 nonce bits will affect the values of the cube sums (note that two constant nonce bits n_i and n_{i+64} should not be equal, otherwise they will not affect the cube sum). As a result, we can repeat the attack process with different constant values to increase the successful probability with an increasing data and time complexity. Suppose we repeat the attack with 16 different values of these 8 nonce bits, and the cube sum is zero with a probability of 0.5 (0.5 is from the 6-round experiment, which should be reasonable considering the trend). Then, the probability of the event that all 16 times of experiments lead to zero cube sum is 2^{-16} only. In our worst case, all 4,480 cubes should lead to non-zero cubes, the probability is $(1 - 2^{-16})^{4,480} \approx 0.93$. That is to say, we can increase the successful probability to 0.93 with a cost of increasing the data/time complexity by a factor of 16. In this case, the complexity is still better than [LDW17].

7 Conclusion

In this paper, we proposed a new *break-fix* strategy for the conditional cube attack, which for the first time succeeded in transforming the cube distinguisher provided in [RHSS21] to a key-recovery attack, which was thought to be difficult before this paper. Thanks to the lower dimension of the cube distinguisher, our attacks led to improvements over the previous attacks on Ascon-128 and Ascon-128a, and gave the first key-recovery attack on Ascon-80pq. The break-fix strategy provides new insights into the conditional cube attacks. Different from most previous conditional cube attacks that tend to use an elimination strategy, we construct a higher-degree term by breaking the special cube structure and then fix the break to force the degree to go back. The basic idea of our attack is generic, so we believe that this strategy can find usage in other ciphers, which will be a direction of our future works. Besides, verifying Assumption 1 for the 7-round Ascon is also an interesting future work.

Acknowledgments

The author is grateful to Thomas Peyrin for his valuable comments on this paper. The author also wants to thank the anonymous referees for their comments that improved the quality of this article significantly. The authors are supported by the Program of Qilu Young Scholars of Shandong University, the France-Singapore NRF-ANR research grant NRF2020-NRF-ANR072, the Singapore NRF Investigatorship research grant NRF-NRFI08-2022-0013, the National Key Research and Development Program of China (Grant No. 2018YFA0704702), the National Natural Science Foundation of China (Grant No. 62032014, U2336207), the Major Basic Research Project of Natural Science Foundation of Shandong Province, China (Grant No. ZR202010220025), Department of Science & Technology of Shandong Province (No.SYS202201), Quan Cheng Laboratory (Grant No. QCLZD202301, QCLZD202306).

References

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube testers and key recovery attacks on reduced-round MD6 and trivium. In Orr Dunkelman, editor, *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2009.

[Ann] Announcing Lightweight Cryptography Selection | CSRC. https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon.

[Azz85] Adelchi Azzalini. A class of distributions which includes the normal ones. *Scandinavian journal of statistics*, pages 171–178, 1985.

[BCP22] Jules Baudrin, Anne Canteaut, and Léo Perrin. Practical cube attack against nonce-misused Ascon. *IACR Trans. Symmetric Cryptol.*, 2022(4):120–144, 2022.

[CHKT23] Donghoon Chang, Deukjo Hong, Jinkeon Kang, and Meltem Sönmez Turan. Resistance of Ascon family against conditional cube attacks in nonce-misuse setting. *IEEE Access*, 11:4501–4516, 2023.

[Dae12] Joan Daemen. Permutation-based Encryption, Authentication and Authenticated Encryption. DIAC 2012, 2012.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, pages 371–387, 2015.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *J. Cryptol.*, 34(3):33, 2021.

[DLWQ17] Xiaoyang Dong, Zheng Li, Xiaoyun Wang, and Ling Qin. Cube-like attack on round-reduced initialization of Ketje Sr. *IACR Trans. Symmetric Cryptol.*, 2017(1):259–280, 2017.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 278–299, 2009.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ASCON. *IACR Cryptol. ePrint Arch.*, page 1103, 2021.

[HLLT20] Phil Hebborn, Baptiste Lambin, Gregor Leander, and Yosuke Todo. Lower bounds on the degree of block ciphers. *IACR Cryptol. ePrint Arch.*, 2020:1051, 2020.

[HLM$^+$20] Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang. Modeling for three-subset division property without unknown subset - improved cube attacks against trivium and grain-128aead. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 466–495. Springer, 2020.

[HPTY22] Kai Hu, Thomas Peyrin, Quan Quan Tan, and Trevor Yap. Revisiting higher-order differential-linear attacks from an algebraic perspective. *Cryptology ePrint Archive*, 2022.

[HSWW20] Kai Hu, Siwei Sun, Meiqin Wang, and Qingju Wang. An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums. *IACR Cryptol. ePrint Arch.*, 2020:1048, 2020.

[HWX$^+$17] Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao. Conditional Cube Attack on Reduced-Round Keccak Sponge Function. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 259–288, 2017.

[Jea16] Jérémy Jean. TikZ for Cryptographers. https://www.iacr.org/authors/tikz/, 2016.

[LBDW17] Zheng Li, Wenquan Bi, Xiaoyang Dong, and Xiaoyun Wang. Improved Conditional Cube Attacks on Keccak Keyed Modes with MILP Method. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 99–127. Springer, 2017.

[LDW17] Zheng Li, Xiaoyang Dong, and Xiaoyun Wang. Conditional cube attack on round-reduced ASCON. *IACR Trans. Symmetric Cryptol.*, 2017(1):175–202, 2017.

[LLL21] Meicheng Liu, Xiaojuan Lu, and Dongdai Lin. Differential-linear cryptanalysis from an algebraic perspective. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 247–277. Springer, 2021.

[LZWW17] Yanbin Li, Guoyan Zhang, Wei Wang, and Meiqin Wang. Cryptanalysis of round-reduced ASCON. *Sci. China Inf. Sci.*, 60(3):38102, 2017.

[RHSS21] Raghvendra Rohit, Kai Hu, Sumanta Sarkar, and Siwei Sun. Misuse-free key-recovery and distinguishing attacks on 7-round Ascon. *IACR Trans. Symmetric Cryptol.*, 2021(1):130–155, 2021.

[RS21] Raghvendra Rohit and Santanu Sarkar. Diving deep into the weak keys of round reduced Ascon. *IACR Trans. Symmetric Cryptol.*, 2021(4):74–99, 2021.

[SG18] Ling Song and Jian Guo. Cube-Attack-Like Cryptanalysis of Round-Reduced Keccak Using MILP. *IACR Trans. Symmetric Cryptol.*, 2018(3):182–214, 2018.

[SGSL18] Ling Song, Jian Guo, Danping Shi, and San Ling. New MILP Modeling: Improved Conditional Cube Attacks on Keccak-Based Constructions. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2018.

[Tez20] Cihangir Tezcan. Analysis of Ascon, DryGASCON, and Shamash permutations. *IACR Cryptol. ePrint Arch.*, page 1458, 2020.

[TIHM17] Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier. Cube attacks on non-blackbox polynomials based on division property. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 250–279. Springer, 2017.

[TM16] Yosuke Todo and Masakatu Morii. Bit-based division property and application to simon family. In Thomas Peyrin, editor, *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, volume 9783 of *Lecture Notes in Computer Science*, pages 357–377. Springer, 2016.

[TMC$^+$23] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Lawrence E Bassham, Jinkeon Kang, Noah D Waller, John M Kelsey, and Deukjo Hong. Status report on the final round of the nist lightweight cryptography standardization process. 2023.

[Tod15] Yosuke Todo. Structural evaluation by generalized integral property. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 287–314, 2015.

[WHG+19] Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi. Milp-aided method of searching division property using three subsets and applications. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 398–427. Springer, 2019.

[WHT+18] Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier. Improved division property based cube attacks exploiting algebraic properties of superpoly. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 275–305. Springer, 2018.

[XZBL16] Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin. Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 648–678, 2016.