IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2023, No. 4, pp. 420–451. DOI:10.46586/tosc.v2023.i4.420-451

Committing Security of Ascon: Cryptanalysis on Primitive and Proof on Mode

Yusuke Naito[1], Yu Sasaki[2] and Takeshi Sugawara[3]

[1] Mitsubishi Electric Corporation, Kanagawa, Japan
Naito.Yusuke@ce.MitsubishiElectric.co.jp

[2] NTT Social Informatics Laboratories, Tokyo, Japan
yusk.sasaki@ntt.com

[3] The University of Electro-Communications, Tokyo, Japan
sugawara@uec.ac.jp

Abstract. Context-committing security of authenticated encryption (AE) that prevents ciphertexts from being decrypted with distinct decryption contexts, (K, N, A) comprising a key K, a nonce N, and associate data A is an active research field motivated by several real-world attacks. In this paper, we study the context-committing security of Ascon, the lightweight permutation-based AE selected by the NIST LWC in 2023, for cryptanalysis on primitive and proof on mode. The attacker's goal is to find a collision of a ciphertext and a tag with distinct decryption contexts in which an attacker can control all the parameters including the key. First, we propose new attacks with primitives that inject differences in N and A. The new attack on Ascon-128 improves the number of rounds from 2 to 3 and practically generates distinct decryption contexts. The new attack also works in a practical complexity on 3 rounds of Ascon-128a. Second, we prove the context-committing security of Ascon with zero padding, namely Ascon-zp, in the random permutation model. Ascon-zp achieves $\min\left\{\frac{t+z}{2}, \frac{n+t-k-\nu}{2}, \frac{c}{2}\right\}$-bit security with a t-bit tag, a z-bit padding, an n-bit state, a ν-bit nonce, and a c-bit inner part. This bound corresponds to $\min\left\{64 + \frac{z}{2}, 96\right\}$ with Ascon-128 and Ascon-128a, and $\min\left\{64 + \frac{z}{2}, 80\right\}$ with Ascon-80pq. The original Ascon ($z = 0$) achieves 64-bit security bounded by a generic birthday attack. By appending zeroes to the plaintext, the security can be enhanced up to 96 bits for Ascon-128 and Ascon-128a and 80 bits for Ascon-80pq.

Keywords: Ascon · Authenticated Encryption · Key Commitment · Context Commitment · Differential Cryptanalysis · MILP · Security Proof · Zero Padding

1 Introduction

Authenticated encryption with associated data (AE) methods that ensure confidentiality and authenticity are critical components in symmetric-key cryptography. The security of AE has been extensively studied, and the schemes often provide security proofs based on a formal security notion. However, AE schemes are sometimes abused in ways that go beyond their intended purpose, leading to security issues. This category includes committing security of AEs, which has been widely studied in recent years [FOR17, GLR17, DGRW18, LGR21, ADG+22, BH22, CR22, MLGR23, BCG+23].

A nonce-based AE encryption takes a key K, nonce N, associated data A, and a plaintext M to generate a ciphertext C and a tag T. The decryption takes (C, T) and the triple (K, N, A) called *the decryption context* and outputs either the original plaintext M

Received: 2023-09-01 Accepted: 2023-11-01 Published: 2023-12-08

or the invalid symbol $\perp$.[1] An attacker wins the game of committing security by finding a ciphertext and a tag (C, T) and the distinct decryption contexts (K, N, A) and (K', N', A') such that decryption succeeds with both contexts. Farshim et al. initiated the field by studying the cases with $K \neq K'$, now called *key commitment* [FOR17, GLR17].

The conventional AE security notions do not imply key-committing security, and there are attacks on popular schemes, including GCM [GLR17, DGRW18], GCM-SIV [LGR21], CCM [Dwo07, MLGR23], and ChaCha20-Poly1305 [GLR17, NL18]. These attacks even lead to application-level attacks, e.g., the multi-recipient integrity attack that targets a specific user and sends malicious content to them and the partitioning oracle attack that effectively performs password brute-force attacks [LGR21]. Researchers are studying AE schemes with committing security to address the issue [GLR17, DGRW18, LGR21, ADG+22]. Meanwhile, standardization organizations are also starting to consider committing security. For instance, key commitment appears in the RFC draft classifying the properties of AEAD algorithms [Boz23]. Besides, NIST's workshop on updating block-cipher modes explicitly mentions key commitment as an extra security feature [NIS23b].

Bellare and Hoang [BH22] proposed generalized security notions called **CMT**-1, **CMT**-3, and **CMT**-4. **CMT**-1 represents the conventional key commitment with $K \neq K'$. Meanwhile, **CMT**-3 considers commitment to an entire decryption context, i.e., $(K, N, A) \neq (K', N', A')$, called *context commitment*. **CMT**-4 considers commitment to a plaintext and a decryption context, which is shown to be equivalent to **CMT**-3. Throughout this paper, we consider **CMT**-3 for cryptanalysis because it requires less attacker capability. Meanwhile, we consider the most general **CMT**-4 for security proofs. Chan and Rogaway independently studied the generalized definitions of committing AEs [CR22]. **CMT**-4 covers a broader range of misuses and is strictly more secure than **CMT**-1. Consequently, ensuring and building AEs with **CMT**-4 security is an ongoing research challenge [BH22, CR22, MLGR23].

Besides the research on committing security, lightweight cryptography has been an important subject driven by the NIST LWC competition. The competition finally selected Ascon in 2023 [NIS23a]. Ascon is a family of cryptographic schemes using the Ascon permutation, including AEAD (Ascon-128, Ascon-128a, and Ascon-80pq) and hash functions (Ascon-Hash and Ascon-Xof) [DEMS21]. Ascon AEADs are based on the duplex construction [BDPV11] but use a stronger keyed initialization and keyed finalization function [DEMS21]. NIST is going to standardize Ascon, and real-world systems will migrate to Ascon in the near future. *Will migration to Ascon solve the problems regarding committing security?* This research question is non-trivial because Ascon has several essential differences from the conventional standards.

1.1 State-of-the-Art and Challenges

Security Proof. Collision-resistant hash functions can be used to achieve committing security [DGRW18, BH22, CR22]. For permutation-based schemes, AEs based on the duplex construction can easily achieve the committing security because Bertoni et al. [BDPV11] showed that its security is reducible to the indifferentiability of the sponge construction [BDPV08]. The duplex construction shown in Figure 1 absorbs an r-bit data block into the outer part, performs a permutation, and outputs the outer part. The output can be seen as that from a random oracle (RO) up to the $\frac{c}{2}$-bit bound by the indifferentiablity of the sponge construction, where c is the capacity. For example, Dodis et al. proposed a concrete duplex-based scheme that satisfies the key-committing security [DGRW18].

However, Ascon has essential differences from the duplex-based AE schemes. First, Ascon initializes the state using a ν-bit nonce and a k-bit key, enabling a **CMT**-4 attacker

[1] More generally, T can be considered as a part of a ciphertext. The generalized definition covers non-tag-based AE schemes such as AEZ [HKR15].

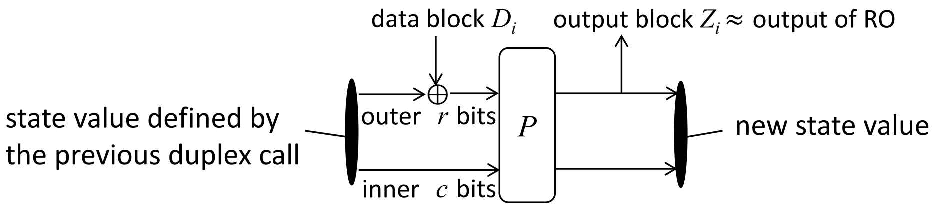
data block D_i
output block $Z_i \approx$ output of RO
state value defined by the previous duplex call
outer r bits
inner c bits
P
new state value

Figure 1: Duplex Construction [BDPV11]. P is a $(r+c)$-bit permutation.

to control $(\nu + k)$ bits of the initial state, including the inner part. Second, Ascon emits t bits of the final inner part as a tag, in addition to the r-bit outer part. As a result, the security of the duplex construction no longer ensures the **CMT**-4 security, and there is no guarantee unless there is a new proof.

Besides the security proof, a generic attack upper-bounds the security to $t/2$ bits. The attacker computes the decryption function offline with a fixed ciphertext while changing (K, N, A) until getting a tag collision. This corresponds to 64 bits for Ascon-128, Ascon-128a, and Ascon-80pq with $t = 128$.

Cryptanalysis on Primitives. There are many research works studying the attacks on the primitives of Ascon for AEAD or hashing. We would like to know if they can be used to attack the context-committing security with less than 64 bits of the generic attack. Table 1 summarizes a part of the state-of-the-art attacks of the Ascon family that is particularly relevant to our paper.

The conventional AEAD attacks include (i) key recovery, (ii) forgery, and (iii) state recovery. The key-recovery attacks aim at recovering a secret key [LDW17, Tez20, GPT21, RHSS21]. In contrast, the **CMT**-3 attacker aims at generating colliding data by choosing keys. Those two goals are significantly different.

Gerault et al.'s 3-round forgery attack on iteration that injects differences in consecutive AD blocks succeeds with 2^{117} complexity [GPT21] and potentially breaks the **CMT**-3 security. Unlike the forgery attack, a **CMT**-3 attacker can observe the internal state using a known key to find the paired values satisfying the differential trail more efficiently. This can significantly reduce the complexity from 2^{117}, and the attack succeeds if the complexity becomes lower than 2^{64}, i.e., the birthday bound by a generic attack. Although there are more efficient forgery attacks on finalization [DEMS15, GPT21], they are not useful for attacking the **CMT**-3 security because (i) a difference in plaintexts generates different ciphertexts and (ii) the differences in the truncated part ignored in the finalization attacks are non-negligible in analyzing either initialization or iteration.

The attacks on Ascon hashing are more relevant to the **CMT**-3 security because of the keyless setting. In particular, the collision attacks exploiting the internal state collision [ZDW19, GPT21, YLW$^+$23] can be used to break the **CMT**-3 security like the AEAD forgery attack. The state-of-the-art in this category is the one by Yu et al. [YLW$^+$23] that breaks 2 rounds of Ascon-Hash with $2^{62.6}$ complexity. This attack is directly applicable to 2 rounds of Ascon-128 and Ascon-128a. Meanwhile, the other hashing attacks are somewhat irrelevant to the **CMT**-3 security. The semi-free-start collision attack [YLW$^+$23] propagates the difference from the middle to the initial, which is efficiently prevented by the feed-forward structure in the initialization of the AEAD mode. Similarly, making a ciphertext that can be decrypted with different decryption contexts is non-trivial even if a preimage is available [DEMS21]. Moreover, applications of the preimage attacks on Ascon-Xof with 64-bit outputs to the committing security is unclear because of the differences of the finalization function in the AEAD mode and the hashing mode.

In summary, the **CMT**-3 security of 2 rounds of Ascon-128 can be attacked with $2^{62.6}$ complexity [YLW$^+$23], which is already better than the generic attack. Meanwhile,

Table 1: New and conventional attacks on Ascon relevant to the **CMT**-3 security.

Target	Type	Round	Complexity	Ref.
Ascon-128	Forgery, Finalization	4	2^{101}	[DEMS15]
Ascon-128	Forgery, Finalization	4	$2^{96.61}$	[GPT21]
Ascon-128a	Forgery, Iteration	3	2^{117}	[GPT21]
Ascon-128a	Forgery, Finalization	3	2^{20}	[GPT21]
Ascon-Xof	Collision	2	2^{15}	[ZDW19]
Ascon-Hash	Collision	2	2^{125}	[ZDW19]
Ascon-Hash	Collision	2	2^{103}	[GPT21]
Ascon-Hash	Collision	2	$2^{62.6}$	[YLW+23]
Ascon-Hash	Semi-free-start collision	4	Practical	[YLW+23]
Ascon-Xof	Preimage	2	2^{39}	[DEMS21]
Ascon-Xof	Preimage	6	$2^{63.3}$	[DEMS21]
Ascon-128, Ascon-80pq	**CMT**-3	2	$2^{62.6}$	[YLW+23]
Ascon-128a	**CMT**-3	3	$2^{117\dagger}$	[GPT21]
Ascon-128, Ascon-80pq	**CMT**-3	3	$2^{48\ddagger}$	This Work
Ascon-128a	**CMT**-3	3	$2^{36\ddagger}$	This Work

†Validity of the attack has not been confirmed. It is valid only if the differential trail with probability 2^{-116} can be satisfied with a complexity below 2^{64} by using the knowledge of the internal state values.
‡The actual attack complexity is based on the MILP solving algorithm and should be much smaller. However we refer to this number to provide certain convenience for future reference.

the attack on 3 rounds of Ascon-128a with 2^{117} complexity [GPT21] has room for improvements, but it is far beyond practical.

Security Enhancement. Even if there is a security proof, the offline security of Ascon-128, Ascon-128a, and Ascon-80pq is upper bounded to 64 bits by the generic attack, which is significantly lower than that of the key-recovery attack. 64-bit offline security is insufficient in practice, and Chan and Rogaway recommended 80 bits or more [CR22]. Therefore, achieving higher **CMT**-4 security by applying a blackbox conversion to Ascon is another important challenge.

Two conventional approaches are Hash-then-Encrypt (HtE) [BH22] and CTX [CR22], which add a hash function before or after an AE encryption to use collision resistance for improving the **CMT**-4 security. However, HtE's security is limited by the same generic attack. Meanwhile, CTX requires an unverified tag for decryption and cannot be used as a blackbox conversion. Additional costs for a hash function are another concern.

In contrast, Albertini et al.'s Padding Fix [ADG+22] appends zeroes to a plaintext and checks them after decryption, which maintains compatibility with the original AE and can be more efficient. The security after padding should be evaluated for each scheme, and Albertini et al. proved that z-bit padding improves the **CMT**-1 security of AES-GCM and Chacha20-Poly1305 from 0 to $\frac{z}{2}$ bits. The security of Ascon with zero padding needs a dedicated security evaluation, which includes the following open research questions: (i) the effectiveness of Padding Fix in the **CMT**-4 (cf. **CMT**-1) security and (ii) improving the security beyond $\frac{z}{2}$ bits by using a tag and zero padding.

1.2 Contributions

Cryptanalysis on Primitives (Section 3). We propose new attacks that inject differences in the nonce and the first AD block. The setting of injecting differences in the nonce is specific to the **CMT**-3 security and was not considered by the state-of-the-art attacks [GPT21,

Table 2: Ascon's recommended parameters and the corresponding **CMT**-4 bounds

Name	Key	Nonce	Tag	State	Rate	Capacity	Rounds		**CMT**-4 Bound[‡]
Parameter	k	ν	t	$n^{\dagger}$	r	c	P_1	P_2	
General	—	—	—	—	—	—	—	—	$\min\left\{\frac{t+z}{2}, \frac{n+t-k-\nu}{2}, \frac{c}{2}\right\}$
Ascon-128	128	128	128	320	64	256	12	6	$\min\left\{64+\frac{z}{2}, 96\right\}$
Ascon-128a	128	128	128	320	128	192	12	8	$\min\left\{64+\frac{z}{2}, 96\right\}$
Ascon-80pq	160	128	128	320	64	256	12	6	$\min\left\{64+\frac{z}{2}, 80\right\}$

[†]The state size $n = 320$ is not parameterized.
[‡]z is the number of padded zero bits.

YLW+23]. This approach enables a better differential trail because the attacker can directly control 256 bits of the state by choosing K and N. Moreover, the attacker can inject 128-bit differences to N while the controllable data size in the previous hash collision attacks was limited by the data block size (or rate), e.g., 64 bits in the outer part, and other state bits are the output of the previous block, which behaves only randomly.

To search for differential trails, considering the resource available to us, we use the mixed integer linear programming (MILP). Simple applications of existing MILP modeling generate a too large problem to be solved. Hence, we add heuristics to finish the search in a reasonable time. Also, it is important to confirm validity of the trails. As [YLW+23], we search for actual paired values with a tool, MILP in our case, by setting value conditions to satisfy the differential trail. It is interesting that our 3-round trail for Ascon-128 can only be satisfied if the 64-bit IV constant defined in the specification, `80400c0600000000`, has two bits of '1' in bit position j and $j-12$ for some $j \in \{0, \ldots, 63\}$. Because bit '1' is sparse in IV, it is fortunate for us that this configuration is satisfied. As summarized in Table 1, our attack breaks **CMT**-3 security up to 3 rounds with Ascon-128, which improves the attack using the 2-round collision attack on Ascon-Hash by 1 round. Moreover, we practically generate a collision pair. With Ascon-128a, while the applicability of the previous work is unclear, we successfully generate an actual collision pair for 3 rounds.

Proof on Mode (Sections 4 and 5). We consider Ascon-zp, the Ascon with z-bit zero padding that extends a plaintext M to $M \| 0^z$, and prove its **CMT**-4 security in the random permutation (RP) model. We prove that Ascon-zp achieves $\min\left\{\frac{t+z}{2}, \frac{n+t-k-\nu}{2}, \frac{c}{2}\right\}$-bit **CMT**-4-security with an n-bit state and the relation $k \geq t$. The bound with $z = 0$ represents the security of the original Ascon.

We directly prove the **CMT**-4 security of Ascon-zp. Unlike the existing proofs of duplex-based schemes, the proof must take into account the Ascon's structural features: an attacker can control the inputs to Ascon including the key, i.e., the $(k+\nu)$-bit of the initial state is controllable; an attacker can obtain the t bits of the inner part, i.e., r bits of the outer part and t bits of the inner part are observable. These features might degrade the security to $\frac{n-\max\{k+\nu, r+t\}}{2}$ bits. However, regarding the initial state, the key masking in Ascon serves as the feed-forward operation of Davies-Meyer mode and prevents security degradation. Regarding the t-bit output, two permutations P_1 and P_2 prevent security degradation. The attacker obtains only t bits from P_1 used in initialization and finalization. the remaining bits, including the outer part, are unavailable. In contrast, the attacker only obtains r bits in the outer part from P_2 used in the other processes. By the feed-forward and the use of the two permutations, we can prove the $\min\left\{\frac{t+z}{2}, \frac{n+t-k-\nu}{2}, \frac{c}{2}\right\}$-bit **CMT**-4-security.

With the concrete parameters, the obtained bound becomes $\min\left\{64+\frac{z}{2}, 96\right\}$ for Ascon-128 and Ascon-128a, and $\min\left\{64+\frac{z}{2}, 80\right\}$ for Ascon-80pq, as summarized in

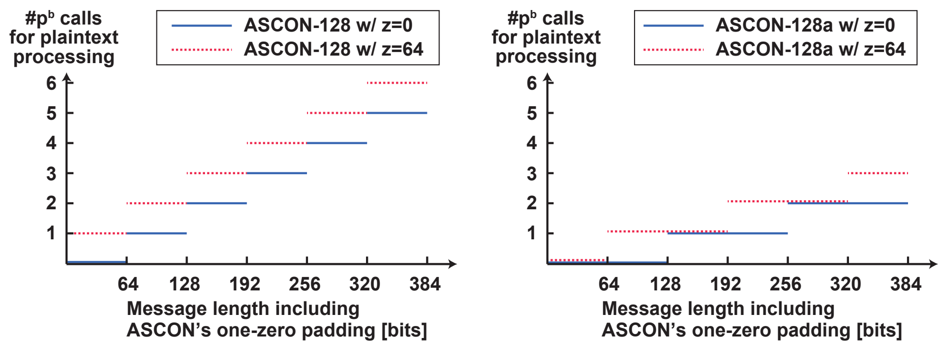

Figure 2: The number of the primitive (p^b) calls with and without 64-bit zero padding with (left) Ascon-128 and (right) Ascon-128a.

Table 2. The original AEAD modes achieve 64-bit **CMT**-4 security, which can be improved up to 96 (resp. 80) bits by adding 64 (resp. 32) zero bits for Ascon-128 and Ascon-128a (resp. Ascon-80pq). Fig. 2 compares the number of primitive call with and without 64-bit zero padding for Ascon-128 and Ascon-128a. One additional primitive call is necessary for Ascon-128 with 64-bit padding because $r = 64$. In case of Ascon-128a with $r = 128$, on the other hand, there is a chance of causing no extra primitive call when the original message has a 64-bit space until the block boundary.

1.3 Organization

We begin by recalling Ascon in Section 2. Section 3 gives the attacks on Ascon-128, Ascon-128a, and Ascon-80pq. Then, **CMT**-4 security bound of Ascon with zero padding is described in Section 4, followed by the security proof in Section 5. Section 7 is conclusion.

2 Specification of Ascon

2.1 Ascon Mode

The encryption of Ascon takes a key K, a nonce N, associated data (AD) A, and a plaintext M as input and returns a ciphertext C and a tag T. Ascon is parameterized by the size of key k, nonce ν, tag t, and data block (rate) r, and the number of rounds for the initialization and finalization a and for processing AD and plaintext b. With the state size n, r determines the size of non-data-block bits (capacity) c. Ascon specifies recommended parameters. The instance for the primary and secondary choices are called Ascon-128 and Ascon-128a, respectively. Besides, Ascon-80pq providing 80-bit security for post-quantum security is specified. The recommended parameters are listed in Table 2.

Ascon operates on a state of 320 bits. The encryption of Ascon is divided into four parts: initialization, processing AD, processing plaintext/ciphertext, and finalization. A diagram of those operations are depicted in Fig. 3.

Initialization. The 320-bit state $\mathcal{S}$ is formed by a key K, a nonce N, and $320 - k - 128$ bits of constant $IV_{k,r,a,b}$ including the parameter information of k, r, a, and b, each written as an 8-bit integer. Specifically, $IV_{k,r,a,b}$ is defined by $k\|r\|a\|b\|0^{160-k}$, which is `80400c0600000000`, `80800c0800000000`, and `a0400c06` for Ascon-128, Ascon-128a, and Ascon-80pq, respectively. Then, $\mathcal{S}$ is set to $IV_{k,r,a,b}\|K\|N$, and updated by $\mathcal{S} \leftarrow P_1(\mathcal{S}) \oplus (0^{320-k}\|K)$, where P_1 is a rounds of the round transformation.

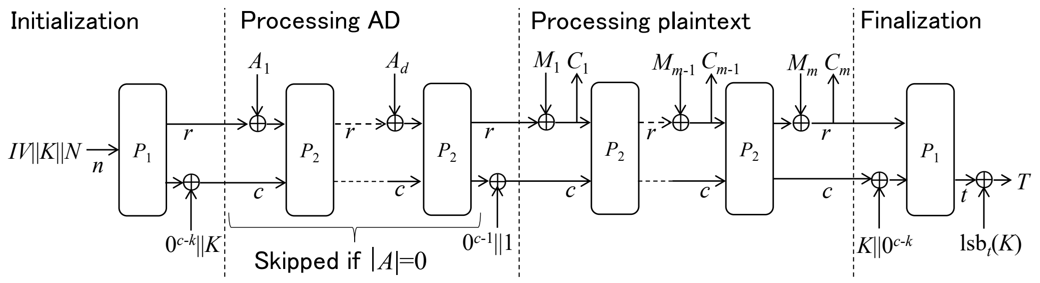

Figure 3: Encryption of Ascon.

Table 3: Ascon's 5-bit S-box S as a lookup table.

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
$S(x)$	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17

Processing AD. If A is non-empty, first A is padded by appending a single bit of 1 and the smallest number of 0s to be a multiple of r bits. It is then split into r-bit blocks $A_1, \ldots, A_d$. If A is empty, the padding is omitted. Then, $\mathcal{S} \leftarrow P_2((\mathcal{S}_r \oplus A_i)\|\mathcal{S}_c)$ is computed for $i = 1, \ldots, d$, where $\mathcal{S}_r$ and $\mathcal{S}_c$ are the first r bits and the last $320-r$ bits of $\mathcal{S}$, respectively, and P_2 is b rounds of the round transformation. After processing A_a, a 1-bit domain separation constant is XORed to $\mathcal{S}$: $\mathcal{S} = \mathcal{S} \oplus (0^{319}\|1)$.

Processing plaintext/ciphertext. Similarly to processing AD, M is padded to a multiple of r bits by the one-zero padding, and then is split into r-bit blocks $M_1, \ldots, M_m$. For each block except the last one, the ciphertext block C_i is computed by $\mathcal{S}_r \oplus M_i$, then $\mathcal{S}$ is updated with $P_2(C_i\|\mathcal{S}_c)$. For the last block m, C_m is truncated to match the input plaintext length and the update by P_2 is not computed.

Finalization. In finalization, K is XORed to $\mathcal{S}$ and $\mathcal{S}$ is updated by P_1. The tag T consists of the least significant 128 bits of $\mathcal{S}$ XORed with the last 128 bits of K.

2.2 Ascon Permutations

The 320-bit state $\mathcal{S}$ is split into five 64-bit words $x_i, i = 0, \ldots, 4$, hence $\mathcal{S} = x_0\|x_1\|x_2\|x_3\|x_4$, which is represented by a 5×64-bit two-dimensional array or a 320-bit one-dimensional array. $\mathcal{S}$ is iteratively updated by an SPN-based round transformation a (resp. b) times for P_1 (resp. P_2). The round transformation applies the three operations; round constant addition p_C, a substitution layer p_S, and a linear layer p_L.

- p_C: XOR an 8-bit constant value to the least significant 8 bits of x_2.
- p_S: Update $\mathcal{S}$ with 64 parallel applications of the 5-bit S-box $S(x)$ defined in Table 3 to each bit-slice of the five registers $x_0, \ldots, x_4$.
- p_L: Apply a linear function defined below to each word x_i.

$$\begin{aligned}
x_0 &\leftarrow x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28) & x_3 &\leftarrow x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)\\
x_1 &\leftarrow x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39) & x_4 &\leftarrow x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)\\
x_2 &\leftarrow x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6) & &
\end{aligned}$$

3 Reduced-Round Attacks to Break CMT-3 Security

3.1 Framework

Attacker's goal to break **CMT**-3 security is to find a pair of input tuples (K, N, A, M) and (K', N', A', M') that output the same C, T and satisfy $(K, N, A) \neq (K', N', A')$. In this paper, we set $K = K'$ because otherwise differential propagation needs to be controlled both in the initialization and the finalization. Then, there are two possible approaches.

- Inject a difference to A_i and cancel it with a difference of A_{i+1} after the permutation.
- Inject a difference to N and cancel it with a difference of A_1 after the initialization.

The first approach has been taken by the previous 2-round collision attacks on Ascon-Hash [ZDW19, GPT21, YLW+23] and the forgery attack on 3-round Ascon-128a [GPT21, Table 22]. The 2-round collision attack on Ascon-Hash with 64-bit rate [YLW+23] only requires $2^{62.4}$ complexity, hence it immediately forms a valid **CMT**-3 attack against 2-round Ascon-128, Ascon-128a, and Ascon-80pq with the same complexity. The 3-round forgery attack against Ascon-128a uses the differential trail with a probability of 2^{-116}. Hence, the straightforward application of the trail would exceed the generic attack complexity of 2^{64} based on the birthday-level security of the tag size. Because **CMT**-3 security operates in the keyless setting, the attack complexity could be improved significantly from 2^{117}. However, it would be expected that the attack complexity of both those 2-round and 3-round attacks remain quite expensive to be a practical attack.

In this paper, we take the second approach. The main advantage is that the attacker can directly choose values of four rows, and have two rows to inject differences even for Ascon-128. The main drawback is that x_0 are fixed to $IV_{k,r,a,b}$, which behaves as a constraint. To search for differential trails, using automated tools such as SAT, MILP, and CP is a popular approach. Considering the resource available to us, we use MILP.

3.2 Basics of Mixed Integer Linear Programming

The basic framework of differential trail search by MILP is well summarized in [MWGP11, SHW+14]. To be solved by MILP, we need to create the model that consists of declaration of variables, constraints defined by linear inequalities, and an objective function.

Variables. For 3-rounds, we use 3×320 binary variables X_{320i+j} to denote whether each bit before p_S is active or not, where $i = 0, 1, 2$ denotes the round and $j = 0, \ldots, 319$ denotes the bit position. If the bit is active, X_{320i+j} is 1. Otherwise, it is 0. We also use 3×320 binary variables Y_{320i+j} to denote whether each bit after p_S is active or not. Besides, we use binary variables $n_{64i+l}, l = 0, \ldots, 63$ to denote if each S-box is active or not.

Constraints by linear inequalities. We mainly need to set up three types of constraints: relationships between n_{64i+l} and X_{320i+j}, relationships between X_{320i+j} and Y_{320i+j} over p_S, and relationships between Y_{320i+j} and $X_{320(i+1)+j}$ over p_L.

For the first type, we set n_{64i+l} to 0 when the lth S-box in the ith round is inactive, namely, when X_{360i+l}, $X_{360i+64+l}$, $X_{360i+128+l}$, $X_{360i+192+l}$, and $X_{360i+256+l}$ are all 0. We set n_{64i+l} to 1 otherwise. For simplicity, we explain the case with $i = 0$ and $l = 0$, i.e. n_0 is defined based on 5 input bits $X_h, h \in \{0, 64, 128, 192, 256\}$. Our set up is achieved by 6 inequalities $\Sigma_{h \in \{0,64,128,192,256\}} X_h - n_0 \geq 0$ and $n_0 - X_h \geq 0, h \in \{0, 64, 128, 192, 256\}$.

For the second type, we model Ascon's differential distribution table (DDT). Several researchers have proposed the way to model DDT. We follow $*$-DDT by [ST17] which only cares whether each entry of DDT is zero or non-zero. In [ST17], $*$-DDT is introduced because their goal is to identify the impossible differentials determined only by the possibility of differential transitions. **CMT**-3 security studied in this paper is the keyless notion. Thus, even if the probability is low, an attacker can choose paired values satisfying the differential transition as long as the probability is non-zero. This matches the goal of $*$-DDT. How to derive inequalities to model DDT is the same as the previous work and we omit details. A set of linear inequalities is shown in Sect. A in Supplementary Material.

For the third type, because p_L only computes three-input XORs, the model is simple. The propagation over $A \oplus B \oplus C = D$ can be modeled as follows;

$$\begin{array}{llll} A+B+C-D \geq 0, & A+B-C+D \geq 0, & A-B+C+D \geq 0, & -A+B+C+D \geq 0, \\ -A-B-C+D \geq -2, & -A-B+C-D \geq -2, & -A+B-C-D \geq -2, & A-B-C-D \geq -2. \end{array}$$

Finally, we set the conditions to restrict the active bit positions to generate a collision. Difference cannot be injected to $IV_{k,r,a,b}$ and K, which derives $X_j = 0$ for $j = 0, \ldots, 191$. Some difference needs to be injected to N, which derives $X_{192} + \cdots + X_{319} \geq 1$. For the difference to be canceled after three rounds, we need conditions $Y_j = 0$ for $j = 704, \ldots, 959$ for Ascon-128 and $j = 768, \ldots, 959$ for Ascon-128a.

Objective Function. In general, it is desired if the number of active S-boxes is small. Hence, the objective function is to minimize $n_0 + \cdots + n_{191}$.

3.3 Sparse DDT for the First and Last Rounds

Because the problem space is large, it is hard to efficiently find solutions with the straightforward application of the basic model. One complex part is the S-box. Gerault et al. [GPT21] observed that during the first and the last round, only a small fraction of DDT can be exploited, thus there is no need to model the entire DDT. This strategy can also be applied to our setting. More specifically, the standard DDT contains the information computed by

$$\mathrm{DDT} := \forall \Delta_i, \forall \Delta_o, \frac{\#\{x \in \{0,1\}^5 | S(x) \oplus S(x \oplus \Delta_i) = \Delta_o\}}{2^5}.$$

In our setting, the MSB of the input is fixed to $\mathbf{b} \in \{0,1\}$ by $IV_{k,r,a,b}$ and the input difference can only take 0, 1, 2, and 3. Hence, the following distribution table is considered.

$$\mathrm{DDT}^{\mathbf{b}} := \forall \Delta_i \in \{0,1,2,3\}, \forall \Delta_o, \frac{\#\{x \in \{0,1\}^4 | S(\mathbf{b}\|x) \oplus S(\mathbf{b}\|x \oplus \Delta_i) = \Delta_o\}}{2^4}.$$

This makes the table significantly simple, and can be modeled with a few linear inequalities. The comparison of DDT and DDT^0 are given in Sect. B in Supplementary Material. The set of inequalities to model those tables are also compared in Sect. A.

3.4 Practical 3-Round CMT-3 Attack on Ascon-128a

Search Strategy. Even with the sparse $\mathrm{DDT}^{\mathbf{b}}$, the problem is too large to finish in a reasonable time in our environment, which requires us to add heuristics. Considering the fast diffusion by p_L, our strategy is to keep many 64-bit words x_i inactive. Specifically, we limit that the number of active words at the beginning of the second round is 2 or less. This implies that the Hamming weight of the output difference of each S-box in the first round is 2 or less. Considering that the input difference can only be 01, 02, or 03, the possible output differences are 01, 05, 09, 10, 11, 14, or 18. None of them has a difference in the second bit. Hence we further have a 64-bit condition $y_i = 0$, $i = 192, \ldots, 255$ on $\mathcal{S}$ after p_S, and also have $x_i = 0$, $i = 512, \ldots, 575$ on $\mathcal{S}$ after p_L. We then perform the divide-and-conquer approach [ZZDX19] to solve multiple models separately. We similarly limit that the number of active words at the beginning of the third round is 3 or less. The

Table 4: 3-Round Differential Trail for Ascon-128a (left) and Ascon-128 (right).

round	Δ before S-box	Δ after S-box
	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000000000000000`
1	`0000000000000000`	`0000000000000000`
	`8008000080000200`	`0000000000000000`
	`8008000080000200`	`8008000080000200`
	`0000000000000000`	`0000100100400400`
	`0000000000000000`	`8148100181400604`
2	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000000000000000`
	`8148100181400604`	`8148000081000204`
	`0480500102412418`	`0c82400100432018`
	`880a10018242a600`	`8488500082018618`
3	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000000000000000`
	`880a10018242a600`	`0000000000000000`

round	Δ before S-box	Δ after S-box
	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000140000401000`
1	`0000000000000000`	`0000000000000000`
	`2140995204509144`	`0000000000000000`
	`2140995204509144`	`21409d5204509144`
	`0000000000000000`	`0000000000000000`
	`0000342002409028`	`0000342000489008`
2	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000000000000000`
	`0000342002489028`	`0000000002080020`
	`0000000000000000`	`40010400120c1020`
	`40010400120c1020`	`0000000000000000`
3	`0000000000000000`	`0000000000000000`
	`0000000000000000`	`0000000000000000`
	`40010400120c1020`	`0000000000000000`

derived 3-round differential trail is given in the left of Table 4, where the output difference after p_L in the third round is `6cb3401180670000` at x_0 and `a1ced308a304a67c` at x_1. Regarding the search time, we limit it to 8 hours for each case in the divide-and-conquer approach. If efficient trails exist, the search is fast. Our trail was detected in 9 minutes, though it did not finish after 8 hours and thus the optimality is unknown.

As pointed out by Liu et al. [LIM20] against the previous 2-round differential trail [ZDW19], assuming the Markov property for the Ascon permutation may result in an invalid trail. Hence, it is important to show the validity of our trail. Here, we adopt the same approach as one proposed by Liu et al. [LIM20] and particularly demonstrated for Ascon in [YLW$^+$23]. Namely, suppose that X and X' are two paired values we want. We first impose linear conditions on the internal state value at the beginning of each round so that input bits to the active S-box can satisfy the desired differential transition. Then we use an automated tool to search for internal values of X satisfying all the conditions.

Deriving Conditions. We denote the j-th bit of x_i from the left by $x_i[j]$. Recall that 5 input bits of an S-box in column j are $x_0[j], x_1[j], x_2[j], x_3[j], x_4[j]$. Visual representation of the conditions along with the trail is available in Sect. C in Supplementary Material.

In the first round, 64 bits of x_0 are fixed to $IV_{k,r,a,b}$, which directly defines the conditions on $x_0[j]$ for all j. There are four active S-boxes where the input difference `3` is transferred to `1` in columns 0, 12, 32, and 54. The corresponding paired input values are $\{$`0c`, `0f`$\}$ and $\{$`1c`, `1f`$\}$. Hence, regardless of the fixed MSB of $x_0[j]$, we obtain three linear conditions; $x_1[j] = 1$, $x_2[j] = 1$, and $x_3[j] = x_4[j]$ for $j = 0, 12, 32, 54$.

In the second round, we have two transitions; from `1` to `9` (columns 0, 7, 9, 12, 32, 39, 54, 61) and from `1` to `18` (columns 19, 31, 41, 53). The corresponding input values are $\{$`12`, `13`, `16`, `17`$\}$ and $\{$`1a`, `1b`, `1e`, `1f`$\}$, respectively. Hence for the columns with differential transition from `1` to `9`, we have three conditions $x_0[j] = 1$, $x_1[j] = 0$, $x_3[j] = 1$, and for the columns with `1` to `18`, we have three conditions $x_0[j] = 1$, $x_1[j] = 1$, $x_3[j] = 1$.

In the third round, the input difference of active S-boxes are either `9` (columns 0, 4, 12, 14, 32, 46, 48, 54), `10` (columns 5, 8, 17, 47, 59, 60), or `19` (columns 19, 31, 38, 41, 50, 53). To be canceled with the difference in the next AD block, the output difference can only exist in the first two rows. For the input difference `10`, the output difference must be `18`, which occurs with probability of 2^{-2} and the conditions are $x_1[j] = 0$, and $x_3[j] \neq x_4[j]$. For the input difference `9`, the output difference can be `8` or `10`, where each occurs with probability of 2^{-4} and thus the sum is 2^{-3}. The corresponding conditions are $x_0[j] = 0$, $x_3[j] = 0$, and $x_1[j] \neq x_4[j]$. For the input difference `19`, the output difference can be `8` or `10`, where each occurs with probability of 2^{-3} and thus the sum is 2^{-2} with the conditions $x_3[j] = 0$ and $x_0[j] = x_4[j]$. Note that if the state value is chosen randomly, the probability of the transition in the third round is $(2^{-2})^{12} \times (2^{-3})^8 = 2^{-48}$.

Complexity analysis. As Liu et al. [LIM20], we modeled the value propagation and the above conditions in MILP. The S-box is viewed as a 32×32 table in which only 32 entries specified by the S-box are possible transitions. This table can be modeled in the same way as a DDT. The XOR operation is the same between values and differences, so p_C and p_L can be modeled easily. With Gurobi Optimizer provided by NEOS Server [CMM98, Dol01, ZZDX19], we obtained the input value satisfying all the conditions in 25 minutes. The generated paired values are shown in Table 6 in Sect. E.

Evaluating the attack complexity is hard because the search procedure is based on the MILP solving algorithm, and is not a simple iterative test by randomly choosing values. The conditions for the first round S-boxes can be satisfied with complexity of 1. Then, we assume that the bits of inactive S-boxes in the first round can be used as the source to satisfy the second round conditions bit by bit, and the conditions on the third round is satisfied only randomly. With this assumption, the attack complexity is 2^{48}. Given that we found a solution only in 25 minutes, the actual complexity should be much smaller. However we refer to this complexity to provide certain convenience for future reference.

3.5 Practical 3-Round CMT-3 Attack on Ascon-128

Search Strategy. Finding an efficient trail for Ascon-128 is much harder than Ascon-128a due to the smaller rate; i.e. rate of Ascon-128 is 64 bits, thus the output difference must be only in the top row. Ascon's S-box requires at least 2 active bits to produce the output difference `10`. Namely, as a result of p_L in the second round, each active S-box in the third round must have active bits in multiple rows, which makes it hard to keep the differential trail sparse. Due to this strong constraints, we fix the number of active rows during p_L in the second round (and the beginning of the third round) to 2. On the other hand, active S-boxes in the third round must produce the output difference `10`, and from DDT, the possible input differences are `03`, `09`, `0c`, `0d`, `12`, `16`, `19`, `1c`, and `1d`. By combining them, the choices of differences at the beginning of the third round are reduced to 4; `03`, `09`, `0c`, and `12`, which are then searched by a divide-and-conquer approach.

The discovered 3-round differential trail is given in the right of Table 4 and Sect. D, where the output difference after p_L in the third round is `e2c40e04329c5260` at x_0. The set of the search time is the same as one for Ascon-128a. After choosing the right case of the divide-and-conquer approach, our trail wad detected in 66 minutes.

Deriving Conditions. In the first round, 64 bits of x_0 are fixed to $IV_{k,r,a,b}$. There are 15 active S-boxes where the input difference `3` is transferred to `1` (columns 2, 7, 9, 16, 20, 23, 25, 27, 30, 37, 43, 48, 55, 57, 61). Conditions for those columns are the same as explained in Ascon-128a, thus we omit them. We also have three columns whose differences are transferred from `3` to `9` (columns 19, 41, 51) and one column from `1` to `9` (column 21). The paired input values for `3` to `9` are $\{\texttt{08}, \texttt{0b}\}$ with 0 at $x_0[j]$. Hence, we obtain three linear conditions; $x_1[j] = 1$, $x_2[j] = 0$, and $x_3[j] = x_4[j]$ for $j = 19, 41, 51$. The paired input values for `1` to `9` are $\{\texttt{12}, \texttt{13}\}$ to have 0 in the MSB of the output, which is required to solve the contradiction explained in the next paragraph. Hence, we obtain three linear conditions; $x_1[j] = 0$, $x_2[j] = 0$, and $x_3[j] = 1$ for $j = 21$.

In the second round, we have three differential transitions; from `9` to `8` (columns 18, 19, 21, 26, 41, 48, 51, 60), from `9` to `1` (columns 38, 58), and from `1` to `9` (column 44). Conditions for `9` to `8` are $x_0[j] = 0$, $x_2[j] = 1$, $x_3[j] = 0$, and $x_1[j] \neq x_4[j]$, for `9` to `1` are $x_0[j] = 0$, $x_2[j] = 0$, $x_3[j] = 0$, and $x_1[j] = x_4[j]$, and for `1` to `9` are as explained in Ascon-128a.

In the third round, we have 9 active S-boxes that transfer the input difference `9` to `10` (columns 1, 15, 21, 35, 38, 44, 45, 51, 58) with probability 2^{-4}. The corresponding conditions for each column is $x_0[j] = 0$, $x_2[j] = 0$, $x_3[j] = 0$, and $x_1[j] \neq x_4[j]$.

No existence of rotated trails. In Ascon, differential trails can generally be rotated because p_S and p_L are symmetric and asymmetric factors $IV_{k,r,a,b}$ and p_C are generally irrelevant to differential propagation. However, the rotated variants of our trail are invalid for any number because positions of '1'-bits in $IV_{k,r,a,b}$ play an important role in our trail.

Invalidity comes from the contradiction of conditions. In our trail, active S-boxes in columns 19 and 21 in the second round impose two conditions $x_0[21] = 0$ and $x_3[19] = 0$ at the input to p_S. On the other hand, active S-boxes in columns 21, 2, and 57 in the first round impose $x_0[21] = 0$, $x_0[2] = 1$, $x_0[57] = 1$ at the output of p_S and this results in $x_0[21] = 0$ after p_L, which matches the condition in the second round. Similarly, active S-boxes in columns 19, 9, and 2 in the first round impose $x_3[19] = 1$, $x_3[9] = 1$, $x_3[2] = 0$ at the output of p_S and this results in $x_3[19] = 0$ after p_L; no contradiction in our trail.

To discuss rotated variants, let the conditions of the second round be $x_0[\ell] = 0$ and $x_3[\ell-2] = 0$, $\ell \in \{0, 1, \ldots, 63\}$. To satisfy $x_0[\ell] = 0$, the conditions for the first round after p_S is $x_0[\ell] \oplus x_0[\ell-19] \oplus x_0[\ell-29] = 0$. Here, $x_0[\ell-19]$ and $x_0[\ell-29]$ are always 0 to satisfy the differential transitions of p_S in the first round. Hence, we have a condition $x_0[\ell] = 0$, but this occurs only when the MSB of the S-box input (specified by $IV_{k,r,a,b}$) is 1. To satisfy $x_3[\ell-2] = 0$, the conditions for the first round after p_S is $x_3[\ell-2] \oplus x_3[\ell-12] \oplus x_3[\ell-19] = 0$. This occurs only when the MSB of the S-box input is 1 in one of those columns. Therefore, to avoid the contradiction, we have the necessary condition;

$IV_{k,r,a,b}$ has two bits of 1 in bit positions $(\ell, \ell-2)$, $(\ell, \ell-12)$, or $(\ell, \ell-19)$.

$IV_{k,r,a,b}$ of Ascon-128 is `80400c0600000000`. There is no ℓ having two bits of 1 with distance 2 and distance 19. The only choice is $\ell = 21$ with distance 12, which is the choice of our trail. This concludes that any rotated variant of our trail is invalid.

One may think that by slightly tweaking our trail, conditions on IV can be avoided. In fact, this was our first attempt when we found the contradiction of conditions for $\ell \neq 21$. However, due to the strong constrains that all the third round S-boxes must have multiple active bits, we have not found any of such slightly tweaked variants of our trail so far.

Complexity analysis. Similarly to Ascon-128a, we modeled the value propagation to satisfy the conditions for MILP. With Gurobi on NEOS Server, we obtained the input value satisfying all the conditions in 3 minutes. The generated paired values are shown in Table 7. With the same evaluation as for Ascon-128a, the attack complexity is 2^{36}.

3.6 Remarks on Ascon-80pq

The difference between Ascon-128 and Ascon-80pq is the key size and the size and value of $IV_{k,r,a,b}$. For the **CMT**-3 security, this only gives advantages to the attacker, where the key value can be chosen by the attacker while $IV_{k,r,a,b}$ is a fixed constant. $IV_{k,r,a,b}$ for Ascon-80pq is `a0400c06` that also satisfies "two bits of 1 with the distance of 12 bits." Hence, the 3-round trail of Ascon-128 can also be used for Ascon-80pq.

4 Committing Security of ASCON Mode with Zero Padding

In this section, we prove the committing security of the ASCON mode in the RP model.[2,3]

4.1 Basic Notations

We use the following basic notations.

- Let ε be an empty string and $\emptyset$ an empty set.
- For an integer $i \geq 0$, let $\{0,1\}^i$ be the set of all i-bit strings, $\{0,1\}^0 := \{\varepsilon\}$, and $\{0,1\}^*$ the set of all bit strings.
- Let 0^i be the bit string of i-bit zeros.
- For $X \in \{0,1\}^j$, let $|X| := j$.
- The concatenation of two bit-strings X and Y is written as $X\|Y$ or XY when no confusion is possible.
- Let $\mathbf{1}_c := 0^{c-1}1$, and $\emptyset$ an empty set.
- For integers $0 \leq i, j$, let $[i,j] := \{i, i+1, \ldots, j\}$ and $[j] := [1,j]$. If $i > j$ then $[i,j] := \emptyset$.
- For integers $0 \leq j \leq i$ and $X \in \{0,1\}^i$, let $\mathsf{msb}_j(X)$ (resp. $\mathsf{lsb}_j(X)$) be the most (resp. least) significant j bits of X.
- For $V \in \{0,1\}^n$, let $\mathsf{ExtK}(V) := \mathsf{msb}_k \circ \mathsf{lsb}_{k+\nu}(V)$ (resp. $\mathsf{ExtIV}(V) := \mathsf{msb}_{n-k-\nu}(V)$) be a function that extracts the k (resp. $n-k-\nu$) bits of V corresponding with the key (resp. IV) of the initial state of ASCON.
- For $V \in \{0,1\}^*$, let $\mathsf{Len}_{10^*}(V)$ be the length of the 10^* part in V from the least significant bit, i.e., if $\mathsf{Len}_{10^*}(V) = j$, then $\mathsf{lsb}_j(V) = 10^{j-1}$. If $V = 0^{|V|}$, then $\mathsf{Len}_{10^*}(V) := 0$.

4.2 Ascon with Zero Padding

Let ASCON$[P_1, P_2]$ be ASCON with n-bit permutations P_1 and P_2, and ASCON$[P_1, P_2]$.Enc the encryption algorithm of ASCON$[P_1, P_2]$ that on an input tuple of a key, a nonce, AD, and a plaintext $(K, N, A, M) \in \{0,1\}^k \times \{0,1\}^\nu \times \{0,1\}^* \times \{0,1\}^*$, returns a pair of the ciphertext and the tag $(C, T) \in \{0,1\}^{|M|} \times \{0,1\}^t$. The specification of ASCON is given in Figure 3 (and the full specification of ASCON is also given in Algorithm 2 in Supporting Material F).

The generic attack on ASCON yields the birthday bound for the tag length that is less than the key size. To enhance the committing security, we extend the black-box technique by [ADG+22] that appends zeros to plaintexts. Let z be the length of zeros and ASCON-ZP the ASCON with the zero padding, i.e., for an input tuple (K, N, A, M), the output of the encryption is defined as ASCON-ZP$[P_1, P_2]$.$\mathsf{Enc}(K, N, A, M) :=$ ASCON$[P_1, P_2]$.$\mathsf{Enc}(K, N, A, M\|0^z)$. We prove that ASCON-ZP achieves $\min\left\{\frac{t+z}{2}, \frac{c}{2}, \frac{n+t-k-\nu}{2}\right\}$-bit committing security, assuming $k \geq t$.

[2]An n-bit RP is chosen uniformly at random from a set of all n-bit permutations.

[3]Due to the different number of rounds and round constants, P_1 and P_2 have different outputs for the same input. Assuming independent RPs for different primitives is common in security proofs.

4.3 Definition of Committing Security

Regarding a committing security notion, we use the definitions given in [BH22].

We define functions WiC_i ($i \in \{1,3,4\}$) that on an input tuple (K,N,A,M), returns the first i elements to which a ciphertext is committed: $\mathsf{WiC}_1(K,N,A,M) = K$, $\mathsf{WiC}_3(K,N,A,M) = (K,N,A)$, and $\mathsf{WiC}_4(K,N,A,M) = (K,N,A,M)$.

In the **CMT**-i-security game where $i \in \{1,3,4\}$, the goal of an adversary $\mathbf{A}$, which is a computationally unbounded one, with access to RPs $P_1, P_1^{-1}, P_2, P_2^{-1}$ is to return two distinct input tuples with respect to WiC_i on which the outputs of Ascon-zp$[P_1,P_2]$ are the same. The **CMT**-i-security advantage of an adversary $\mathbf{A}$ for $i \in \{1,3,4\}$ is defined as

$$\begin{aligned}\mathbf{Adv}^{\mathsf{cmt}\text{-}i}_{\textsc{Ascon-zp}[P_1,P_2]}(\mathbf{A}) := \Pr\Big[&(K^\dagger,N^\dagger,A^\dagger,M^\dagger),(K^\ddagger,N^\ddagger,A^\ddagger,M^\ddagger) \leftarrow \mathbf{A}^{P_1,P_1^{-1},P_2,P_2^{-1}} \text{ s.t.}\\ &\Big(\mathsf{WiC}_i(K^\dagger,N^\dagger,A^\dagger,M^\dagger) \neq \mathsf{WiC}_i(K^\ddagger,N^\ddagger,A^\ddagger,M^\ddagger)\Big)\\ &\wedge\Big(\textsc{Ascon-zp}[P_1,P_2].\mathsf{Enc}(K^\dagger,N^\dagger,A^\dagger,M^\dagger)\\ &\qquad = \textsc{Ascon-zp}[P_1,P_2].\mathsf{Enc}(K^\ddagger,N^\ddagger,A^\ddagger,M^\ddagger)\Big)\Big].\end{aligned}$$

We assume that the input-output pairs of RPs to calculate Ascon's outputs with inputs $(K^\dagger,N^\dagger,A^\dagger,M^\dagger)$ and $(K^\ddagger,N^\ddagger,A^\ddagger,M^\ddagger)$ are defined by adversary's queries. This assumption is valid since all the pairs are necessary to check the condition for **CMT**-i-security, and it is natural to count the number of the pairs for calculating the Ascon's outputs in the number of queries by the adversary. The same assumption is used for proving collision resistance of hash functions in the ideal model, e.g., [LS15].

Bellare and Hoang [BH22] proved that **CMT**-4 implies **CMT**-1, and **CMT**-4 and **CMT**-3 are equivalent.

Lemma 1. *For any* **CMT**-4 *adversary* $\mathbf{A}_4$ *making* p *queries, there exists a* **CMT**-3 *adversary* $\mathbf{A}_3$ *making* p *queries such that* $\mathbf{Adv}^{\mathsf{cmt}\text{-}4}_{\textsc{Ascon-zp}[P_1,P_2]}(\mathbf{A}_4) \le \mathbf{Adv}^{\mathsf{cmt}\text{-}3}_{\textsc{Ascon-zp}[P_1,P_2]}(\mathbf{A}_3)$.

4.4 CMT-4-Security Bound

The following theorem shows a **CMT**-4-security bound of Ascon-zp in the RP model. The proof is given in Section 5.[4]

Theorem 1. *Let* P_1 *and* P_2 *be independent RPs. For any* **CMT**-4 *adversary making* p *queries to* P_1, P_1^{-1}, P_2, *or* P_2^{-1}, *we have*

$$\mathbf{Adv}^{\mathsf{cmt}\text{-}4}_{\textsc{Ascon-zp}}(\mathbf{A}) \le \left(\frac{11p^2}{2^c} + \frac{5p^2}{2^{n-\nu}} + \frac{0.5p^2}{2^{t+z}} + \frac{0.5p^2}{2^{n+t-k-\nu}}\right) \cdot \left(1 - \frac{0.5p^2}{2^n}\right)^{-1} .$$

Assuming $p \le 2^{n/2}$, the term $\left(1 - \frac{0.5p^2}{2^n}\right)^{-1}$ is $O(1)$. Then, assuming $k \ge t$, the above bound shows that Ascon-zp is **CMT**-4-secure as long as $p \ll \min\left\{2^{\frac{c}{2}}, 2^{\frac{t+z}{2}}, 2^{\frac{n+t-k-\nu}{2}}\right\}$, ensuring $\min\left\{\frac{c}{2}, \frac{t+z}{2}, \frac{n+t-k-\nu}{2}\right\}$-bit **CMT**-4 security.

5 Proof of Theorem 1

Let $\mathbf{A}$ be an adversary breaking the **CMT**-4-security of Ascon-zp. Let p_1 (resp. p_2) be the number of queries to R_1 or R_1^{-1} (resp. R_2 or R_2^{-1}). Let $p_{1\mathsf{f}}$, $p_{1\mathsf{i}}$, $p_{2\mathsf{f}}$, and $p_{2\mathsf{i}}$ be

[4]Note that the proof is longer than the previous proofs with zero padding [ADG+22]. This is because the proof considers **CMT**-4 security while the previous ones consider weaker **CMT**-1 security.

Algorithm 1 Lazy Sampled RFs R_i for each $i \in [2]$

```
Initialization
1: $\mathcal{T}_{R_i}$ is initialized as an empty set for each $i \in [2]$

Forward query $X$ to $R_i$ for each $i \in [2]$
1: if $\exists\, j \geq 1$ s.t. $(X, Y_1), \ldots, (X, Y_j) \in \mathcal{T}_{R_i} \wedge Y_1, \ldots, Y_j$ are all distinct then
2:     $Y$ is chosen uniformly at random from $\{Y_1, \ldots, Y_j\}$
3: else
4:     $Y$ is chosen uniformly at random from $\{0,1\}^n$
5: end if
6: $\mathcal{T}_{R_i} \leftarrow \mathcal{T}_{R_i} \cup (X, Y)$; return $Y$

Inverse query $Y$ to $R_i^{-1}$ for each $i \in [2]$
1: if $\exists j \geq 1$ s.t. $(X_1, Y), \ldots, (X_j, Y) \in \mathcal{T}_{R_i} \wedge X_1, \ldots, X_j$ are all distinct then
2:     $X$ is chosen uniformly at random from $\{X_1, \ldots, X_j\}$
3: else
4:     $X$ is chosen uniformly at random from $\{0,1\}^n$
5: end if
6: $\mathcal{T}_{R_i} \leftarrow \mathcal{T}_{R_i} \cup (X, Y)$; return $X$
```

respectively the numbers of queries to R_1, R_1^{-1}, R_2, and R_2^{-1}. Hence, $p_1 = p_{1\mathsf{f}} + p_{1\mathsf{i}}$ and $p_2 = p_{2\mathsf{f}} + p_{2\mathsf{i}}$. For query-response pairs $(X, Y), (X', Y')$ of RPs,, if (X, Y) is defined after (resp. before) (X', Y') is defined then the relation is denoted by $(X, Y) \succ (X', Y')$ (resp. $(X, Y) \prec (X', Y')$).

5.1 Replacing Random Permutations with Random Functions

We first replace RPs (P_1, P_2) by lazy sampled random functions (RFs) (R_1, R_2) where for each new query, the output is chosen uniformly at random from $\{0,1\}^n$. The detail procedures are given in Algorithm 1. In this algorithm, $i = 1$ (resp. $i = 2$) for R_1 (resp. R_2). $\mathcal{T}_{R_i}$ is a table that keeps query-response pairs. $\mathcal{T}_{R_1}$ (resp. $\mathcal{T}_{R_2}$) is for R_1 (resp. R_2). For $i \in [2]$, let $\mathcal{T}_{R_i}^{\mathsf{fwd}}$ (resp. $\mathcal{T}_{R_i}^{\mathsf{inv}}$) be the set of query-response pairs in $\mathcal{T}_{R_i}$ that are defined by forward (resp. inverse) queries. For a new forward (resp. inverse) query to R_i (resp. R_i^{-1}), it returns an n-bit random value. Hence, there probabilistically exist pairs whose first or second elements are the same. For a query that is stored in $\mathcal{T}_{R_i}$, it returns the corresponding value. If there exist multiple pairs, the response is chosen uniformly at random from the pairs.

Let Coll be a collision event for R_1 and R_2 where $\exists i \in [2], (X_1, Y_1), (X_2, Y_2) \in \mathcal{T}_{R_i}$ s.t. $(X_1 = X_2 \wedge Y_1 \neq Y_2) \vee (X_1 \neq X_2 \wedge Y_1 = Y_2)$. The RP-RF switch offers the following bound.

$$\Pr[\mathsf{Coll}] \leq \sum_{i \in [p_1]} \frac{i-1}{2^n} + \sum_{i \in [p_2]} \frac{i-1}{2^n} \leq \frac{0.5p_1^2}{2^n} + \frac{0.5p_2^2}{2^n} \leq \frac{0.5p^2}{2^n} \ .$$

Let Win be an event that $\mathbf{A}$ breaks the $\mathbf{CMT}$-4-security of Ascon-zp$[R_1, R_2]$. Since the RFs behave as RPs as long as Coll does not occur, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{cmt\text{-}4}}_{\textsc{Ascon-zp}[P_1,P_2]}(\mathbf{A}) = \Pr[\mathsf{Win} \mid \neg\mathsf{Coll}] &= \frac{\Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}]}{\Pr[\neg\mathsf{Coll}]} = \frac{\Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}]}{1 - \Pr[\mathsf{Coll}]} \\ &\leq \Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}] \cdot \left(1 - \frac{0.5p^2}{2^n}\right)^{-1} \ . \end{aligned} \tag{1}$$

Note that assuming $p \leq 2^{n/2}$, the term $\left(1 - \frac{0.5p^2}{2^n}\right)^{-1}$ is $O(1)$, and the above bound

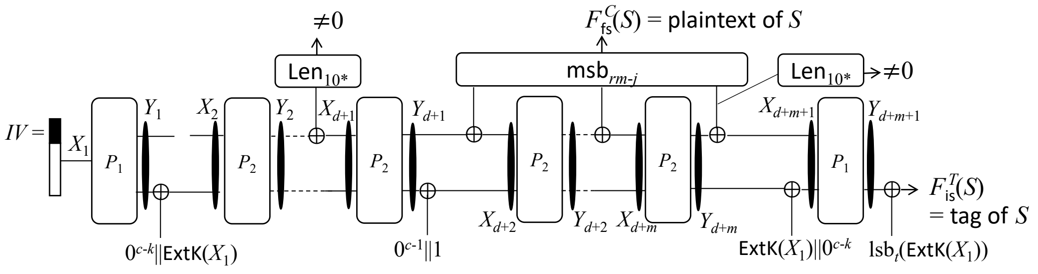

Figure 4: Full Sequence $S = \{(X_1, Y_1), \ldots, (X_{d+m+1}, Y_{d+m+1})\}$.

is dominated by the bound of $\Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}]$. In the following proof, we evaluate the probability $\Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}]$ that is bounded by the probability that Win occurs before Coll occurs.

5.2 Sequences with Ascon's Structure

In this evaluation, we consider sequences of input-output pairs $\{(X_1, Y_1), \ldots, (X_{d+m+1}, Y_{d+m+1})\}$ of R_1 or R_2 that have the structure of Ascon, i.e., for $i \in [d+m+1]$, (X_i, Y_i) is the i-th input-output pair of Ascon. The sequences are formally defined below. The structure of the sequence is depicted in Figure 4.

Definition 1. A set of pairs $\{(X_1, Y_1), \ldots, (X_{d+m+1}, Y_{d+m+1})\}$ is a full sequence if the input-output pairs have the structure of Ascon, i.e.,

1. $(X_1, Y_1) \in \mathcal{T}_{R_1}$, $(X_i, Y_i) \in \mathcal{T}_{R_2}$ for $i \in [2, d+m]$, $(X_{d+m+1}, Y_{d+m+1}) \in \mathcal{T}_{R_1}$,
2. $\mathsf{lsb}_c(Y_1 \oplus X_2) = 0^{c-k}\|\mathsf{ExtK}(X_1)$ if $d \geq 1$;
   $\mathsf{lsb}_c(Y_1 \oplus X_2) = (0^{c-k}\|\mathsf{ExtK}(X_1)) \oplus \mathbf{1}_c$ if $d = 0$ and $m \geq 2$;
   $\mathsf{lsb}_c(Y_1 \oplus X_2) = (0^{c-k}\|\mathsf{ExtK}(X_1)) \oplus (\mathsf{ExtK}(X_1)\|0^{c-k}) \oplus 0^{c-1}1$ if $d = 0$ and $m = 1$,
3. $\mathsf{lsb}_c(Y_i \oplus X_{i+1}) = 0^c$ for $i \in [2, d]$,
4. $\mathsf{Len}_{10^*}(\mathsf{msb}_r(Y_d \oplus X_{d+1})) \neq 0$ if $d \geq 1$,
5. $\mathsf{lsb}_c(Y_{d+1} \oplus X_{d+2}) = \mathbf{1}_c$ if $d \geq 1$ and $m \geq 2$,
   $\mathsf{lsb}_c(Y_{d+1} \oplus X_{d+2}) = (\mathsf{ExtK}(X_1)\|0^{c-k}) \oplus \mathbf{1}_c$ if $d \geq 1$ and $m = 1$,
6. $\mathsf{lsb}_c(Y_i \oplus X_{i+1}) = 0^c$ for $i \in [d+2, d+m-1]$,
7. $\mathsf{lsb}_c(Y_{d+m} \oplus X_{d+m+1}) = \mathsf{ExtK}(X_1)\|0^{c-k}$ if $d \geq 1$ and $m \geq 2$,
8. $\mathsf{Len}_{10^*}(\mathsf{msb}_r(Y_{d+m} \oplus X_{d+m+1})) \neq 0$.

$\mathsf{lsb}_t(Y_{d+m+1}) \oplus \mathsf{ExtK}(X_1)$ is called a "tag of the full sequence". $\mathsf{msb}_{rm-j}\big((Y_{d+1} \oplus X_{d+2})\|(Y_{d+2} \oplus X_{d+3})\| \cdots \|(Y_{d+m} \oplus X_{d+m+1})\big)$ is called a "plaintext of the full sequence" where $j = \mathsf{Len}_{10^*}(\mathsf{msb}_r(Y_{d+m} \oplus X_{d+m+1}))$. $\mathsf{msb}_{rm-j}\big(X_{d+2}\|X_{d+3}\| \cdots \|X_{d+m+1}\big)$ is called a "ciphertext of the full sequence".

A set of pairs $\{(X_1, Y_1), \ldots, (X_{d+m}, Y_{d+m})\}$ is an Ascon's internal sequence if the input-output pairs are of some Ascon's process excluding the last block, i.e., these pairs satisfy the above conditions 1-6 (the last two conditions are excluded). The c-bit value defined by using $\mathsf{lsb}_c(Y_{d+m})$, i.e.,

- $\mathsf{lsb}_c(Y_{d+m}) \oplus \mathsf{ExtK}(X_1)\|0^{c-k}$ if $m \geq 2$;

- $\mathsf{lsb}_c(Y_{d+m}) \oplus (\mathsf{ExtK}(X_1) \| 0^{c-k}) \oplus \mathbf{1}_c$ if $d \geq 1$ and $m = 1$;
- $\mathsf{lsb}_c(Y_{d+m}) \oplus (\mathsf{ExtK}(X_1) \| 0^{c-k}) \oplus (0^{c-k} \| \mathsf{ExtK}(X_1)) \oplus \mathbf{1}_c$ if $d = 0$ and $m = 1$

is called an "output of the internal sequence".

Some definitions for the sequences are given below.

- Let $\mathcal{IS}$ be the set of internal sequences.
- Let $\mathcal{FS}$ be the set of full sequences.
- For $S \in \mathcal{IS}$ (resp. $S \in \mathcal{FS}$), let $S[i] := (X_i, Y_i)$ be the i-th pair of the internal (resp. full) sequence.
- We abuse the notations ExtK and ExtIV. For $S \in \mathcal{FS}$, let $\mathsf{ExtK}(S)$ (resp. $\mathsf{ExtIV}(S)$) be the key (resp. IV) element of S, i.e., $\mathsf{ExtK}(S) = \mathsf{ExtK}(X_1)$ (resp. $\mathsf{ExtIV}(S) = \mathsf{ExtIV}(X_1)$) where $S[1] = (X_1, Y_1)$.
- Let $F_{\mathsf{is}} : \mathcal{IS} \to \{0,1\}^*$ be a function that on an input $S \in \mathcal{IS}$, returns the output of S.
- Let $F_{\mathsf{fs}} : \mathcal{FS} \to \{0,1\}^* \times \{0,1\}^t$ be a function that on an input $S \in \mathcal{FS}$, returns a pair of the ciphertext and the tag of S.
- Let $F_{\mathsf{fs}}^T : \mathcal{FS} \to \{0,1\}^t$ be a function that on an input $S \in \mathcal{FS}$, returns the tag of S.
- For $S \in \mathcal{IS}$ or $S \in \mathcal{FS}$, let $\ell_S := |S|$ be the number of pairs in S.
- For $S \in \mathcal{IS}$ or $S \in \mathcal{FS}$, let $\mathsf{Last}(S) \in [\ell_S]$ be the block number such that $\forall i \in [\ell_S] \backslash \{\mathsf{Last}(S)\} : S[\mathsf{Last}(S)] \succ S[i]$.

5.3 Outline of Remaining Proof

In the remaining proof, we evaluate the probability $\Pr[\mathsf{Win} \wedge \neg \mathsf{Coll}]$.

We first define bad events in Section 5.4. The bad events are defined so that if no bad event occur, for each full sequence $S \in \mathcal{FS}$,

1. $S[1] \prec S[2] \prec \cdots \prec S[\ell_S]$, and
2. $S[2], \ldots, S[\ell_S - 1] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}, S[\ell_S] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$,

and for each pair of distinct full sequences $(S', S^*) \in \mathcal{FS}^2$,

3. the pairs of the tag and the ciphetext are independently defined if $S'[1], S^*[1] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$, and
4. the tags are independently defined if $S'[1] \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$ or $S^*[1] \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$.

The bad events are defined in Section 5.4. To ensure the above condition 1, we define bad events with the case where for some i, $S[i] \succ S[i+1]$. To ensure the above condition 2, we define bad events with the case where for some $i \geq 2$, $S[i] \in \mathcal{T}_{R_1}^{\mathsf{Inv}} \cup \mathcal{T}_{R_2}^{\mathsf{Inv}}$. To ensure the above conditions 3 and 4, we define bad events with the case where for some full sequences, a collision of some inner parts occur. The probabilities for the bad events are upper-bounded in Sections 5.7-5.16 with Lemmas 2 and 3 in Section 5.5.

Assume that no bad event occur. For each pair of distinct full sequences $(S', S^*) \in \mathcal{FS}^2$,

- if $S'[1], S^*[1] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$, then we have $\Pr[F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)] \leq \frac{1}{2^{z+t}}$, and

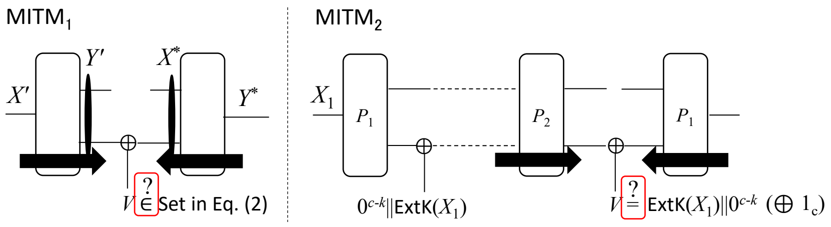
MITM$_1$
X'
Y'
X^*
Y^*
$V \overset{?}{\in}$ Set in Eq. (2)
MITM$_2$
X_1
P_1
P_2
P_1
0^{c-k}||ExtK(X_1)
$V \overset{?}{=}$ ExtK(X_1)||0^{c-k} ($\oplus$ 1_c)

Figure 5: Bad Events MITM_1 (left) and MITM_2 (right).

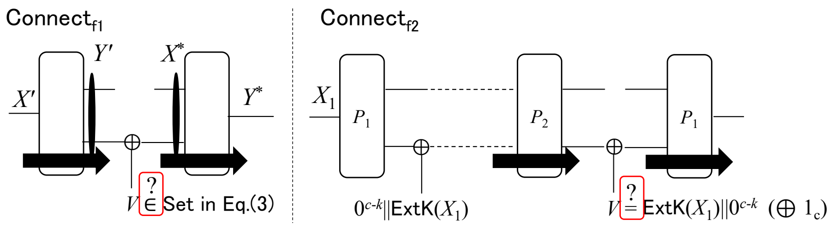
Connect$_{f1}$
X'
Y'
X^*
Y^*
$V \overset{?}{\in}$ Set in Eq.(3)
Connect$_{f2}$
X_1
P_1
P_2
P_1
0^{c-k}||ExtK(X_1)
$V \overset{?}{=}$ ExtK(X_1)||0^{c-k} ($\oplus$ 1_c)

Figure 6: Bad Events $\mathsf{Connect}_{\mathsf{f}1}$ (left) and $\mathsf{Connect}_{\mathsf{f}2}$ (right).

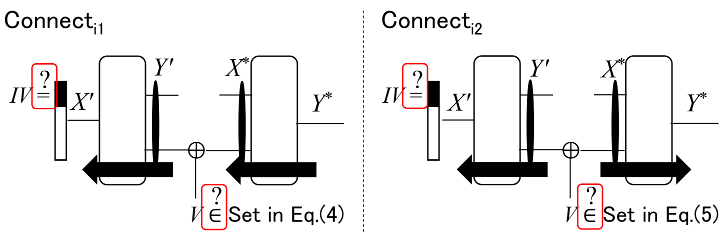

Figure 7: Bad Events $\mathsf{Connect}_{\mathsf{i}1}$ (left) and $\mathsf{Connect}_{\mathsf{i}2}$ (right).

- if $S'[1] \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$ or $S^*[1] \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$, then the response of the inverse query collides with IV, and by the IV collision and the tag collision, we have $\Pr[F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)] \leq \frac{1}{2^{n+t-k-\nu}}$.

Lemma 3 shows that if no bad event occur, then we have $|\mathcal{FS}| \leq p_{1\mathsf{f}}$, and the probability that $\mathbf{A}$ wins (the event is defined as Win) is at most $\binom{p_{1\mathsf{f}}}{2} \cdot \left(\frac{1}{2^{z+t}} + \frac{1}{2^{n+t-k-\nu}}\right) \leq \frac{0.5p_{1\mathsf{f}}^2}{2^{z+t}} + \frac{0.5p_{1\mathsf{f}}^2}{2^{n+t-k-\nu}}$. The evaluation is given in Section 5.17, the first case is defined as an event Win_1 and the second case is defined as an event Win_2.

Using these bounds, we have the upper-bound of $\Pr[\mathsf{Win} \wedge \neg\mathsf{Coll}]$. The bound is given in Eq. (9) in Section 5.6. Putting the bound in Eq. (9) into the one in in Eq. (1), we obtain the bound in Theorem 1.

5.4 Bad Events

The events MITM_1 and MITM_2 depicted in Figure 5 are meet-in-the-middle events for full sequences. MITM_2 considers the last two pairs of full sequences of length ≥ 3. MITM_1 considers the other adjacent pairs of full sequences. The events $\mathsf{Connect}_{\mathsf{f}1}$ and $\mathsf{Connect}_{\mathsf{f}2}$ depicted in Figure 6 are connecting events for full sequences where two forward queries probabilistically connect. $\mathsf{Connect}_{\mathsf{f}2}$ considers the last two pairs of full sequences of length ≥ 3. $\mathsf{Connect}_{\mathsf{f}1}$ considers the other adjacent pairs of full sequences. The event $\mathsf{Connect}_{\mathsf{i}1}$ (resp. $\mathsf{Connect}_{\mathsf{i}2}$) depicted in Figure 7 is a connecting event for the first two pairs of full sequences where two inverse (resp. inverse and forward) queries probabilistically connect. The events Coll_1, Coll_2, and Coll_3 depicted in Figure 8 are collision events for two outputs

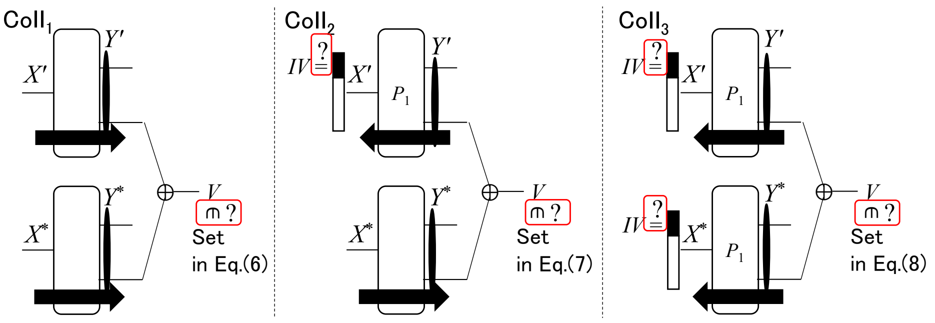

Figure 8: Bad Events Coll_1 (left), Coll_2 (middle), and Coll_3 (right).

except for a collision in outputs of internal sequences whose size is ≥ 2. The event Coll_4 is a collision event for outputs of internal sequences whose size is ≥ 2.

- MITM_1: $\exists (X', Y') \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2} \ \wedge \ (X^*, Y^*) \in \mathcal{T}^{\mathsf{Inv}}_{R_1} \cup \mathcal{T}^{\mathsf{Inv}}_{R_2}$ s.t.

$$\mathsf{lsb}_c(Y' \oplus X^*) \in \big\{0^c,\ \mathbf{1}_c,\ 0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus (\mathsf{ExtK}(X')\|0^{c-k}) \oplus \mathbf{1}_c\big\}. \quad (2)$$

- MITM_2: $\exists S \in \mathcal{FS}$ s.t. $\ell_S \geq 3 \ \wedge \ S[\ell_S - 1] \in \mathcal{T}^{\mathsf{Fwd}}_{R_2} \ \wedge \ S[\ell_S] \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$.

- $\mathsf{Connect}_{\mathsf{f1}}$: $\exists (X', Y'), (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ s.t. $(X', Y') \succ (X^*, Y^*) \wedge$

$$\mathsf{lsb}_c(Y' \oplus X^*) \in \big\{0^c,\ \mathbf{1}_c,\ 0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus (\mathsf{ExtK}(X')\|0^{c-k}) \oplus \mathbf{1}_c\big\}. \quad (3)$$

- $\mathsf{Connect}_{\mathsf{f2}}$: $\exists S \in \mathcal{FS}$ s.t. $\ell_S \geq 3 \ \wedge \ S[\ell_S - 1] \succ S[\ell_S]$
$\wedge \ S[\ell_S - 1] \in \mathcal{T}^{\mathsf{Fwd}}_{R_2} \ \wedge \ S[\ell_S] \in \mathcal{T}^{\mathsf{Fwd}}_{R_1}$.

- $\mathsf{Connect}_{\mathsf{i1}}$: $\exists (X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}, (X^*, Y^*) \in \mathcal{T}^{\mathsf{Inv}}_{R_1} \cup \mathcal{T}^{\mathsf{Inv}}_{R_2}$ s.t. $\mathsf{ExtIV}(X') = IV \ \wedge$

$$\mathsf{lsb}_c(Y' \oplus X^*) \in \big\{0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus (\mathsf{ExtK}(X')\|0^{c-k}) \oplus \mathbf{1}_c\big\}. \quad (4)$$

- $\mathsf{Connect}_{\mathsf{i2}}$: $\exists (X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}, (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ s.t. $\mathsf{ExtIV}(X') = IV \wedge (X', Y') \succ (X^*, Y^*) \ \wedge$

$$\mathsf{lsb}_c(Y' \oplus X^*) \in \big\{0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus (\mathsf{ExtK}(X')\|0^{c-k}) \oplus \mathbf{1}_c\big\}. \quad (5)$$

- Coll_1: $\exists (X', Y'), (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ s.t.

$$\mathsf{lsb}_c(Y' \oplus Y^*) \in \big\{0^c,\ \mathbf{1}_c,\ 0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c \\ 0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*),\ (0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*)) \oplus (\mathsf{ExtK}(X' \oplus X^*)\|0^{c-k})\big\}. \quad (6)$$

- Coll_2: $\exists (X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}, (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ s.t. $\mathsf{ExtIV}(X') = IV \ \wedge$

$$\mathsf{lsb}_c(Y' \oplus Y^*) \in \big\{0^{c-k}\|\mathsf{ExtK}(X'),\ (0^{c-k}\|\mathsf{ExtK}(X')) \oplus \mathbf{1}_c \\ 0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*),\ (0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*) \oplus \mathbf{1}_c, \\ (0^{c-k}\|\mathsf{ExtK}(X' \oplus X^*)) \oplus (\mathsf{ExtK}(X' \oplus X^*)\|0^{c-k})\big\}. \quad (7)$$

- Coll_3: $\exists (X', Y'), (X^*, Y^*) \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$ s.t. $\mathsf{ExtIV}(X') = \mathsf{ExtIV}(X^*) = IV \wedge$

$$\begin{aligned} \mathsf{lsb}_c(Y' \oplus Y^*) \in \{ & 0^{c-k} \| \mathsf{ExtK}(X'),\ (0^{c-k} \| \mathsf{ExtK}(X')) \oplus \mathbf{1}_c \\ & 0^{c-k} \| \mathsf{ExtK}(X' \oplus X^*),\ (0^{c-k} \| \mathsf{ExtK}(X' \oplus X^*) \oplus \mathbf{1}_c, \\ & (0^{c-k} \| \mathsf{ExtK}(X' \oplus X^*)) \oplus (\mathsf{ExtK}(X' \oplus X^*) \| 0^{c-k}) \}. \end{aligned} \tag{8}$$

- Coll_4: $\exists S', S^* \in \mathcal{IS}$ s.t. $S' \neq S^* \wedge \ell_{S'} \geq 2 \wedge F_{\mathsf{is}}(S') = F_{\mathsf{is}}(S^*)$.

Let $\mathbf{Bad} := \{\mathsf{Win}, \mathsf{MITM}_1, \mathsf{MITM}_2, \mathsf{MITM}_3, \mathsf{Connect}_{\mathsf{f1}}, \mathsf{Connect}_{\mathsf{f2}}, \mathsf{Connect}_{\mathsf{i}}, \mathsf{Coll}_1, \mathsf{Coll}_2, \mathsf{Coll}_3, \mathsf{Coll}_4\}$.

5.5 Useful Lemmas for Sequences

The following lemma shows that each of internal or full sequences is defined in the order from the first pair to the last one, and all pairs except for the first pair are defined by forward queries. The proof is given in Subsection 5.18.

Lemma 2. *If the bad events in* **Bad** *do not occur, then*

- $\forall S \in \mathcal{IS} : (\forall i \in [\ell_S - 1] : S[i] \prec S[i+1]) \wedge (\forall i \in [2, \ell_S] : S[i] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}})$, *and*
- $\forall S \in \mathcal{FS} : (\forall i \in [\ell_S - 1] : S[i] \prec S[i+1])$
  $\wedge (\forall i \in [2, \ell_S - 1] : S[i] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}) \wedge (S[\ell_S] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}})$.

The following lemma shows the upper-bounds of the numbers of full sequences and of internal sequences. The proof is given in Subsection 5.19.

Lemma 3. *If the bad events in* **Bad** *do not occur, then we have* $|\mathcal{IS}| \leq p$ *and* $|\mathcal{FS}| \leq p_{1\mathsf{f}}$.

5.6 Upper-Bound of CMT-4-Advantage

For each event $\mathsf{E} \in \mathbf{Bad}$, let E^* be an event that E occurs before the other events in $\mathbf{Bad} \cup \{\mathsf{Coll}\}$ occur. For each $\mathsf{E} \in \mathsf{bad}$, the bound of $\Pr[\mathsf{E}]$ is given in the following sections. Using the bounds (given in Eqs. (10)-(20)), we have

$$\Pr[\mathsf{Win} \wedge \neg \mathsf{Coll}] \leq \sum_{\mathsf{E} \in \mathbf{Bad}} \Pr[\mathsf{E}^*] \leq \frac{11p^2}{2^c} + \frac{5p^2}{2^{n-\nu}} + \frac{0.5p^2}{2^{t+z}} + \frac{0.5p^2}{2^{n+t-k-\nu}}\ . \tag{9}$$

5.7 Upper-Bound of $\Pr[\mathsf{MITM}_1^*]$

For each $(X', Y') \in \mathcal{T}_{R_1}^{\mathsf{Fwd}} \cup \mathcal{T}_{R_2}^{\mathsf{Fwd}}, (X^*, Y^*) \in \mathcal{T}_{R_1}^{\mathsf{Inv}} \cup \mathcal{T}_{R_2}^{\mathsf{Inv}}$, by the randomnesses of Y' and X^*, the probability that the condition in Eq. (2) is satisfied is at most $\frac{5}{2^c}$, giving the following bound.

$$\Pr[\mathsf{MITM}_1^*] \leq \frac{5 p_{\mathsf{f}} p_{\mathsf{i}}}{2^c}\ . \tag{10}$$

5.8 Upper-Bound of $\Pr[\mathsf{MITM}_2^*]$

The event MITM_2 implies that for some $(X^*, Y^*) \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$ and $S \in \mathcal{IS}$ such that $\ell_S \geq 2$, S and X^* probabilistically connect, yielding a full sequence.

Consider $(X^*, Y^*) \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$ and $S \in \mathcal{IS}$ such that $\ell_S \geq 2$. Let $S[\ell_S] := (X', Y')$. By Lemma 3, we have $(X', Y') \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$. By the randomnesses of Y' and X^*, we have $\Pr[\mathsf{lsb}_c(Y' \oplus X^*) \in \{\mathsf{ExtK}(S) \| 0^{c-k}, \mathsf{ExtK}(S) \| 0^{c-k} \oplus \mathbf{1}_c\}] \leq \frac{2}{2^c}$.

By $|\mathcal{T}_{R_1}^{\mathsf{Inv}}| = p_{1\mathsf{i}}$ and $|\mathcal{IS}| \leq p$ (from Lemma 3), we have

$$\Pr[\mathsf{MITM}_2^*] \leq \frac{2 p p_{1\mathsf{i}}}{2^c}\ . \tag{11}$$

5.9 Upper-Bound of $\Pr[\mathsf{Connect}^*_{\mathsf{f1}}]$

For each $(X', Y'), (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ such that $(X', Y') \succ (X^*, Y^*)$, by the randomness of Y', the probability that the condition in Eq. (3) is satisfied is at most $\frac{5}{2^c}$. As $|\mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}| = p_{\mathsf{f}}$, we have

$$\Pr[\mathsf{Connect}^*_{\mathsf{f1}}] \le \sum_{i \in [p_{\mathsf{f}}]} \frac{5i}{2^c} \le \frac{5p_{\mathsf{f}}^2}{2^c} \ . \tag{12}$$

5.10 Upper-Bound of $\Pr[\mathsf{Connect}^*_{\mathsf{f2}}]$

The event $\mathsf{Connect}_{\mathsf{f2}}$ implies that for some $(X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1}$ and $S \in \mathcal{IS}$ such that $\ell_S \ge 2$, S and Y^* probabilistically connect, yielding a full sequence.

Consider $(X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1}$ and $S \in \mathcal{IS}$ such that $S \ge 2$ and $S[\ell_S] \succ (X^*, Y^*)$. Let $S[\ell_S] := (X', Y')$. By Lemma 3, we have $(X', Y') \in \mathcal{T}^{\mathsf{Fwd}}_{R_2}$. By the randomness of Y', we have $\Pr[\mathsf{lsb}_c(Y' \oplus X^*) \in \{\mathsf{ExtK}(S)\|0^{c-k}, \mathsf{ExtK}(S)\|0^{c-k} \oplus \mathbf{1}_c\}] \le \frac{2}{2^c}$.

By $|\mathcal{T}^{\mathsf{Fwd}}_{R_1}| = p_{1\mathsf{f}}$ and $|\mathcal{IS}| \le p$ (from Lemma 3), we have

$$\Pr[\mathsf{Connect}^*_{\mathsf{f2}}] \le \frac{2pp_{1\mathsf{f}}}{2^c} \ . \tag{13}$$

5.11 Upper-Bound of $\Pr[\mathsf{Connect}^*_{\mathsf{i1}}]$

For each $(X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$ and $(X^*, Y^*) \in \mathcal{T}^{\mathsf{Inv}}_{R_1} \cup \mathcal{T}^{\mathsf{Inv}}_{R_2}$, we have $\Pr[\mathsf{ExtIV}(X') = IV] \le \frac{1}{2^{n-k-\nu}}$, and if $(X', Y') \succ (X^*, Y^*)$ (resp. $(X', Y') \prec (X^*, Y^*)$), then the probability that the condition in Eq. (4) is satisfied is at most $\frac{3}{2^k}$ (resp. $\frac{3}{2^c}$). These bounds give

$$\Pr[\mathsf{Connect}^*_{\mathsf{i1}}] \le p_{1\mathsf{i}} \cdot p_{\mathsf{i}} \cdot \frac{1}{2^{n-k-\nu}} \cdot \frac{3}{2^k} = \frac{3p_{1\mathsf{i}}p_{\mathsf{i}}}{2^{n-\nu}} \ . \tag{14}$$

5.12 Upper-Bound of $\Pr[\mathsf{Connect}^*_{\mathsf{i2}}]$

For each $(X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$ and $(X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$ such that $(X', Y') \succ (X^*, Y^*)$, we have $\Pr[\mathsf{ExtIV}(X') = IV] \le \frac{1}{2^{n-k-\nu}}$ and the probability that the condition in Eq (5) is satisfied is at most $\frac{3}{2^k}$. These bounds give

$$\Pr[\mathsf{Connect}^*_{\mathsf{i2}}] \le p_{1\mathsf{i}} \cdot p_{\mathsf{f}} \cdot \frac{1}{2^{n-k-\nu}} \cdot \frac{3}{2^k} = \frac{3p_{1\mathsf{i}}p_{\mathsf{f}}}{2^{n-\nu}} \ . \tag{15}$$

5.13 Upper-Bound of $\Pr[\mathsf{Coll}^*_1]$

For each $(X', Y'), (X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$, the probability that the condition in Eq (6) is satisfied is at most $\frac{7}{2^c}$. The bound gives the following one.

$$\Pr[\mathsf{Coll}^*_1] \le \binom{p_{\mathsf{f}}}{2} \cdot \frac{7}{2^c} \le \frac{3.5p_{\mathsf{f}}^2}{2^c} \ . \tag{16}$$

5.14 Upper-Bound of $\Pr[\mathsf{Coll}^*_2]$

For each $(X', Y') \in \mathcal{T}^{\mathsf{Inv}}_{R_1}$ and $(X^*, Y^*) \in \mathcal{T}^{\mathsf{Fwd}}_{R_1} \cup \mathcal{T}^{\mathsf{Fwd}}_{R_2}$, we have $\Pr[\mathsf{ExtIV}(X') = IV] \le \frac{1}{2^{n-k-\nu}}$ and the probability that the condition in Eq. (7) is satisfied is at most $\frac{5}{2^k}$. These bounds give

$$\Pr[\mathsf{Coll}^*_2] \le p_{1\mathsf{i}} \cdot p_{\mathsf{f}} \cdot \frac{1}{2^{n-k-\nu}} \cdot \frac{5}{2^k} = \frac{5p_{1\mathsf{i}}p_{\mathsf{f}}}{2^{n-\nu}} \ . \tag{17}$$

5.15 Upper-Bound of $\Pr[\mathsf{Coll}_3^*]$

For each $(X', Y'), (X^*, Y^*) \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$, we have $\Pr[\mathsf{ExtIV}(X') = \mathsf{ExtIV}(X^*) = IV] \leq \left(\frac{1}{2^{n-k-\nu}}\right)^2$ and the probability that the condition in Eq (8) is satisfied is at most $\frac{5}{2^k}$. Using these bounds, we have

$$\Pr[\mathsf{Coll}_3^*] \leq \binom{p_{1\mathsf{i}}}{2} \cdot \left(\frac{1}{2^{n-k-\nu}}\right)^2 \cdot \frac{5}{2^k} \leq \frac{2.5p_{1\mathsf{i}}^2}{2^{n-\nu}} \ . \tag{18}$$

5.16 Upper-Bound of $\Pr[\mathsf{Coll}_4^*]$

Fix two distinct internal sequences $S', S^* \in \mathcal{IS}$ such that $\ell_{S'} = \ell_{S^*} \geq 2$. Let $S'[\ell_{S'}] := (X', Y')$ and $S^*[\ell_{S^*}] := (X^*, Y^*)$. Lemma 2 ensures that $S'[\ell_{S'}], S^*[\ell_{S^*}] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$. The events Coll_1, Coll_2, and Coll_3 ensure that X' and X^* are distinct. Thus, Y' and Y^* are independently chosen, and we have $\Pr[F_{\mathsf{is}}(S') = F_{\mathsf{is}}(S^*)] \leq \frac{1}{2^c}$. By using Lemma 3 and the bound, we have

$$\Pr[\mathsf{Coll}_4^*] \leq \binom{p}{2} \cdot \frac{1}{2^c} \leq \frac{0.5p^2}{2^c} \ . \tag{19}$$

5.17 Upper-Bound of $\Pr[\mathsf{Win}]$

The event Win implies $\exists S', S^* \in \mathcal{FS}$ s.t. $F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)$. In this evaluation, we consider the following events where $\mathsf{Win} = \mathsf{Win}_1 \vee \mathsf{Win}_2$.

- Win_1: $\exists S', S^* \in \mathcal{FS}$ s.t. $S'[1], S^*[1] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}} \wedge F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)$.
- Win_2: $\exists S', S^* \in \mathcal{FS}$ s.t. $S'[1] \in \mathcal{T}_{R_1}^{\mathsf{Inv}} \wedge F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)$.

We first evaluate $\Pr[\mathsf{Win}_1]$. By Coll_1, Coll_4, and Lemma 2, for each pair of distinct full sequences $S', S^* \in \mathcal{FS}$, we have $\Pr[F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S')] \leq \frac{1}{2^{t+z}}$. By Lemma 3, we have $\Pr[\mathsf{Win}_1] \leq \binom{p_{\mathsf{f}}}{2} \cdot \frac{1}{2^{t+z}} \leq \frac{0.5p_{\mathsf{f}}^2}{2^{t+z}}$.

We next evaluate $\Pr[\mathsf{Win}_2]$. For each $(X, Y) \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$, we have $\Pr[\mathsf{ExtIV}(X)] \leq \frac{1}{2^{n-k-\nu}}$. For each $S', S^* \in \mathcal{FS}$ such that $\ell_{S'} = \ell_{S^*}$, by Coll_2, Coll_3, and Coll_4, we have $\Pr[F_{\mathsf{fs}}^T(S') = F_{\mathsf{fs}}^T(S^*)] \leq \frac{1}{2^t}$. By Lemma 3, we have $\Pr[\mathsf{Win}_2] \leq \binom{p_{\mathsf{f}}}{2} \cdot \frac{1}{2^{n-k-\nu}} \cdot \frac{1}{2^t} \leq \frac{0.5p_{\mathsf{f}}^2}{2^{n+t-k-\nu}}$.

These bounds give

$$\Pr[\mathsf{Win}] \leq \frac{0.5p_{\mathsf{f}}^2}{2^{t+z}} + \frac{0.5p_{\mathsf{f}}^2}{2^{n+t-k-\nu}} \ . \tag{20}$$

5.18 Proof of Lemma 2

Assume that the bad events in **Bad** do not occur.

We consider the first condition of the lemma. For each $S \in \mathcal{IS}$,

- if $S[1] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$, then by MITM_1, $\forall i \in [2, \ell_S] : S[i] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$ is satisfied;
- if $S[1] \in \mathcal{T}_{R_1}^{\mathsf{Inv}}$, then by $\mathsf{Connect}_{\mathsf{i}1}$, $S[2] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$ is satisfied, and by MITM_1, $\forall i \in [3, \ell_S]$: $S[i] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$ is satisfied.

By $\mathsf{Connect}_{\mathsf{f}1}$ and $\mathsf{Connect}_{\mathsf{i}2}$, $\forall i \in [2, \ell_S] : \mathcal{B}_S[i-1] \prec \mathcal{B}_S[i]$ is satisfied. Hence, the first condition of the lemma holds.

We next consider the second condition of the lemma. For each $S \in \mathcal{FS}$, by the above analysis, $\forall i \in [2, \ell_S - 1] : (S[i-1] \prec S[i]) \wedge (S[i] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}})$ is satisfied. By MITM_2, $S[\ell_S] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$ is satisfied, and by $\mathsf{Connect}_{\mathsf{f}2}$, $S[\ell_S - 1] \prec S[\ell_S]$ is satisfied. Hence, the second condition of the lemma holds.

5.19 Proof of Lemma 3

Assume that the bad events in **Bad** do not occur.

For each internal sequence $S \in \mathcal{IS}$, by Lemma 2, $S[\ell_S] \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$ (resp. $S[\ell_S] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}} \cup \mathcal{T}_{R_1}^{\mathsf{Inv}}$) is satisfied if $\ell_S \geq 2$ (resp. $\ell_S = 1$). The collision events Coll_1, Coll_2, and Coll_3 ensure that for each $(X, Y) \in \mathcal{T}_{R_2}^{\mathsf{Fwd}}$, the number of internal sequences whose last pair is (X, Y) is at most 1. Hence, we have $|\mathcal{IS}| \leq p_1 + p_{2\mathsf{f}} \leq p$.

For each full sequence $S \in \mathcal{FS}$, by Lemma 2, $S[\ell_S] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$ is satisfied. For each $(X, Y) \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$, by Coll_4, the number of internal sequences $S \in \mathcal{IS}$ such that $F_{\mathsf{is}}(S) = \mathsf{lsb}_c(X)$ is at most 1. Hence, we have $|\mathcal{FS}| \leq p_{1\mathsf{f}}$.

6 Ascon with Other Zero Paddings

In the security bound (Theorem 1) and the proof in Sections 4 and 5, we considered Ascon-zp with the particular padding scheme that pads M into $M \| 0^z$. Here, we discuss the other padding schemes, including $0^z \| M$ instead of $M \| 0^z$. Let $\pi : \{0,1\}^* \to \{0,1\}^*$ be a length-preserving bijective function, i.e., $\forall X \in \{0,1\}^* : |\pi(X)| = |X|$ and $\forall X, X' \in \{0,1\}^*$ s.t. $X \neq X'$: $\pi(X) \neq \pi(X')$. The function π defines the positions of the padded zeroes. Then, Ascon with the zero padding is defined as Ascon-zp$^*(M) :=$ Ascon$[P_1, P_2].\mathsf{Enc}(K, N, A, \pi(M \| 0^z))$. The proof of Ascon-zp* is the same as that of Ascon-zp. In the proof, the evaluation of the event Win_1 appears in Section 5.17, depends on the length of 0^z, and considers the collision $F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)$ for full sequences $S', S^* \in \mathcal{FS}$ s.t. $S'[1], S^*[1] \in \mathcal{T}_{R_1}^{\mathsf{Fwd}}$. Regarding Ascon-zp*, even with π, the collision $F_{\mathsf{fs}}(S') = F_{\mathsf{fs}}(S^*)$ depends on the length z, and thus the proof with π offers the same bound as Theorem 1. Hence, we have the following theorem for the **CMT**-4 security of Ascon-zp*.

Theorem 2. *Let P_1 and P_2 be independent RPs. For any length-preserving bijective function π and* **CMT**-4 *adversary making p queries to P_1, P_1^{-1}, P_2, or P_2^{-1}, we have*

$$\mathbf{Adv}^{\mathsf{cmt}\text{-}4}_{\textsc{Ascon-zp}^*}(\mathbf{A}) \leq \left(\frac{11p^2}{2^c} + \frac{5p^2}{2^{n-\nu}} + \frac{0.5p^2}{2^{t+z}} + \frac{0.5p^2}{2^{n+t-k-\nu}}\right) \cdot \left(1 - \frac{0.5p^2}{2^n}\right)^{-1} .$$

7 Conclusion

We proposed the new attacks and proof regarding the context-committing, i.e., **CMT**-3 or equivalently **CMT**-4, security of Ascon. The **CMT**-4 attacker's capability of knowing and controlling the secret key enabled new differential trails between a nonce and an AD block that was unavailable in the conventional attack scenarios. With Ascon-128, the new attack improved the number of rounds from 2 to 3 and practically generated a collision pair. Although the number of rounds stayed the same with Ascon-128a, the new attack significantly improved the complexity, achieving practical generation. Although the new attack succeeds one more round than hash collisions, it is limited to 3 out of 12 rounds, and there is a significant margin until the full rounds.

Then, we gave the **CMT**-4 bounds of Ascon with zero padding. Migration to Ascon can solve the problems regarding committing security. Despite the keyed initialization and finalization functions [DEMS21], Ascon still enjoys collision resistance from the sponge construction and achieves improved committing security compared with the conventional BC-based schemes, e.g., GCM and GCM-SIV. Although the original Ascon's committing security is $t/2$ limited by the generic tag-birthday attack, we can improve it by padding zeroes to the original message. More concretely, Ascon with zero padding achieves $\min\left\{\frac{t+z}{2}, \frac{n-\nu}{2}, \frac{c}{2}, \frac{n+t-k-\nu}{2}\right\}$-bit **CMT**-4-security. This bound corresponds to

$\min\left\{64 + \frac{z}{2}, 96\right\}$ bits with Ascon-128 and Ascon-128a and $\min\left\{64 + \frac{z}{2}, 80\right\}$ bits with Ascon-80pq. The bound is tight with Ascon-128. Meanwhile, tightness with Ascon-128a/Ascon-80pq is unclear and open for future research.

Acknowledgments

We thank the anonymous reviewers and the shepherd for their valuable comments.

References

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In *USENIX Security 2022*, pages 3291–3308, 2022.

[BCG+23] Mihir Bellare, John Chan, Paul Grubbs, Viet Tung Hoang, Sanketh Menda, Julia Len, Thomas Ristenpart, and Phillip Rogaway. Ask your cryptographer if context-committing AEAD is right for you. In *Real World Crypto Symposium (RWC) 2023*, 2023.

[BDPV08] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 181–197, 2008.

[BDPV11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In *SAC 2011*, volume 7118 of *LNCS*, pages 320–337, 2011.

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In *EUROCRYPT 2022*, volume 13276 of *LNCS*, pages 845–875, 2022.

[Boz23] Andrey Bozhko. Properties of AEAD algorithms. https://www.ietf.org/archive/id/draft-bozhko-cfrg-aead-properties-02.txt, 2023.

[CMM98] Joseph Czyzyk, Michael P. Mesnier, and Jorge J. Moré. The NEOS server. *IEEE Journal on Computational Science and Engineering*, 5(3):68–75, 1998.

[CR22] John Chan and Phillip Rogaway. On committing authenticated-encryption. In *ESORICS 2022*, volume 13555 of *LNCS*, pages 275–294, 2022.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In *CT-RSA 2015*, 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.

[DGRW18] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In *CRYPTO 2018*, volume 10991 of *LNCS*, pages 155–186, 2018.

[Dol01] Elizabeth D. Dolan. The NEOS server 4.0 administrative guide. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory, 2001.

[Dwo07] Morris Dworkin. NIST Special Publication 800-38C: Recommendation for block cipher modes of operation: the CCM mode for authentication and confidentiality. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf, 2007.

[FOR17] Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symmetric Cryptol.*, 2017(1):449–473, 2017.

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In *CRYPTO 2017*, volume 10403 of *LNCS*, pages 66–97, 2017.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan. Exploring differential-based distinguishers and forgeries for ASCON. *IACR Trans. Symmetric Cryptol.*, 2021(3):102–136, 2021.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust Authenticated-Encryption AEZ and the Problem That It Solves. In *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 15–44. Springer, 2015.

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

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In *USENIX Security 2021*, pages 195–212, 2021.

[LIM20] Fukang Liu, Takanori Isobe, and Willi Meier. Automatic verification of differential characteristics: Application to reduced Gimli. In *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 219–248, 2020.

[LS15] Jooyoung Lee and Martijn Stam. MJH: a faster alternative to MDC-2. *Des. Codes Cryptogr.*, 76(2):179–205, 2015.

[MLGR23] Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart. Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more. In *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 379–407, 2023.

[MWGP11] Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In *Inscrypt 2011*, volume 7537 of *LNCS*, pages 57–76, 2011.

[NIS23a] NIST. Lightweight cryptography standardization process: NIST selects Ascon. https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon, 2023. Accessed: 2023-08-31.

[NIS23b] NIST. The third NIST workshop on block cipher modes of operation 2023. https://csrc.nist.gov/Events/2023/third-workshop-on-block-cipher-modes-of-operation, 2023. Accessed: 2023-08-31.

[NL18] Yoav Nir and Adam Langley. Chacha20 and poly1305 for IETF protocols. *RFC*, 8439:1–46, 2018.

[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.

[SHW$^+$14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 158–178, 2014.

[ST17] Yu Sasaki and Yosuke Todo. New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In *EUROCRYPT 2017*, volume 10212 of *LNCS*, pages 185–215, 2017.

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

[YLW$^+$23] Xiaorui Yu, Fukang Liu, Gaoli Wang, Siwei Sun, and Willi Meier. A closer look at the S-box: Deeper analysis of round-reduced ASCON-HASH. In *SAC 2023*, 2023. https://sac-workshop.github.io/sac-2023/preproceedings/15XiaoruiYu.pdf.

[ZDW19] Rui Zong, Xiaoyang Dong, and Xiaoyun Wang. Collision attacks on round-reduced Gimli-Hash/Ascon-Xof/Ascon-Hash. Cryptology ePrint Archive, Paper 2019/1115, 2019. https://eprint.iacr.org/2019/1115.

[ZZDX19] Chunning Zhou, Wentao Zhang, Tianyou Ding, and Zejun Xiang. Improving the MILP-based security evaluation algorithm against differential/linear cryptanalysis using a divide-and-conquer approach. *IACR Trans. Symmetric Cryptol.*, 2019(4):438–469, 2019.

A Linear Inequalities To Model DDT and $\mathrm{DDT}^{\mathbf{0}}$

The below is a set of linear inequalities to model DDT of Ascon's S-box without considering the probability of each differential transition. Here we suppose that the S-box input is $X_0\|X_{64}\|X_{128}\|X_{192}\|X_{256}$ and the S-box output is $Y_0\|Y_{64}\|Y_{128}\|Y_{192}\|Y_{256}$.

$$-X_{64} - X_{128} + X_{192} + X_{256} - Y_{128} \geq -2$$
$$X_0 - X_{64} - X_{128} - X_{192} + X_{256} - Y_{64} \geq -3$$
$$X_0 - X_{64} + X_{128} - X_{192} - X_{256} + Y_{192} \geq -2$$
$$X_0 - X_{64} + X_{256} + Y_{128} - Y_{192} - Y_{256} \geq -2$$
$$X_0 - X_{192} + X_{256} - Y_{128} - Y_{192} + Y_{256} \geq -2$$
$$X_{128} + X_{192} - X_{256} + Y_0 + Y_{64} - Y_{128} - Y_{256} \geq -2$$
$$X_{128} + X_{192} - X_{256} - Y_0 + Y_{64} - Y_{128} + Y_{256} \geq -2$$
$$X_{64} + X_{128} + X_{192} + X_{256} - Y_{128} \geq 0$$
$$-X_{64} + X_{128} + X_{192} + X_{256} + Y_{128} \geq 0$$
$$X_0 + X_{192} + X_{256} - Y_{128} + Y_{192} \geq 0$$
$$-X_0 + X_{64} - X_{192} - X_{256} - Y_{256} \geq -3$$
$$X_0 + X_{64} - X_{192} + X_{256} + Y_{256} \geq 0$$
$$X_0 + X_{64} - X_{128} - X_{192} - X_{256} - Y_0 \geq -3$$
$$-X_0 - X_{64} - X_{128} - X_{192} - X_{256} - Y_{64} \geq -5$$
$$X_0 + X_{64} + X_{128} - X_{192} - X_{256} - Y_{192} \geq -2$$
$$X_0 - X_{128} - X_{192} - X_{256} + Y_{64} + Y_{192} \geq -2$$
$$-X_0 + X_{64} + X_{128} + X_{256} - Y_0 - Y_{256} \geq -2$$
$$-X_0 + X_{64} + X_{128} - X_{256} + Y_0 - Y_{256} \geq -2$$
$$-X_{128} + X_{192} - Y_0 + Y_{64} - Y_{128} - Y_{256} \geq -3$$
$$-X_{64} + X_{192} - Y_0 + Y_{64} + Y_{128} - Y_{256} \geq -2$$
$$-X_0 - X_{64} - X_{256} - Y_{128} - Y_{192} - Y_{256} \geq -5$$
$$-X_0 - X_{64} - X_{256} + Y_{128} + Y_{192} - Y_{256} \geq -3$$
$$-X_0 + X_{64} + X_{128} - X_{256} - Y_0 + Y_{256} \geq -2$$
$$-X_{64} - X_{128} + X_{192} + Y_0 + Y_{64} + Y_{256} \geq -1$$
$$X_{64} + X_{192} - Y_0 + Y_{64} + Y_{128} + Y_{256} \geq 0$$
$$-X_0 + X_{192} - X_{256} + Y_{128} - Y_{192} + Y_{256} \geq -2$$
$$-X_0 + X_{192} - X_{256} - Y_{128} + Y_{192} + Y_{256} \geq -2$$
$$-X_{128} + X_{192} - X_{256} + Y_0 - Y_{64} - Y_{128} - Y_{256} \geq -4$$
$$-X_0 - X_{64} - X_{192} + X_{256} + Y_{64} - Y_{192} - Y_{256} \geq -4$$
$$-X_{128} + X_{192} - X_{256} - Y_0 - Y_{64} - Y_{128} + Y_{256} \geq -4$$
$$-X_0 - X_{64} + X_{128} - X_{192} + X_{256} - Y_{64} + Y_{192} - Y_{256} \geq -4$$
$$-X_0 - X_{64} + X_{128} - X_{192} + X_{256} - Y_{64} - Y_{192} + Y_{256} \geq -4$$
$$-X_0 - X_{192} - X_{256} - Y_{128} - Y_{192} \geq -4$$
$$-X_0 - X_{64} - X_{128} - X_{192} + X_{256} + Y_{64} \geq -3$$
$$-X_{64} + X_{192} + Y_0 - Y_{64} + Y_{128} - Y_{256} \geq -2$$
$$-X_{64} + X_{192} - Y_0 - Y_{64} + Y_{128} + Y_{256} \geq -2$$
$$X_{128} + X_{192} - X_{256} - Y_0 - Y_{64} - Y_{128} - Y_{256} \geq -4$$
$$X_{128} + X_{192} - X_{256} + Y_0 - Y_{64} - Y_{128} + Y_{256} \geq -2$$
$$-X_0 - X_{64} - X_{192} + X_{256} + Y_{64} + Y_{192} + Y_{256} \geq -2$$
$$X_{64} - X_{128} + X_{192} + X_{256} + Y_{128} \geq 0$$
$$X_0 + X_{192} + X_{256} + Y_{128} - Y_{192} \geq 0$$
$$-X_0 - X_{192} - X_{256} + Y_{128} + Y_{192} \geq -2$$
$$X_{64} + X_{192} + X_{256} - Y_0 - Y_{256} \geq -1$$
$$X_{64} + X_{192} - X_{256} + Y_0 + Y_{256} \geq 0$$
$$-X_0 + X_{64} - X_{128} - X_{192} + X_{256} + Y_0 \geq -2$$
$$X_0 - X_{64} - X_{128} - X_{192} - X_{256} - Y_{192} \geq -4$$
$$X_0 + X_{64} - X_{128} - X_{192} - X_{256} + Y_{192} \geq -2$$
$$X_0 + X_{64} + X_{128} - X_{256} - Y_0 - Y_{256} \geq -2$$
$$X_0 + X_{64} - X_{128} + X_{192} + Y_0 - Y_{256} \geq -1$$
$$X_0 + X_{64} + X_{128} + X_{256} + Y_0 - Y_{256} \geq 0$$
$$X_{64} + X_{192} - Y_0 - Y_{64} + Y_{128} - Y_{256} \geq -2$$
$$X_{64} + X_{192} + Y_0 + Y_{64} + Y_{128} - Y_{256} \geq 0$$
$$X_0 - X_{64} + X_{256} - Y_{128} + Y_{192} - Y_{256} \geq -2$$
$$-X_0 + X_{64} - X_{128} + X_{192} - Y_0 + Y_{256} \geq -2$$
$$-X_0 + X_{64} + X_{128} + X_{256} + Y_0 + Y_{256} \geq 0$$
$$X_0 + X_{64} + X_{256} - Y_{64} + Y_{128} + Y_{256} \geq 0$$
$$X_{192} - X_{256} + Y_0 + Y_{64} + Y_{128} + Y_{256} \geq 0$$
$$X_0 + X_{64} - X_{192} + Y_0 + Y_{192} + Y_{256} \geq 0$$
$$X_0 - X_{192} + X_{256} + Y_{128} + Y_{192} + Y_{256} \geq 0$$

The below is a set of linear inequalities to model the sparse $\mathrm{DDT}^{\mathbf{b}}$ with $\mathbf{b} = 0$ for the first round, which shows that the number of required linear inequalities is much smaller than that of DDT.

$$-X_0 \geq 0$$
$$-X_{256} \geq 0$$
$$X_{64} - Y_{256} \geq 0$$
$$X_{64} + X_{128} - Y_{128} \geq 0$$
$$X_{64} - Y_0 + Y_{192} \geq 0$$
$$Y_0 - Y_{64} + Y_{192} - Y_{256} \geq -1$$
$$-Y_0 - Y_{64} + Y_{192} + Y_{256} \geq -1$$
$$-X_{192} \geq 0$$
$$Y_{128} - Y_{192} \geq 0$$
$$-X_{64} - X_{128} - Y_{128} \geq -2$$
$$-X_{64} + X_{128} + Y_{192} \geq 0$$
$$X_{64} - Y_{64} + Y_{192} \geq 0$$
$$-Y_0 + Y_{64} + Y_{192} - Y_{256} \geq -1$$
$$-X_{128} + Y_0 + Y_{64} + Y_{192} + Y_{256} \geq 0$$

B Sparse DDT for the First Round in Our Setting

Table 5: Standard DDT (top) and Sparse DDT^0 for the First Round (bottom).

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	32	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	·	·	·	·	·	·	·	·	4	·	4	·	4	·	4	·	·	·	·	·	·	·	·	4	·	4	·	4	·	4	·
2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	4	·	4	·	4	·	4	·	4	·	4	·	4	·	4
3	·	4	·	·	·	4	·	·	·	4	·	·	·	4	·	·	4	·	·	·	4	·	·	·	4	·	·	·	4	·	·	·
4	·	·	·	·	·	·	8	·	·	·	·	·	·	·	8	·	·	·	·	·	·	·	8	·	·	·	·	·	·	·	8	·
5	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	4	·	4	4	·	4	·	4	·	4	·	·	4	·	4
6	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2
7	·	·	4	4	·	·	4	4	·	·	4	4	·	·	4	4	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
8	·	·	·	·	·	·	4	4	·	·	·	·	·	·	4	4	·	·	·	·	·	·	4	4	·	·	·	·	·	·	4	4
9	·	2	·	2	2	·	2	·	2	·	2	·	·	2	·	2	2	·	2	·	·	2	·	2	·	2	·	2	2	·	2	·
a	·	2	2	·	2	·	·	2	·	2	2	·	2	·	·	2	·	2	2	·	2	·	·	2	·	2	2	·	2	·	·	2
b	·	·	2	2	·	·	2	2	·	·	2	2	·	·	2	2	·	·	2	2	·	·	2	2	·	·	2	2	·	·	2	2
c	·	8	·	·	·	·	·	·	8	·	·	·	·	·	·	·	8	·	·	·	·	·	·	·	·	8	·	·	·	·	·	·
d	·	2	·	2	·	2	·	2	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	·	2	·	2	·	2	·	2
e	·	4	4	·	4	·	·	4	·	·	·	·	·	·	·	·	·	4	4	·	4	·	·	4	·	·	·	·	·	·	·	·
f	·	·	·	·	·	·	·	·	4	4	·	·	4	4	·	·	·	·	·	·	·	·	·	·	4	4	·	·	4	4	·	·
10	·	·	·	·	·	·	·	·	·	8	·	8	·	·	·	·	·	·	·	·	·	·	·	·	8	·	8	·	·	·	·	·
11	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	8	·	8	·	8	·	8	·	·	·	·	·	·	·	·
12	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·
13	·	·	8	·	8	·	·	·	·	·	8	·	8	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
14	·	·	·	·	4	4	4	4	·	·	·	·	4	4	4	4	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
15	·	·	·	·	·	4	·	4	·	4	·	4	·	·	·	·	·	4	·	4	·	·	·	·	·	·	·	·	·	4	·	4
16	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
17	·	·	4	·	4	·	·	·	·	·	4	·	4	·	·	·	·	·	4	·	4	·	·	·	·	·	4	·	4	·	·	·
18	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2
19	·	·	·	4	·	·	4	·	4	·	·	·	·	4	·	·	4	·	·	·	·	·	4	·	·	·	·	4	·	·	4	·
1a	·	2	2	·	·	2	2	·	2	·	·	2	2	·	·	2	·	2	2	·	·	2	2	·	2	·	·	2	2	·	·	2
1b	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·
1c	·	4	·	4	·	·	·	·	4	·	4	·	·	·	·	·	4	·	4	·	·	·	·	·	·	4	·	4	·	·	·	·
1d	·	·	·	4	·	4	·	·	4	·	·	·	·	·	4	·	4	·	·	·	·	·	4	·	·	·	·	4	·	4	·	·
1e	·	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2	·	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2
1f	·	·	4	4	4	4	·	·	·	·	·	·	·	·	·	·	·	·	4	4	4	4	·	·	·	·	·	·	·	·	·	·

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	16	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1	·	·	·	·	·	·	·	·	·	·	·	4	·	·	·	4	·	·	·	·	·	·	·	·	·	·	4	·	·	·	4	·
2	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	4	·	·	·	4	·	·	·	4	·	·	·	4
3	·	2	·	·	·	2	·	·	·	2	·	·	·	2	·	·	2	·	·	·	2	·	·	·	2	·	·	·	2	·	·	·
4	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
5	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
6	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
7	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
8	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
9	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
a	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
b	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
c	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
d	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
e	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
10	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
11	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
12	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
13	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
14	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
15	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
16	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
17	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
18	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
19	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1a	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1b	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1c	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1d	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1e	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
1f	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·

C Visual Representation of 3-Round Trail of Ascon-128a

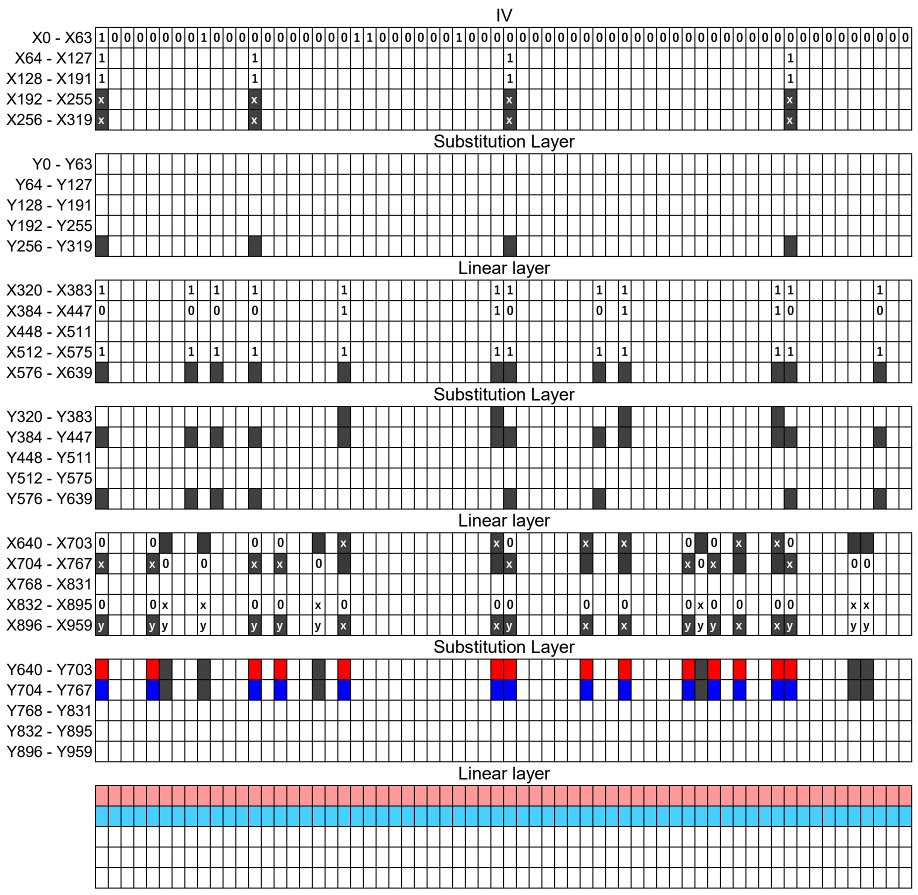

Figure 9: Visual Representation of 3-Round Differential Trail of Ascon-128a and Conditions to Satisfy the Trail. 5×64 cells represent the 320-bit state before and after the substitution layer p_S and the linear layer p_L. The constant addition p_C is omitted. $X_i, Y_i, 0 \leq i \leq 959$ in the left of each state is the variables in MILP. White and black cells represent inactive and active bits, respectively. The red and blue cells in the state after p_S in the third round denote that either one of them can be active because the input differences `09` and `19` to the S-box can propagate to both `10` and `08`. Light red and light blue cells at the output represent that positions of active bits depend on the red and blue cells and thus are not fixed in advance. '1' and '0' in white cells represent the value conditions to be satisfied (by the first value in a pair) to satisfy the proposed differential trail. Two 'x's in a column mean that those values must be identical, while one 'x' and one 'y' in a column mean that those values must be different.

D Visual Representation of 3-Round Trail of Ascon-128

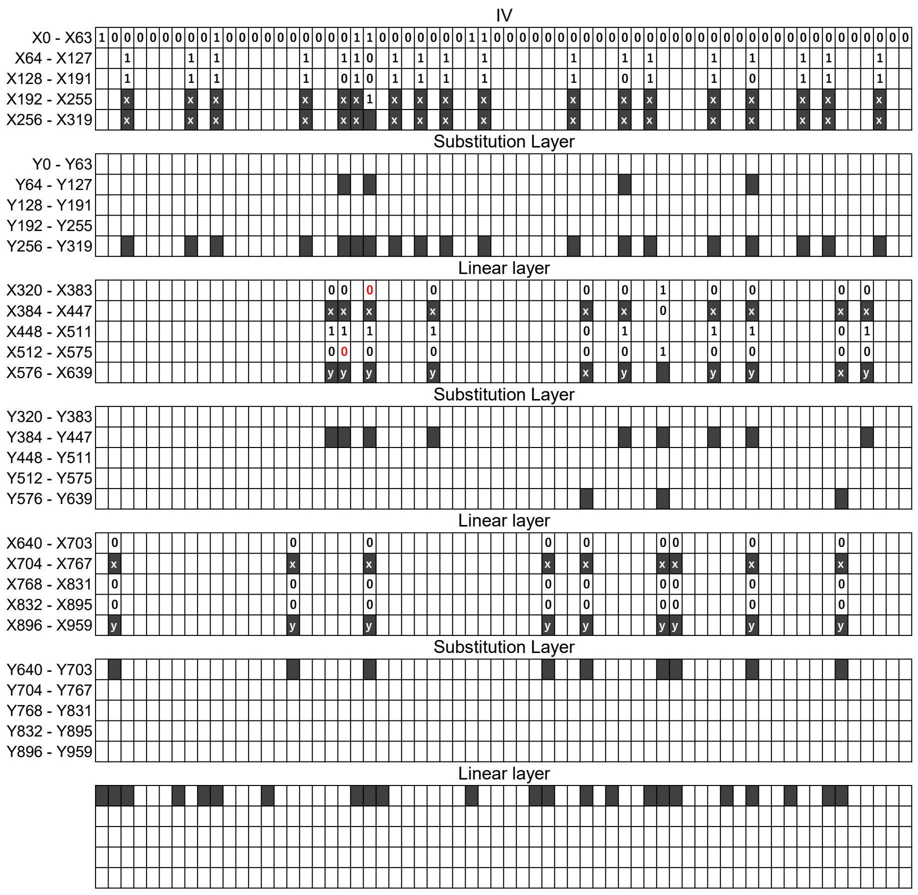

Figure 10: Visual Representation of 3-Round Differential Trail of Ascon-128 and Conditions to Satisfy the Trail. The notations are the same as in Fig. 9. Two '0's in red at the beginning of the second round are the bits we carefully need to avoid the inconsistency by finding two bits of '1' in $IV_{k,r,a,b}$ with 12-bit distance.

E Generated Colliding Paired Values

Table 6: An Example of Paired Values for 3-Round Ascon-128a.

	Value 1	Value 2	Difference
$IV_{k,r,a,b}$	80800c0800000000	80800c0800000000	0000000000000000
K_{MSB}	841b4401c00a0201	841b4401c00a0201	0000000000000000
K_{LSB}	a008c500e0100ef1	a008c500e0100ef1	0000000000000000
N_{MSB}	e300430220100a87	63084302a0100887	8008000080000200
N_{LSB}	a1b30009444a03c2	21bb0009c44a01c2	8008000080000200
	c7583e97e949b62e	c7583e97e949b62e	0000000000000000
	722473090e5c2470	722473090e5c2470	0000000000000000
After 1R	c99dbc778816b42a	c99dbc778816b42a	0000000000000000
	b57b9569a1d43764	b57b9569a1d43764	0000000000000000
	cac4580951f1e0d7	4b8c4808d0b1e6d3	8148100181400604
	02111ba211a96086	06914ba313e8449e	0480500102412418
	a151236a21002862	295b336ba3428e62	880a10018242a600
After 2R	d5984d23b9f8aa83	d5984d23b9f8aa83	0000000000000000
	36d50326453c114e	36d50326453c114e	0000000000000000
	5b7fd322ec0bfbb2	d375c3236e495db2	880a10018242a600
	bcf33658674754ea	d0407649e72054ea	6cb3401180670000
	41bc2ce51cf2671b	e072ffedbff6c167	a1ced308a304a67c
After 3R	501a106b73bace9f	501a106b73bace9f	0000000000000000
(Output)	9c52e4fb0078a9b4	9c52e4fb0078a9b4	0000000000000000
	07741f6d2b0378d2	07741f6d2b0378d2	0000000000000000

Table 7: An Example of Paired Values for 3-Round Ascon-128.

	Value 1	Value 2	Difference
$IV_{k,r,a,b}$	80400c0600000000	80400c0600000000	0000000000000000
K_{MSB}	2164995204d2b154	2164995204d2b154	0000000000000000
K_{LSB}	21408952161a8984	21408952161a8984	0000000000000000
N_{MSB}	8040043400204008	a1009d660470d14c	2140995204509144
N_{LSB}	0470021110020000	25309f4314529144	21409d5204509144
	51e48a98919f2c82	51e48a98919f2c82	0000000000000000
	efbdf90bc9751bbb	efbdcd2bcb358b93	0000342002409028
After 1R	79f1b4b6785bf32f	79f1b4b6785bf32f	0000000000000000
	b261490a843943c3	b261490a843943c3	0000000000000000
	aeb407337089aef5	aeb4331372c13edd	0000342002489028
	9d6061940da22156	9d6061940da22156	0000000000000000
	08d70052ebfab2bb	48d60452f9f6a29b	40010400120c1020
After 2R	ae20f09b6d80208f	ae20f09b6d80208f	0000000000000000
	39aa88b8440203ca	39aa88b8440203ca	0000000000000000
	7cbea6bfd0266b48	3cbfa2bfc22a7b68	40010400120c1020
	a50d1f38a255a0d4	47c9113c90c9f2b4	e2c40e04329c5260
	67cc3c30332574dc	67cc3c30332574dc	0000000000000000
After 3R	f7e64d0ddad70381	f7e64d0ddad70381	0000000000000000
(Output)	ca05427803f501e0	ca05427803f501e0	0000000000000000
	20542b670894ef04	20542b670894ef04	0000000000000000

F Specification of Ascon Mode

We define notations. Let ε be an empty string For integers $i \leq j$ and $X \in \{0,1\}^i$, let $\mathsf{zp}_j(X) := X \| 0^{j-i}$. For an integer $l \geq 0$ and a bit string X, $(X_1, \ldots, X_\ell) \xleftarrow{l} X$ means parsing of X into fixed-length l-bit strings, where if $X \neq \varepsilon$ then $X = X_1 \| \cdots \| X_\ell$, $|X_i| = l$ for $i \in [\ell - 1]$, and $0 < |X_\ell| \leq l$; if $X = \varepsilon$ then $\ell = 1$ and $X_1 = \varepsilon$.

The specification of Ascon is given in Algorithm 2.

Algorithm 2 Ascon

```
Encryption Ascon[$P_1, P_2$].Enc($K, N, A, M$)
1: $S \leftarrow$ Ascon[$P_1, P_2$].ADProcess($K, N, A$)
2: $(M_1, \ldots, M_m) \xleftarrow{r} M \| 1$
3: for $i \in [m]$ do
4:     $S \leftarrow S \oplus \mathsf{zp}_n(M_i)$; $C_i \leftarrow \mathsf{msb}_{|M_i|}(S)$; if $i \neq m$ then $S \leftarrow P_2(S)$ end if
5: end for
6: $C \leftarrow \mathsf{msb}_{|M|}(C_1 \| \ldots \| C_m)$
7: $S \leftarrow P_1(S \oplus 0^r \| K \| 0^{c-k})$; $T \leftarrow \mathsf{lsb}_k(S) \oplus K$; return $(C, T)$

Decryption Ascon[$P_1, P_2$].Dec($K, N, A, C, T'$)
1: $S \leftarrow$ Ascon[$P_1, P_2$].ADProcess($K, N, A$)
2: $(C_1, \ldots, C_m) \xleftarrow{r} C \| 1$
3: for $i \in [m-1]$ do
4:     $M_i \leftarrow C_i \oplus \mathsf{msb}_{|C_i|}(S)$; $S \leftarrow P_2(C_i \| \mathsf{lsb}_c(S))$
5: end for
6: $M_m \leftarrow \mathsf{msb}_{|C_m|-1}(C_m) \oplus \mathsf{msb}_{|C_m|-1}(S)$; $S \leftarrow S \oplus \mathsf{zp}_n(M_m \| 1)$
7: $M \leftarrow M_1 \| \ldots \| M_m$; $S \leftarrow P_1(S \oplus 0^r \| K \| 0^{c-k})$; $T \leftarrow \mathsf{lsb}_k(S) \oplus K$
8: if $T = T^*$ then return $M$; else return $\perp$ end if

Processing AD Ascon[$P_1, P_2$].Hash($K, N, A$)
1: $S \leftarrow IV \| K \| N$; $S \leftarrow P_1(S) \oplus (0^{n-k} \| K)$;
2: if $|A| > 0$ then
3:     $(A_1, \ldots, A_d) \xleftarrow{r} A \| 1$; for $i \in [d]$ do $S \leftarrow P_2(S \oplus \mathsf{zp}_r(A_i))$ end for
4: end if
5: $S \leftarrow S \oplus 0^{n-1}1$; return $S$
```