Compilation of Public Comments on SP 800-232

Public Comment Period: November 8, 2024 – February 7, 2025

On November 8, 2024, NIST published the initial public draft of SP 800–232 Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions. This document compiles the received public comments.

Comments

1. Niux_dannyniu@icloud.com, November 10, 2024

I realize that proliferation of algorithm may be undesirable, but nonetheless I hope NIST can standardize a MAC/PRF based on Ascon.

Also, since the Ascon-Sign team's submission didn't advance into 2nd round, I would hope NIST consider taking a lead in approving Ascon as a parameter family usable with SLH-DSA.

2. Weimer Friedrich, November 13, 2024

Dear NIST Team,

regarding comment period of the new Ascon standard, I'd like to request the specification of a deterministic MAC construction based on Ascon.

In the automotive domain we have use cases, where the authenticity & integrity of messages must be ensured on resource constrained networks and control units.

Specifically, this is the case for CAN bus systems, where communication nodes are resource constrained and frames are also quite constrained in their size.

For these applications, a deterministic – i.e. not relying on the correct usage of a nonce – MAC would be very handy.

Do you have any plans for extending the standard in this direction?

Mit freundlichen Grüßen / Best regards

Dr. Friedrich Wiemer

3. Tom Broström, November 18, 2024

"PRF" appears in Table 1, but nowhere else in the document. Should this document acknowledge either/both hash and xof as suitable PRFs? Maybe that does not quite fit into the document's purpose, but it would also be great if (somewhere, sometime) you show how an Ascon-based PRF can be used in a KDF.

Thanks, -- Tom

4. Joachim Strömbergson, November 21, 2024

Hi,

I don’t see it mentioned in the public draft, but are there any example vectors for Ascon, similar to what is available for AES (fips 197)?

BR

Joachim Strömbergson

Assured AB

5. Bobby McGee, December 2, 2024

The whole point of lightweight crypto and ASCON in particular is to compactly reuse resources, especially the ASCON permutation. NIST should use the same rate for all of AEAD/HASH/XOF so as to easily reuse the same hardware for the permutation across all modes. (Yes, FPGAs and ASICs exist.) Why is NIST mixing the parameter sets from the ASCON submission?

6. Vadím Sukhomlínov, December 2, 2024

Dear Sir/Madam,

Thank you very much for this draft! It is a long awaited document.

I have several comments:

1. Encryption/Decryption are currently defined only as the AEAD-Ascon128 algorithm, which processes plain-text in a CTR block mode. In the extreme case with no additional data and no tag needed it can be reduced to something like Ascon-CTR. But at the same time CTR mode has issues with error propagation only to specific bits modified in ciphertext. Other block cipher modes have more favorable properties. Do you plan to define ECB and other modes built on it? Are NIST SP 800-38* standards applicable to Ascon? One of the applications would be key-wrapping mode and guidance on using Ascon-AEAD128 to encrypt key material.

2. The document defines Ascon-XOF and Ascon-HASH, but doesn't state those can be used in PRFs. In fact, the term PRF is only defined in Table 1, but not used. Defining variants of Ascon as PRF would enable use cases for KDF using constructs defined in NIST SP 800-108. With current draft key derivation with Ascon seems not possible.

3. Use of Ascon for DRBG implementation is not clear. Can Ascon be used instead of AES for AES_CTR_DRBG? Can Ascon-Hash be used for HASH_DRBG? This is another valuable application of Ascon on resource-constrained systems.

4. Use of Ascon for Keyed Message Authentication codes is not covered. While it seems that Ascon-AEAD128's tag can be used for this purpose by dropping ciphertext output, there might be more convenient ways to implement it on top of Ascon-Hash or Ascon-XOF, e.g. by reusing HMAC construction or something simpler, like Ascon-HASH(Key || Message).

Thanks in advance,

Vadim

7. Charlotte Lefevre and Bart Mennink, December 6, 2024

Dear all,

It is nice to see that the draft Ascon standard got out!

Independently, Charlotte and I have been working on a systemization of knowledge on the generic security of the Ascon modes. The work considers the authenticated encryption and hashing modes, as well as the mode underlying Ascon-PRF. The work appeared on ePrint today [1].

Relatedly, note that the draft standard mentions in Table 9 that the Ascon-Hash256 mode guarantees 128 bits of preimage resistance, and that Ascon-XOF128 and Ascon-CXOF128 guarantee min(L,128) bits of preimage resistance. As a matter of fact, generically, these modes achieve 192 and min(L,192) bits of generic preimage resistance, in light of [2]. See also Section 8 of our SoK [1].

[1] Charlotte Lefevre, Bart Mennink: SoK: Security of the Ascon Modes. Cryptology ePrint Archive, Report 2024/1969, 2024. https://eprint.iacr.org/2024/1969

[2] Charlotte Lefevre, Bart Mennink: Tight Preimage Resistance of the Sponge Construction. CRYPTO (4) 2022: 185-204. https://doi.org/10.1007/978-3-031-15985-5_7

Kind regards,

Charlotte Lefevre and Bart Mennink.

8. Clint Powell, December 15, 2024

To Whom it May Concern,

Attached please find the Task Group 802.15.4ae (of the IEEE 802.15 Work Group on Wireless Specialty Networks) comments on the NIST SP 800-232 initial public draft. If you have questions please do note hesitate in reaching out to us.

Best Regards,

Clint

IEEE P802.15

Wireless Specialty Networks

Project	IEEE P802.15 Working Group for Wireless Specialty Networks (WSNs)
Title	**TG 802.15.4ae comments to NIST SP 800-232 initial public draft**
Date Submitted	11th November 2024
Source	Tero Kivinen E-Mail: kivinen@iki.fi
Abstract	Comments relating to the NIST SP 800-232 initial public draft related to the ASCON Tag generation requirement R3
Purpose	Propose changes to NIST SP 800-232 initial public draft so that it can be used with current IEEE Std 802.15.4.
Notice	
Release	

IEEE 802.15 WG comments to the NIST SP 800-232 initial public draft

This document solely represents the views of the IEEE 802.15 Working Group and does not necessarily represent a position of either the IEEE 802 LMSC, IEEE, or the IEEE Standards Association.

NIST SP 800-232 initial public draft in section 4.3 has requirement R3 as follows:

R3. Minimum length of truncated tag. When an application uses truncated tags, the bit length of the truncated tags shall be at least 64 bits, and the tag length shall be the same across the life-span of the key.

Comments:

1) In the IEEE Std 802.15.4™ there are three possible tag lengths MIC-32, MIC-64, and MIC-128. The current text in R3 does not allow using 32-bit tag lengths.

2) IEEE Std 802.15.4™ allows using different tag lengths for the same key, and encodes the tag length inside the Nonce, so each tag length uses a different Nonce value. The current text in R3 does not allow this construct to be used.

Background:

Nonce in IEEE Std 802.15.4 is generated as follows:

Octets: 8	**4**	**1**
Source Address	Frame Counter	Nonce Security Level

Where the Nonce Security Level contains the security level of the frame as an integer, which specifies the tag length to be used:

Security level	**Security attributes**	**Data confidentiality**	**Data authenticity**	**MIC length (octets)**
0	None	OFF	NO	0
1	MIC-32	OFF	YES	4
2	MIC-64	OFF	YES	8
3	MIC-128	OFF	YES	16
4	Reserved			
5	ENC-MIC-32	ON	YES	4
6	ENC-MIC-64	ON	YES	8
7	ENC-MIC-128	ON	YES	16

Request:

To allow future IEEE Std 802.15.4™ implementations to use the NIST approved Ascon cipher, the R3 requirement would need to be changed to allow 32-bit tags, and to allow use of different tag lengths with the same key provided that the nonce for each tag length is generated in such way that the same nonce is never used with two different tag lengths.

9. Bryan Queen, February 4, 2025

Hello,

Thank you for providing us the opportunity to comment on NIST Draft SP 800-232 Ascon Based Lightweight Crypt. Below/Attached please find a summary of comments collected and provided by the Center for Cybersecurity Standards at NSA. The summary represents comments from several SMEs within NSA. Should you have any questions, please contact Mike Boyle (cc'ed) or myself.

Page #	Starting Line #	Ending Line #	Section #	Comment/Rationale
	8	8		It's a little confusing that the second argument of the "parse" function is called "rate", and similarly for the pad function. These are general utility functions, but the rate terminology seems specific to use with a cryptographic sponge.
	9	9		Line 308, "represents" should be "represent".
	10	10		In Table 5, it might be good to explain how the constants are generated. If there's a simple mechanism, a small implementation might benefit from generating these on the fly rather than storing them.
	13	13		In Algorithm 3, I found it difficult to infer how the state is being initialized. That may be detailed in the appendix and, if so, it would be good to point to that here. If no, I think some clarifying language would be appropriate.
	16	16		Same comment for Algorithm 4.
	24	24		Same observation. Clarifying the initialization or pointing to where in the document it is described in detail would be helpful.
	25	25		Line 594, it looks like the IV differs in two bits rather than one, unless I'm mistaken.

10. Nicolas Thaddee Courtois, February 5, 2025

[attachment begins on next page]

Some Observations About the Ascon and Keccak S-box and Potential Applications in Cryptanalysis

Nicolas T. Courtois, Frédéric Amiel, Roberto Avanzi, Marco Macchetti, Alexandre Bonnard de Fonvillars, and William Whyte

Qualcomm France S.A.R.L., Qualcomm Germany GmbH, Qualcomm Wireless GmbH, Qualcomm Technologies, Inc.

Abstract. We report recent findings on linear, differential, rotational, algebraic, space partitioning, and translation self-similarity properties of the Ascon S-box. Many of these findings are also applicable to all the affine equivalents of this S-box, including the Keccak S-box, and several results extend to all quadratic S-boxes. While we do not claim these results can break Ascon or Keccak specifically, they are expected to be cryptographically relevant in the general cryptanalysis of ciphers using similar components. We urge NIST to consider these properties as potentially undesirable.

Key Words. Ascon, Keccak, AES, LAT, LAS, simultaneous linear approximations, Canteaut-Boura restricted linearity, Dobbertin-Charpin normality, DDT, APN, Markov ciphers, Mutual Information, quadratic S-box, connectivity tables, Shift Invariance (SI), space partitioning attacks, differential-linear attack.

1 Introduction

This paper provides an overview of certain remarkable properties of two cryptographic components used in modern NIST-approved applied cryptography: Keccak and, more recently, Ascon. We find structural anomalies and potentially undesirable properties. We do not claim any direct application in cryptanalysis or it is too early to report such results.

A key observation is that any nonlinear S-box, and furthermore any larger cryptographic function, can be linearized to some extent, i.e. for well chosen sets of remarkable states. There are four main approaches to nonlinearity, as discussed in [BoFiPe13]. A new information-theoretic point of view on nonlinearity is the notion of DMI (Differential Mutual information), proposed at the NIST workshop on Ascon in 2023 cf. [CoNi23]. What could potentially go wrong with S-boxes with high DMI values? A critical question is the existence of large simultaneous linear approximations [BCQ04,CoDe08], i.e. involving all the outputs. Several papers have studied the notion of Linearizable Affine Spaces (LAS) in the context of Keccak and Ascon. The LAS-2 properties were first introduced at Eurocrypt 2017 cf. [QSMG17, Section 4.2] in an attempt to construct collisions on up to six rounds of Keccak. On some subsets of size four, the Ascon/Keccak S-Box acts like a linear map. More recently researchers [CoAmFo24] shows that these properties can be generalized and extended to substantially larger sets of

up to eleven elements. To achieve this, we must drop the requirement of the points forming an affine space and study instead arbitrary sets, called Linearizable Sub-Sets (LSS). We demonstrate that LSS properties remain stable under input-side translations for all quadratic permutations. Additionally, we show that several LSS approximations can coexist, covering the entire space, and in fact whole configurations of LSS or LAS properties are also stable under input-side translations. We also explore additional properties, such as combinations with differential properties of maximum strength or how certain bits could be fixed with a large probability. In Ascon and Keccak, LSS properties exhibit strong rotational and translation symmetries and numerous identities cf. [CoAmFo24] and Section A.1.

Affine space to affine space mappings (LAS type) and their potential applications such as vector subspace invariant attacks have been extensively researched in [TeDi19,GJNQSS16,CARG19,CoQi20] to name only very few. At this level it seems relatively easy to avoid major cryptanalytic weaknesses. As far as we are aware, no S-box ever proposed in applied cryptography maps an affine space of dimension 3 to another space of dimension 3. The remaining challenge is the study of probabilistic variants, numerous LAS-2 properties of size four, and small DDT sets exceeding size four, which exist but are very few. In our work we generalize linearization properties to arbitrary sets of points. This generalizes the Canteaut-Boura partial linearity [BoCa14], cf. also [CoAmFo24] or Section 5.2.3. in [GJNQSS16]. It also generalizes the Dobbertin-Charpin notion of weak k-normality [ChDo04]. Our new LSS sets are substantially larger than those typically studied in cryptanalysis, comfortably exceeding the size of any DDT set. In a Markov cipher model DDT coefficients are independent of each other Here they "obey" some linearization properties, cf. Thm. 4 in page 19 and Section 10. We show that LSS and DDT properties of maximum strength lead to very closely related space coverings.

Outline. The paper is organized as follows. In Section 2 and 3 we study the key differential and linear properties and how Ascon and Keccak compare to other 5-bit S-boxes in Section 4. In Section 5 we study how some properties fix certain bits with potential applications in cryptanalysis of Ascon/Keccak. In Section 6 we study translation self-similarity properties related to LAT tables and linear cryptanalysis. In Section 7 we present two general theorems about simultaneous linearization LSS properties. In Section 8 and Appendices A.1 A.3. we study some remarkable identities due to [hidden] rotational self-similarity properties which Ascon inherited from Keccak. In Section 9 we show how LSS and DDT properties interact in Ascon leading in particular to maximum-strength space partitions such as 11+11+11+10, and how this is related to a well-known concept of UnDisturbed Bits (UDB). In section 10 we present further precise results connecting differential cryptanalysis with LSS-11 properties. In Section 11 we present our conclusion.

2 Notation and Basic Definitions

We summarize here our notations.

F_2	Finite field $F_2 = \{0, 1\}$
$\oplus$	Bitwise XOR or bitwise addition modulo 2 in F_2^k
S-box	In this paper all S-boxes are assumed to be bijective
$F[], S[]$	Some cryptographic S-box $F_2^k \to F_2^k$
k	size in bits, in this paper $k = 5, 6, 7, 8$ typically
$y = F[x]$	Application of F to x, implies $x, y \in F_2^k$
x_i, y_i	We number bits of x and y from 0 to $k - 1$
I/O	Input and Output sides
MC	Multiplicative Complexity cf. [BoFiPe13,BoPePo00]
LAT	Linear Approximation Table
NL	NonLinearity $2^{k-1} - \lvert max_{\alpha,\beta\neq 0}(LAT(\alpha, \beta))\rvert$
WZ,WN	set of Walsh Zeroes $LAT(\alpha, \beta) = 0$ and its complement
DDT	Differential Distribution Table
δ,max-DDT	the biggest integer in DDT
δ_{in}, δ_{ou}	Two 5-bit I/O differences
DDT-4	Any DDT set of size 4
MI	Mutual Information
DMI	$\text{MI}(\delta_{in}; \delta_{ou})$
LAS-2	Linearizable Affine Subspaces of dim 2
LSS-11	A Linearizable Sub Set of points of size 11
$\sum_{in}(V)$	XORs of all inputs for all $x \in V \subset F_2^k$
$\sum_{ou}(V)$	XORs of all outputs $S[x]$ for all $x \in V \subset F_2^k$
UDB	UnDisturbed Bits property $F[x]_j \oplus F[x \oplus \delta_{in}]_j = \text{Cst}$ for several j
ρ_{in}	Input-side rotation in Ascon, $\rho_{in}[x] = S_{in}^{-1} \circ Rot5 \circ S_{in}$
ρ_{ou}	Output-side rotation in Ascon, $\rho_{ou}[x] = T_{ou} \circ Rot5 \circ T_{ou}^{-1}$
AE	Affine Equivalence with $G = A \circ F \circ B$, where A,B are F_2-affine

2.1 On Mutual Information and DMI

A nice measure of quality of the S-box is to measure the Mutual Information between the input difference δ_{in} and the output difference δ_{ou} across all possible pairs of values. We call this quantity DMI and a simple formula to compute DMI is given below. The idea is that ciphers with high DMI are potentially problematic was first proposed by Nicolas Courtois at a NIST workshop about Ascon in June 2023: cf. [CoNi23]. The study of the DDT() sets, cf. Section 2.2 is motivated by and closely related to Differential Cryptanalysis. Using a standard formula for MI mixing joint and marginal probabilities found in wikipedia [wikiMI] we define:

Definition 1 (DMI). *We define DMI as follows:*

$DMI = MI(\delta_{in}; \delta_{ou})$, done for all possible pairs of inputs, more precisely:

$$\sum_{\forall \delta_{in}, \delta_{ou} \in F_2^k} Pr_x(S[x] \oplus S[x \oplus \delta_{in}] = \delta_{ou}) \cdot log_2\left(\frac{Pr_x(S[x] \oplus S[x \oplus \delta_{in}] = \delta_{ou})}{2^{-k} \cdot Pr_{x,x'}(S[x] \oplus S[x'] = \delta_{ou})}\right)$$

In [CoAmFo24] we also find another more precise definition restricted to arbitrary subsets $S \subseteq F_2^k$ which is not needed here.

In current crypto literature there is plenty of S-boxes with high DMI values. It is possible to see that an S-box, such as in Ascon and Keccak can be considered to be weak just because it is Shift Invariant (SI) cf. [DaPhD95]. All known Shift Invariant (SI) S-boxes have high DMI> 1.35, cf. [DaPhD95] and [GLL19] and hence also plenty of high size LSS and LAT-2 properties. The situation is even worse for S-boxes which can be decomposed into smaller S-boxes cf. [BoPeTi19]. Such S-boxes typically have very high DMI: like close to 2.0 or worse. Best S-boxes such as APN have DMI close to 1.0 and for best APNs which are not bijective it is possible to achieve DMI as low as 0.97429, cf. github.com/lpp-crypto/sboxU/. Similarly the best known result for bijective 8-bit S-boxes is DMI=1.04283.

2.2 The DDT Connection

Let S be an S-box on k bits. We recall the standard notion of DDT sets:

$$DDT(\delta_{in}, \delta_{ou}) = \{x \in F_2^n | S[x] \oplus S[x \oplus \delta_{in}] = \delta_{ou}\}$$

In Observation 1 in [QSMG17] the authors already show that the sets of points of type $DDT(\delta_{in}, \delta_{ou})$ are sometimes LAS of size 4 and are affine spaces, sometimes they are of size 8 and and contain several LAS.

We observed that in Ascon and in Keccak there are exactly 20 sets of type $DDT(\delta_{in}, \delta_{ou})$ which are of size 8 and are "behind" each 8 which appears in the DDT table of Ascon. The set of δ_{in} for which this happens contains 5 elements, and there are 5 lines in the DDT table which contain four 8's each. All the 20 output differences are also disjoint across all the four entries in each of 5 lines. These δ_{in} are also exactly those with the "undisturbed bits" UDB property cf. [TeAs16,TeDi19,MaTe14]. In previous works many authors considered that DDT events are somewhat random and happen in isolation, and could be modeled by a set of independent Poisson random variables [PeRE18], for example in slide 36 in [PeRE18]. In fact there is a typo in slide 36: one needs to replace $2^z z$ by $2^z z!$. In this paper we need to go beyond the traditional "Markovian cipher ideal" modeling where everything is modeled by independent random variables. We claim that this type of modeling fails when we look at larger [or maximum size] sets. We will see that every single line in a DDT table, will be strongly correlated to some [global] large size linear space partitioning properties, cf. Thm. 4. This could explain certain anomalies already observed by Tezcan in differential-linear attacks on Ascon [TeAs16,TeDi19].

2.3 On Interaction Between DDT sets and Linear Properties

In this paper we study how linear and differential properties can be related to each other. The idea of how DDT (Differential Distribution Table) reveals some affine spaces and their transformations is not new, see for example Section 5.2 in [GJNQSS16]. More specifically in ciphers of Ascon/Keccak family, a key result is Observation 1 at Eurocrypt 2017 [QSMG17] and in earlier works. The authors show that properties where an affine space is sent into another affine space in Ascon/Keccak S-box are always identical, or a highly regular subsets, uniformly related to and emanating from, essentially all known DDT sets of sizes 2,4 and 8. In this paper we are going to generalize this type of observation and study stronger and more general S-box linearization properties called LSS. LSS are also stronger variant of Canteaut-Boura notion of linearity of [BoCa14] and closely related to Dobbertin/Charpin notions of normality [ChDo04]. We will see that the set of linear approximation properties of maximum possible size 11 out of 32 is regular and stable by translations, and all 32 best-possible linear properties, and all the 20 best-possible differential properties being DDT sets of size 8, do strongly interact in Ascon and Keccak. More precisely we will show that these pairs of sets are either near-disjoint, or almost identical, see later Thm. 4. They never behave as random sets: medium-size intersections never happen. Many related sets are very regular and obey many remarkable identities: see Section 9 and Table 6 in Section 8 and pages 32-37 in [CoAmFo24] and Section A.1.

3 On [Imperfect/Partial] Linearity of Crypto S-boxes

When linear approximations are studied in isolation, everything seems simple. This topic is not new and was studied since early 1970s, see [CoPoSc08].

3.1 Single Linear Approximations - LAT and WZ

Definition 2 (LAT Table). *Let S be an S-box on k bits. We define the Linear Approximation Table (LAT) as follows:*

$$LAT(\alpha,\beta) = \sum_{x \in F_2^k} (-1)^{\alpha.x+\beta.S[x]}$$

where . is the dot product. We call NL=$\mathcal{L}_S = 2^{k-1} - |max_{\alpha,\beta\neq 0}(LAT(\alpha,\beta))|$.

Definition 3 (WZ). *We call Walsh Zeroes (WZ) all pairs s.t. $LAT(\alpha,\beta) = 0$.*

Similarly, we call the WN (they are Walsh Not-Zeroes) the set of non-zero entries. The study of WZ and WN Every entry in the LAT table leads to partitioning of the 2^k possible states into (twice) two sets of equal size for WZ, with unequal sizes for WN. In theory these partitions could behave just like random sets of points, yet for some S-boxes they are extremely regular and similar (by simple translation) to other sets obtained by the same method, cf. Section 4 in [CoAmFo24]. We revisit these questions in Section 6: we recall the basic

facts and study how these could be relevant in cryptanalysis. A general theory of linearization (for example in Algebraic Attacks) requires a notion of linear I/O equations, cf. [CoDe08,CoAmFo24]. In this paper we only look at functional aspects [simpified view].

3.2 Multiple Linear Approximations - LAS and New Definitions

The topic of cryptanalysis with multiple linear approximations was always considered as very difficult [BCQ04,CoDe08], and remains so even today. One important notion already studied for Ascon and Keccak S-boxes is the notion of LAS introduced at Eurocrypt 2017, cf. [QSMG17].

Definition 4 (LAS = Linearizable Affine Subspace). *Let S be an S-box on k bits. We call LAS or Linearizable Affine Subspace a set of points $V \subseteq F_2^k$ which forms an affine subspace such that*

$$S[x] = A \cdot x + c \quad \forall x \in V$$

where A is $k \times k$ matrix and c is a constant vector in F_2^k.

Here we relax this notion substantially and consider arbitrary sets of points:

Definition 5 (LSS = Linearizable Sub Set). *Let S be an S-box on k bits. We call LSS or Linearizable Sub Set any set of points V such that*

$$S[x] = A \cdot x + c \quad \forall x \in V.$$

Examples: Power Functions in Finite Fields of Even Dimension. In Section 3 of [FaAr08] we discover that a function $x \mapsto x^{37}$ in F_2^{10} has an LSS-18 property where V has 18 elements and will be the set of elements of order 11 in F_2^{10}. For these elements $x^{37} = x^4$ which is a linear function over F_2. They also found one LSS-18 property with the AES S-box.

Multiple [Disjoint] LSS Solutions. It is possible to see that in Ascon there exist two disjoint sets of size 11 covering jointly 22/32 of the whole space [CoAmFo24]. Likewise there is more than one such property in AES. It is possible to see that the power function $x \mapsto x^{254}$ in F_2^8 has 15 disjoint LSS properties of size 18. These sets of 17 form a natural partitioning of $F_2^8 - \{0\}$ with 255=17*15 elements total. We conjecture that these properties of size 18 are optimal and cannot be improved.

Definition 6 (Maximum Size LAS/LSS). *Let S be an S-box on k bits and V be a set of points $V \in F_2^n$. We say that V is maximal size LAS / LSS respectively if*

$$S[x] = A \cdot x + c \quad \forall x \in V.$$

and satisfies one of the previous definitions respectively AND the equality above does no longer hold if we add any additional points to V.

In this paper we claim that multiple large LSS properties exist and they CAN systematically be extended and combined to cover the whole space of all possible inputs. In effect we will work on "two-tier" full linearization properties which allows to model the whole S-box and possibly more than once (with overlaps).

3.3 On Space Covering with Disjoint LSS Approximations

We are now looking at a further type of a "global" linearization property. We would like to be able to approximate a function with several LSS approximations. Moreover, we postulate that the matrices used should live inside a fairly small linear space (for example 2 out of $5^2 = 25$ maximum possible).

Definition 7 (LSS Affine Dimension). *We call the LSS Affine Dimension of the S-box F, the smallest integer D such that there exist matrices $A_1, \ldots, A_D$ and a constant translation matrix A_0, such that the set of all possible 2^D affine combinations of the form, $A_0 \oplus \sum_i A_i$ corresponds to an LSS property for some set V and union of these sets V is the whole input space.*

Lemma: It is obvious that LAS-2 and LSS and the LSS dimension are all stable for ordinary Affine Equivalence (AE) of S-boxes, and also with extended affine equivalence (EA) for cryptographic S-boxes e.g. $G[x] = (A \circ F \circ B)[x] \oplus C[x]$. But not for CCZ equivalence or likewise cf. [BrChMa04].

Toy Example. We found that in Ascon, as will see later, there exist just 5 matrices A_i on 25 bits with corresponding LSS-11 properties which covers the whole space 11 times. We give here two examples of matrices obtained in LSS-11 properties, for A_{11} and $B_{11} = A_{11} \oplus 26$. In the case of $A_{11} = \{0, 3, 4, 12, 16, 17, 19, 20, 21, 29, 31\}$:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{bmatrix} 1\,0\,1\,1\,0 \\ 1\,0\,1\,1\,1 \\ 0\,1\,1\,1\,1 \\ 1\,1\,1\,0\,0 \\ 0\,1\,0\,1\,1 \end{bmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

For example with input $4 = 00100 = x_0x_1x_2x_3x_4$ we get output $26 = 11010 = y_0y_1y_2y_3y_4$ in binary where $x4/y4$ represent the least significant bit.

In the case of $B_{11} = 11, 13, 15, 21, 22, 23, 26, 27, 29, 30, 31$ we have:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{bmatrix} 0\,1\,0\,1\,1 \\ 1\,1\,1\,0\,1 \\ 0\,1\,1\,1\,0 \\ 0\,1\,1\,1\,1 \\ 1\,0\,0\,1\,0 \end{bmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

For example with input $22 = 10110 = x_0x_1x_2x_3x_4$ we get output $17 = 10001 = y_0y_1y_2y_3y_4$ in binary where x_4/y_4 represent the least significant bit. It is possible to see that these matrices and many more are cyclic (circulant) matrices under an appropriate variable change and in Keccak, see Section 8.

Avoiding Overlap. In general it is NOT necessary to cover the whole space so many times, one time (with slight overlap if any) would be enough, hence we expect that the minimum dimension is frequently smaller than k.

Comparison to Other S-boxes. These results should be placed in the context of a well known classification of all quadratic S-boxes, cf. [BoBiSa17].

For example the LSS dimension in Ascon S-box is **only** 2 out of 5. and it is very low also for many other well-known S-boxes cf. Table 1.

Applications in Cryptanalysis. The more the full space if covered by different very large size LSS properties, for example 11+11+11+10 times for Ascon, the more an attacker can select a set of full-size approximations of the whole S-box which might work together inside an attack. Each LSS partitioning is like forcing all input differentials for all pairs which are "similar" in this metric, to have an output difference δ_{ou} which is uniquely and deterministically determined on 5 bits. If this is not the case, Tezcan UDB properties allow to predict 2 or 3 bits of δ_{ou}. After each round with choice of say 4 matrices and 4 LSS approximations, the outputs of this round will live in a union of a small number of affine spaces, and these in turn will limit the choice of LSS approximations to apply in the next round. The number of possibilities to study is enormous.

4 Ascon and Keccak compared to Other of 5-bit S-boxes

We have computed the LSS dimension for all major known 5-bit S-boxes. In the table below we show the results for the 75 classes of quadratic S-boxes from [BoBiSa17]. We also study the 17 "strong" degree 4 S-boxes from Table 7 in [MeBi19], plus few more well-known S-boxes.

ANF deg	class	NL	DMI range	LAS-2 range	LSS range	LSS dim
2	37-52	0	1.96-2.78	104-216	11-13	2
2	35,36	8	2.16	152	13	2
2	53,56-59	8	1.78-2.06	72-104	11	2
2	Ascon,Keccak	8	1.91	80	11	2
2	61-71	8	1.68-2.06	48-120	9-11	2
2	72	8	1.59	40	9	3
2	73	8	1.41	24	8	3
4	Icepole	8	1.81	70	10	3
4	Thakor	8	1.59	45	9	3
4	1-15	10	1.30-1.41	15-25	9-10	3
4	16	10	1.24	10	8	3
4	17	10	1.12	0	9	3
4	Inv-5	10	1.12	0	7	3
2	x^3	12	1.12	0	7	3
2	74/Fides	12	1.12	0	7	3
2	75	12	1.12	0	7	3

Table 1. Results on LSS dimension and closely related parameters for major known permutations on 5 bits.

Each of these LSS dimension computations takes between 1 second and 6 hours on a PC with CryptoMiniSat 5.8. All results are lower than expected. It

may seem difficult to believe that the LSS dimension could be so low in practice for so many S-boxes and in fact for all known S-boxes.

Byond quadratic S-boxes. Some results on LSS and LAT vulnerabilities in 5-bit S-boxes which are not quadratic such as those studied in [MeBi19], can be found in Appendix B in [CoAmFo24].

4.1 On LSS-10 and Smaller Near-Optimal Properties

It is easy to see that the number of LSS$-m$ properties increases as m goes down. In Ascon/Keccak there are exactly 5 classes of **non-trivial** LSS-10 properties see Appendix A.3. These properties are freely translatable by translations and rotations (while the LSS-11 are only stable w.r.t. translations).

4.2 Differential Transparency and Rotation Symmetries

As far as we can see, all known Shift Invariant (SI) S-boxes have average to high DMI>1.35, cf. [DaPhD95] and [GLL19] and hence also high LSS sizes. In contrast bijective APN S-boxes have very low DMI ≈ 1.0 typically, and hence also very low maximum possible LSS sizes which are frequently below $2 \cdot k$ for a permutation on k bits (for example 7 instead of $2 \cdot 5 = 10$ cf. Table 1).

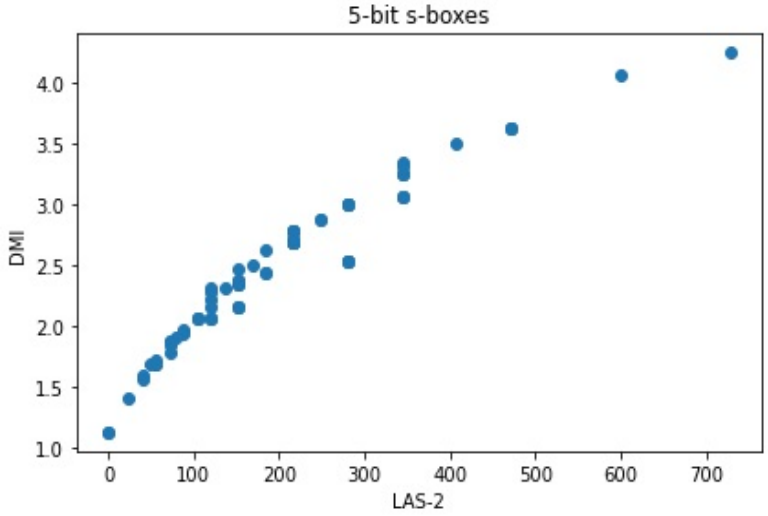

Fig. 1. DMI and LAS-2 count for all 75 affine classes of quadratic permutations on 5 bits following Table 1 in [BoBiSa17].

5 On DDT and LSS Sets Which Fix Some Bits

Definition 8 (ioab). *We say that a set of points $V \subset F_2^k$ is ioab for two integers a, b if for every x, y with $y = S[x]$ there are a bits out of 5 fixed on input side and there are b bits out of 5 fixed on the output side.*

Example 1: We consider one of the LAS sets of size 4 already studied in [QSMG17] which is 13,14,29,30↦3,6,10,15 and which has a property which can be summarized as ?11??=>0??1? in binary with 4 bits being fixed. This set is of type io22: two input bits are fixed at input side, and two output bits are fixed at output side, this simultaneously for all 4 points and their transformations $y = S[x]$. In the same way the set 9,10,25,26↦5,8,12,1 has property ?10??=>0??0? and is of type io22 also.

Example 2: Finally the union of the two sets listed above, which is exactly: $\{9, 10, 13, 14, 25, 26, 29, 30\}$ is of type io11. Moreover this set of 8 values is always in the form (in binary):

$$C_8 = \{9, 10, 13, 14, 25, 26, 29, 30\} \quad ?1x?? => 0??x?$$

where x is the same value, x =0, or 1 on both sides. Moreover all possible translations of this set $C_8 \oplus b$ are also always of type io11, and they are always of the form $DDT(0x10, \delta_{ou})$. Moreover these sets form 4 disjoint cosets with 4 distinct $\delta_{ou} \in \{0x9, 0xB, 0x1A, 0x18\}$ covering the whole space of 32 points. This type of translation uniformity is quite surprising.

Example 3: By exhaustive enumeration we found that there exists exactly 8 sets of type LSS-7 which are io11. For example the set $\{0, 5, 9, 13, 20, 24, 28\}$ which is of type ???0? =>?0??? or set $\{1, 4, 8, 12, 21, 25, 29\}$ which is of type ???0? =>?1???.

Example 4: In Keccak S-box and for χ functions of any size there is a plethora of situations where the χ function fixes some bits and these bits remain invariant for any number of iterations χ^k. With χ functions this happens very frequently, see Section 5 in [ScDa24] and Sections 7.2. and 9.3. in [CoAmFo24].

Applications in Cryptanalysis: It is easy to see that if all Ascon S-boxes fix some bit say $y_i = C$ at output of one round, then [avoiding the action of round constants] at least 56 S-boxes in the next round will have $x_i = C$ at the input of next round, for any constant C. This is due to the fact that 0^{64} and 1^{64} are invariants for all five linear diffusion layer permutations used in Ascon. If in addition, our set of say 4 or 7 points is stable by XOR with 4=0x4, then the action of round constants is neutral and has no effect.

On this basis the attacker can try to construct probabilistic invariant properties on full Ascon, which is very difficult task with a lot of coding and processing of large datasets for favorable events, falling outside the scope of this paper.

6 Pairs of Hyperplanes in LAT and Related Sets

We would like to discover different ways how the Ascon can be partly linear in the spirit of and generalizing our notion of LAS. Before we study arbitrary sets of points, we will first discover many interesting sets by an indirect method. We work in the spirit of LAT (Linear Approximation Table), and we will consider all possible pairs of hyperplanes: (maximum dimension affine spaces). It was already noted in [QSMG17] that we cannot hope that the Ascon S-box would map affine spaces of dimension 3 to another space of dim 3. By extension we can hardly hope that this will happen with hyperplanes or at maximal dimension 4, and nothing like this was ever observed for any major cryptographic S-box. Therefore we must consider a question of more probabilistic or approximative nature:

Key Question: Is it possible that an S-box sends **a "large" subset** of some (maximum size) affine space of dim 4 at the input side, to a "large" subset of another affine space of dim 4 at the output side?

To see this we consider an arbitrary linear space L^i with 16 elements on the input side; and an arbitrary linear space L^o with 16 elements which is intended to be used on the output side. This is related to the well known concepts of LAT tables and Walsh coefficients. In particular we have Walsh Zeros (WZ) which are simply zeroes in LAT tables (sometimes called correlation tables). We define:

$$L^i = \{x \mid \alpha \cdot x = 0\} \quad \textit{and} \quad L^o = \{y \mid \beta \cdot y = 0\}$$

where $\cdot$ is the dot product. There are $961 = 31^2$ possible pairs of linear spaces L^i, L^o. This corresponds to both α, β being non-zero, in other terms we discard the first line and the first column of any LAT table as being degenerated.

In what follows let S^{-1} be the inverse of the Ascon S-box. We will denote the complement by the following notation: $co - X = F_2^5 \setminus X$.

Definition 9 (LAT Hyperplane Pair Approximation Test).
For any **bijective** *S-box $S : F_2^k \to F_2^k$ we consider the following natural partitioning of 2^k elements into four sets:*

$$\begin{array}{ccc} S^{-1}[L^o] \cap L^i & \cup & S^{-1}[L^o] \cap co - L^i \quad \cup \\ co - S^{-1}[L^o] \cap L^i & \cup & co - S^{-1}[L^o] \cap co - L^i \end{array}$$

and we generate these for sets for all possible pairs of hyperplanes $L^i = \{x \mid \alpha \cdot x = 0\}$ and $L^o = \{y \mid \beta \cdot y = 0\}$ with $\alpha \neq 0$ and $\beta \neq 0$.

In this paper the S-boxes studied are always bijective and $k = 5$. In this case these 4 sets are then a **disjoint** partitioning of 32 elements into 4 sets (which is not true in general). It is then easy to see that the sum of sizes in each line is $2^{k-1} = 16$ and the same is true for each column. Therefore the cardinals of each two sets lying opposite on each diagonal are identical.

Let $s_{00}, 16 - s_{00}$ be the sizes of sets defined above in the 1st line, cf. Def. 9. Then in the second line the are swapped: we have $16 - s_{00}, s_{00}$ elements. Then

the number of solutions to the LIO equation $\alpha \cdot x + \beta \cdot y = 0$ is $2s_{00}$ by the union of $S^{-1}[L^o] \cap L^i$ and the other corner with both sets complemented. Likewise the number of solutions to a negated LIO equation namely $\alpha \cdot x + \beta \cdot y = 1$ will be in general $2^k - 2s_{00}$ and both equations are represented typically by the same unique entry in LAT table. We recall that most authors define LAT in terms of relative numbers or "biases" and our numbers are shifted by 2^{k-1} accordingly and therefore we have:

$$LAT(\alpha, \beta) = 2s_{00} - 2^{k-1}$$

Example with Ascon. For example with the Ascon S-box we have 336 entries out of 961 with a partitioning of type 6+10+6+10 or vice versa. This means that in LAT we will see 336 entries of type $16 \pm 4 = 2 \cdot 6$ or $2 \cdot 10$ respectively.

Note. We rediscover a well-known fact that for a bijective S-box, all entries in the LAT table are even. In this paper we emphasize the fact that our pairs of sets of the same size are frequently (but not always) more deeply related: by a translation with a constant.

Observations. It is possible to see that 6+10 situations or $LAT(\alpha, \beta) = \pm 4$ are the most common case in all cryptographic S-boxes at size 5, and other situations are less frequent in any cryptographic S-box. We cannot really hope to have a split of type 0+16 or we would have an extremely weak cipher. We also observe very high frequencies of type 8+8 which cases are perfectly balanced (these pairs of masks α, β are sometimes called Walsh Zeros WZ). They correspond to zeros inside an LAT table, and they do not lead to any linear bias whatsoever. The situations of type 4+12 are less frequent and correspond to stronger biases with $LAT(\alpha, \beta) = \pm 8$ and happen 40 times out of 961 with Ascon S-box. There are also other possibilities such as 5+11 which do not happen in Ascon/Keccak or other quadratic S-boxes studied, yet they do happen for other S-boxes on 5 bits. In the Table 2 we report how all possible 961 pairs of affine spaces of Dim 4 lead to a variety of situations in Ascon and Keccak.

6.1 Unexpected Properties and Translations

In our LAT-based process defined in Definition 9, we always generate pairs of sets of the same size. For example we consider $S^{-1}[L^o] \cap L^i$ and $co - S^{-1}[L^o] \cap co - L^i$. Can these two (disjoint) sets be further more deeply related to each other, for example by a translation (XOR) with a constant?

For example in one case out of 961 with $\alpha = 20$ and $\beta = 11$ we have:

$$\begin{array}{rcl}
S^{-1}[L^o] \cap L^i & = & 0,3,20,21,29,31 \mapsto 0,4,10,13,20,23 \\
S^{-1}[L^o] \cap co - L^i & = & 4,6,12,13,16,17,18,19,24,27 \mapsto 3,7,9,14,16,19,25,26,29,30 \\
co - S^{-1}[L^o] \cap L^i & = & 1,2,8,9,10,11,22,23,28,30 \mapsto 5,8,11,15,17,18,22,24,27,31 \\
co - S^{-1}[L^o] \cap co - L^i & = & 5,7,14,15,25,26 \mapsto 1,2,6,12,21,28
\end{array}$$

It important to see that if these two sets with here $s_{00} = 6$ are related by translation, then also the two other complementary sets of sizes $16 - s_{00} = 10$ are also and always related by the same translation. This is due to the fact that we complement (in several ways) inside affine spaces on two sides, and affine

spaces are more likely to be stable by certain (but not all) translations. Can this be guaranteed to work? Yes, and we have the following result, cf. [CoAmFo24]:

Theorem 1. *We assume that if we translate the 1st set $S^{-1}[L^o] \cap L^i$ by a constant C we get the last set $co - S^{-1}[L^o] \cap co - L^i$, then the two remaining sets on the other diagonal are also related by translation with the same constant C.*

Proof. Our translation C is by definition a XOR of one element of L^i with parity=0 and one from $co - L^i$ with parity=1, therefore we have parity=1 and $C \in co - L^i$. Likewise we also have $C \in co - S^{-1}[L^o]$. Therefore our constant belongs to the last set: $C \in co - S^{-1}[L^o] \cap co - L^i$. We define S by adding C to all elements from the second set $S^{-1}[L^o] \cap co - L^i$, in S we always swap the parity on both sides shifting to the other coset of only two, so $S \subseteq co - S^{-1}[L^o] \cap L^i$. Now we know that the sets are of the same cardinality and so $S = co - S^{-1}[L^o] \cap L^i$. This ends the proof. □

6.2 On Sets Related to LAT in Ascon and Other S-boxes

It is easy to see in this process we obtain $4 \cdot 585 + 2 \cdot (336 + 40) = 3092$ distinct sets of points of sizes 4,6,8,10,12. We start by two essential observations.

Fact 4.1. For all quadratic S-boxes on 5 and 6 bits, and for all sizes, all inputs in all four LAT-related sets V add to zero at input side, i.e. $\sum_{in}(V) = 0$.

Counter-Examples. This is not true in AES and for 5-bit S-boxes which are not quadratic, e.g. Icepole [IceP14].

Can Our Sets Be LSS? We conjecture that for S-boxes which are not quadratic no set of any size generated as above are LSS, not even at size 4 (the easiest). It might seem that many sets of 4 will form an LSS property, and the four I/O pairs can be interpolated by linear algebra in several ways (pairs, space of dimension 5). However this is not true, the system of equations could be contradictory and they systematically are with cryptographic S-boxes. More precisely, all our sets of size 4 for quadratic S-boxes are such that the sum (XOR) of 4 inputs $\sum_{in}(V)$ is zero, and the sum of 4 outputs $\sum_{ou}(V) = 0$ is never zero. This is true in Ascon/Keccak and for all quadratic S-boxes of size 5 where the inverse is not quadratic, following Table 2 in [BoBiSa17]. Then it is impossible to obtain an LSS, because an affine space should be sent to an affine space by our affine transformation with matrix+vector.

In some cases our set of 4 points **can** be LSS, for example when the S-box is quadratic and its inverse is quadratic, cf. [CoAmFo24]. However at size 10 no sets are LSS, even though 512 of LSS-10 properties exist for this S-box. For Ascon, at size 12 we cannot hope to find an LSS-12 property for this S-box, they simply do not exist, cf. [CoAmFo24].

Cases where $\sum_{ou}(V) = 0$. We observed that for all quadratic S-boxes such that the inverse is not quadratic, outputs for all sets of size 4, 6 almost never add to zero at output side. For example in Ascon at size 6 there are only 2 such examples out of 672 which is again our special unique pair $\alpha = 20 = \rho_{in}[31]$ and $\beta = 11 = \rho_{ou}^{-1}[31]$ where $\sum_{ou}(V) = 0$.

6.3 On $\sum_{ou}(V)$ with LSS and their Relevance in Cryptanalysis.

In Appendix A.3 we find 160 examples of LSS-10 where $\sum_{ou}(V) = 0$ is never zero. In contrast with our 32 maximal size shifted $A_{11} \oplus x$ properties LSS-11 all values $\sum_{in}(V)$ and $\sum_{ou}(V)$ are equally probable.

In general, rare cases where $\sum_{ou}(V) = 0$ are those which will have an interest in cryptanalysis for various zero sum attacks, cube distinguishers and many related concepts, see for example [HuCu24].

all	8+8	6+10	4+12	0+16
961	585	336	40	0

Table 2. Ascon S-box interacting with 961 pairs of spaces of dim 4 related to LAT

6.4 On Rotation Invariance with our Partitions

A distinct questions is, does it ever happen that two sets of the same size are related by rotations such as ρ_{in} ? This never happened for any of 961 cases studied in Ascon.

Another question is if a rotation of one set can produce another "twin" set of the same size with a different pair of α, β. This happens systematically for all sets generated no matter their size due to the fact that rotations transform hyperplanes into hyperplanes. This mapping has some fixed points. In our example above, we had $S^{-1}[L^o] \cap L^i = 0, 3, 20, 21, 29, 31$ etc, and this is a very special example. In this case, all our four sets of size 6 and 10 are stable by rotations ρ_{in} and are mapped to themselves. By inspection we verified that this type of internal rotational symmetry happens just once in this exact case known as B in [CoAmFo24], where $\alpha = 20 = \rho_{in}[31]$ and $\beta = 11 = \rho_{ou}^{-1}[31]$.

Remark. In Ascon a large proportion of large size LSS properties are stable by rotations, cf. Appendix A.3, while all LLS properties are stable by translation, cf. Thm. 2.

6.5 Translation Similarities

We have tried all 961 possibilities and found that in balanced WZ cases 8+8, sometimes we get related sets of 8 points, and in **all** unbalanced cases 6+10 or 4+12 we **always** get related sets without any exception. We call these pairs WN: they are Walsh Not-Zeroes. We call WNT those which are input-shifted in our table.

This is a bizarre situation like winning in a game of heads/tails in 336+40 cases in a row, which are exactly those WN cases which are **unbalanced** like 6+10, and therefore are all potentially cryptographically significant. In fact however this property is not specific to Ascon. Similar things happen to all other quadratic S-boxes. In contrast it is important to see that this does not happen **at all** for random S-boxes, not even with a low frequency, in other terms we switch from 100% to 0% for all unbalanced WN pairs, as we will see in Table

all	8+8	6+10	4+12	0+16
961	585	336	40	0
WNT inp-shifted	240	**336**	**40**	0
unrelated	345	0	0	0

Table 3. Pairs spaces of the same size which are equivalent by translation with Ascon.

5 below in page 15. First we look at what happens in Icepole, when the S-box was altered slightly by adding just two high degree products, cf. [IceP14], which destroys overall roughly about half of our translation properties as shown in Table 4 below.

Icepole	8+8	7+9	6+10	5+11	4+12	0+16
961	435	220	240	36	30	0
WNT inp-shifted	120	10	160	6	20	0
unrelated	315	210	80	30	10	0

Table 4. Pairs of spaces equivalent by translation with Icepole S-box.

Finally we see that no weakness whatsoever is observed for Thakor, which has DMI ≈ 1.55, i.e. close to a typical random S-box. The same results with all zeros in WNT or input-shifted category are obtained for a typical S-box generated at random.

Thakor	8+8	7+9	6+10	5+11	4+12	0+16
961	270	420	196	60	15	0
WNT inp-shifted	**0**	**0**	**0**	**0**	**0**	**0**
unrelated	270	420	196	60	15	0

Table 5. Pairs spaces of the same size equivalent by translation with Thakor S-box.

7 Key General Results for all Quadratic S-boxes

We recall that we call LSS or Linearizable Sub Set any set of points V such that

$$F[x] = A \cdot x + a \quad \forall x \in V.$$

Theorem 2 (Translation Invariance of LSS for All Quadratic S-boxes). *Let F be a quadratic S-box on k bits. If there exists a set V forming an LSS-m property for $F[]$ for some integer $|V| = k > 0$,* **then** *for any affine constant d the shifted set $V \oplus d$ also forms another distinct LSS-m property, i.e. there exists another matrix B and vector b such that:*

$$F[x] = B \cdot x + b \quad \forall x \in V \oplus d.$$

Proof. In mathematics, the process of "polarization" of a polynomial produces a unique symmetric bi-linear homogeneous form, from which the original polynomial can be recovered. In the case of multivariate quadratic S-boxes, also known in the literature as Dembowski-Ostrom polynomials, researchers typically study their "difference function" cf. Section 8 in [FeHAPa21] or their "polar form" cf. Appendix 1 in [PaGo98]. In this paper our polynomials are not homogeneous quadratic and our polar form has a constant term, same as in Section 2.2. in [Be20]:

$$\Delta_F(x, d) = F[0] \oplus F[x] \oplus F[d] \oplus F[x \oplus d].$$

It is easy to show that our polar form is a bi-linear multivariate function in $k + k$ variables, and when one variable is fixed, it becomes a multivariate linear function in k variables, also known as a "linearized polynomial". These well-known facts are sufficient to prove our theorem. By the initial property we have $F[x] = A \cdot x + a \quad \forall x \in V$. Given the definition of Δ_F, we are now going to replace $F[x]$ by $\Delta_F(x, d) \oplus F[0] \oplus F[d] \oplus F[x \oplus d]$ which gives:

$$F[x \oplus d] = (\Delta_F(x, d) \oplus A \cdot x) + (a \oplus F[0] \oplus F[d]) \quad \forall x \in V$$

which is the same as:

$$F[x] = (\Delta_F(x \oplus d, d) \oplus A \cdot (x \oplus d)) + (a \oplus F[0] \oplus F[d]) \quad \forall x \in V \oplus d$$

and given that when d is fixed our $\Delta_F(x, d)$ becomes linear, we obtain:

$$F[x] = (\Delta_F(x, d) \oplus A \cdot x) + (\Delta_F(d, d) \oplus F[0] \oplus a \oplus F[d] \oplus A \cdot d) \quad \forall x \in V \oplus d$$

which provides an explicit formula of type $F[x] = B \cdot x + b \quad \forall x \in V \oplus d$ which is exactly the explicit simultaneous linearization property we claimed. □

Applications of Thm. 2. Our similarity result of Thm. 2 holds for an overwhelming majority of 5-bit S-boxes used in applied cryptography including SHA-3, SHAKE and Ascon.

Corollary. It follows that all the 80 of LAS-2 properties with 4 points, from Eurocrypt 2017 cf. [QSMG17] are also stable by arbitrary translations at the input side. This is not true at the output side.

Related Research. Our property can also be rephrased as follows in terms of Boura-Canteaut definition in [BoCa14]: Ascon S-box can be seen to be $(2, 5)$-linear in terms of [BoCa14].

Important Relaxation. In this paper we drop the requirement of sets of points V forming an affine space in [QSMG17,BoCa14], yet our LSS/LAS properties hold again and again for every coset of a given set or space respectively.

More on 80 LAS-2 properties. A detailed enumeration shows that they split into two disjoint classes of 40 cosets stable by arbitrary translations and that properties spanned with $15 \cdot 8$ sets DDT() of size 4 and $5 \cdot 4$ sets DDT() of size 8, both cover uniformly the whole set of 80 LAS-2 properties several times.

Observations. Our result is somewhat very surprising, because we have first fully linearized a **non-linear** S-box on a subset, where it is equal to a multivariate linear function with a matrix and a constant, and then we claim that simple transformed variants of our linear property cover the whole space F_2^k several times (the more the better). This is not the end of the story. It turns out that whole affine spaces, which are also configurations of LSS properties, disjoint or not, can also by translated in arbitrary ways.

7.1 A Level Two Theorem for all Quadratic S-boxes

Initially it might seem that this type of extensions of our result would be trivial. After all, if we had say a configuration of four LSS of size say 5+11+5+11, which definitely exist for Ascon, we can shift all these sets of points by a common constant a, and Thm. 2 says it will be again four LSS properties of the same size. However until now our theorem was existential, it just says that a certain matrix exists, and as such it cannot possibly guarantee that these matrices would form an affine space of a surprisingly small dimension, related or equal to the initial (small) dimension. If this happened this would be either accidental, or it needs to happen due to a deeper next level theorem. This result is new and was not published before.

Theorem 3 (Translation Invariance of LSS Dimension for MQ S-boxes). *Let F be a quadratic S-box on k bits. We assume that for a certain D there exist a set of LSS properties V_i such that the associated matrices A_i lie inside an affine space, and the sets V_i cover the whole space (not always disjoint). Then for any constant a (same for all sets) the shifted sets $a \oplus V_i$ also cover the whole space and there exist a set of associated matrices A'_i forming an affine space of dimension at most D.*

Proof. Again we write:

$$\Delta_F(x, a) = F[0] \oplus F[x] \oplus F[a] \oplus F[x \oplus a].$$

which is bi-linear multivariate function in $k + k$ variables. We assume that $F[x] = A_i \cdot x + b_i \quad \forall x \in V_i$. Given the definition of Δ_F, we replace $F[x]$ by $\Delta_F(x, a) \oplus F[0] \oplus F[a] \oplus F[x \oplus a]$ to obtain:

$$F[x \oplus a] = (\Delta_F(x, a) \oplus A_i \cdot x) + (b_i \oplus F[0] \oplus F[a]) \quad \forall x \in V_i$$

which is the same as:

$$F[x] = (\Delta_F(x \oplus a, a) \oplus A_i \cdot (x \oplus a)) + (b_i \oplus F[0] \oplus F[a]) \quad \forall x \in V_i \oplus a$$

and given that when a is fixed our $\Delta_F(x, a)$ becomes linear, we obtain:

$$F[x] = (\Delta_F(x, a) \oplus A_i \cdot x) + (\Delta_F(a, a) \oplus F[0] \oplus b_i \oplus F[a] \oplus A_i \cdot a) \quad \forall x \in V_i \oplus a$$

which for every i clearly does provide an explicit LSS formula of type $F[x] = C_i \cdot x + c_i \quad \forall x \in V_i \oplus a$. We are almost done: it remains to show that the matrices A'_i obtained from $(\Delta_F(x, a) \oplus A_i \cdot x)$ and nothing else would live in a space of dimension at most D. This is trivial because $(\Delta_F(x, a)$ translates into a fixed matrix for every i not depending which LSS matrix A_i and set V_i we are using. □

8 On Rotational Symmetries in Ascon and Keccak

In this section we study some partitions of 32 points into disjoint or/and overlapping sets which exist because the LSS-dimension of the Ascon and Keccak S-box is only 2. However. Ascon has (hidden) rotational symmetries and sets of points we obtain in our work rarely ever behave like configurations of points chosen at random. Following [CoAmFo24] there exists a **unique** and natural partitioning of all the 32 points into 6 sets of 5 points and two sets of size 1 based on rotational symmetries of the Keccak S-box.

class	Ascon input	Keccak Ki	Kx=$Ki \oplus Ko$	Keccak Ko	Ascon output
a	6,13,18,24,27	7,14,19,25,28	1,2,4,8,16	15,23,27,29,30	3,7,9,16,25
b	1,2,8,23,28	3,6,12,17,24	1,2,4,8,16	11,13,21,22,26	11,22,24,27,31
q	9,10,11,22,30	15,23,27,29,30	1,2,4,8,16	7,14,19,25,28	5,8,15,17,18
r	5,7,14,15,25	11,13,21,22,26	5,9,10,18,20	1,2,4,8,16	2,6,12,21,28
s	3,20,21,29,31	5,9,10,18,20	5,9,10,18,20	3,6,12,17,24	0,10,13,20,23
t	4,12,16,17,19	1,2,4,8,16	1,2,4,8,16	5,9,10,18,20	14,19,26,29,30
y	0	0	0	0	4
z	26	31	0	31	1

Table 6. Classification of points at input and output side of Ascon based on the how circular rotations act on the states of the underlying Keccak S-box.

Following this table we write: $t^i = \{4, 12, 16, 17, 19\}$ to say that t represents a set of 5 points $\{4, 12, 16, 17, 19\}$ at the input side, and in the same way we write $t^o = S[t^i] = \{26, 29, 30, 19, 14\}$ to say that the letter t represents another set of 5 points at the output side. We recall that the set $t^i = \{4, 12, 16, 17, 19\}$ are exactly five input differentials δ_{in} in Ascon of maximum strength. More precisely these 5 values are covering exactly all $DDT(\delta_{in}, \delta_{ou})$ sets of size 8 with 4 values δ_{ou} for each of $\delta_{in} \in t^i = \{4, 12, 16, 17, 19\}$ and all these 20 values δ_{ou} are distinct output differences on 5 bits. We have $A_{11} = s^i \cup t^i \cup y^i$ and $B_{11} = q^i \cup r^i \cup z^i$.

9 Space Covering with Ascon/Keccak and Max-Size LSS

There are thousands of ways in which the Ascon S-box can be described as having LSS dimension only 2 with some 4 matrices and decomposed accordingly using Algorithm 1. This leads to a large variety where 32 inputs are divided in 4 disjoint sets of type 5+5+11+11 or 8+8+9+7. The universe of possible sets of say 5,6,7,8,9,10,11 points which appear in these approximations seems excessively rich; as it is stable by arbitrary translations, see Thm. 2. It is then natural to "grow" our sets: for any affine approximation of type $A.x \oplus c$, additional points x might also satisfy it, and the matrix A it not always unique. For example we found many space partitions of type 11+9+6+6 points and then in fact we could extend them to four LSS properties with 11+11+11+10 points respectively, thus it contains three maximum size LSS-11 properties (with some overlap) and one more sub-optimal of size 10. This property is the strongest LSS covering property ever found in Ascon/Keccak. Here the four matrices in these properties do not add to zero anymore: an LSS extension of 5 or 6 points to 10 can operate with a different matrix than initially. We observe also that 2 out of 3 sets of type LSS-11 are disjoint in each of 128 properties.

On Maximum Strength Space Coverings in Ascon/Keccak. By growing our sets and achieving some local maxima (A is not unique and extensions are not unique) it is possible to seet that we find fewer distinct space coverings: there are fewer LSS properties at large sizes. We found that there only and exactly 128 distinct space coverings of type 11+11+11+10 for Ascon/Keccak.

9.1 On Partitioning the Space in Four

It is perfectly natural to divide the space in 4 sets (and not more) in Ascon/Keccak. This seems to be optimal with various LSS with nice space coverings of type 11+11+11+10 with overlaps. Accidentally, or not, we also observe the all 20 maximum strength DDT() sets of size 8, also always form a partitioning of the whole space in four overlapping sets. Are these two types of input space coverings related? Yes. As much as LSS generalize LAS-2, we expect some sort of precise extension of Observation 1 from Eurocrypt 2017. More precisely, in Observation 1 in [QSMG17] the LAS-2 sets are shown to be derived or equal from (larger or equal) DDT sets. Here LSS sets are larger than any DDT sets in existence, however they still strongly structure the space of DDT sets with some quite precise identities as we will show now. This result is also new.

Theorem 4 (Interaction of LSS-11 and DDT Sets). *Let F be the Ascon or Keccak S-box on $k = 5$ bits or an affine equivalent or these boxes.*
For each LSS-11 set which are always of the form $A_{11} \oplus s$, and any of 20 DDT(t, u) sets of size 8 in existence, where $t \in t^i = \{4, 12, 16, 17, 19\}$, their intersection has exactly 1,2 or 6 points out of 8 and we never observe 0,3,4,5,7,8. For each t, and each (unique) size 6 intersection of one LSS-11 and one DDT-8, these 6 points where LSS and DDT sets intersect are **exactly** *the 6 defined by:*

$$H_{s,t} = A_{11} \oplus s \ \cap \ A_{11} \oplus t \oplus s.$$

Proof. First we verify by inspection (brute force) for any of 32 properties LSS-11 in existence that their intersection with four DDT(t,u) sets for $t \in t^i = \{4,12,16,17,19\}$ in existence is always the same partitioning 11=6+2+2+1. This uniformity is clearly strongly related to the fact that all the LSS-11 properties belong to one single class by translation, cf. Thm. 2 and there are no other properties of size 11 or better, see [CoAmFo24]. Then we examine all properties of size 6 and it is easy to see that if V_i is an LSS-11 property of type $S[x] = A_i.x \oplus c_i$ for all $x \in V_i$ then we must have that any intersection of V_i and any DDT property will be exactly of the form DDT$(t, A_i.t)$. Now we are going to hypothesize that for each $V_i = A_{11} \oplus s_i$, their unique interesection of size 6 which is necessarily of the form DDT$(t, A_i.t)$ is also equal to set $H_{s,t} = A_{11} \oplus s_i \cap A_{11} \oplus t \oplus s_i$ which sets are also always of size 6 for all $t \in t^i = \{4,12,16,17,19\}$, see [CoAmFo24]. Because their cardinalities are the same and $H_{s_i,t} \subseteq V_i$ by its very definition, it is sufficient to show that $H_{s_i,t} \subseteq DDT(t, A_i.t)$. Here is the proof. For any $x \in H$ we have both $S[x] = A_i.x \oplus c_i$ and $S[x \oplus t] = A_i.(x \oplus t) \oplus c_i$. Hence

$$S[x] \oplus S[x \oplus t] = A_i.x \oplus c_i \oplus A_i.(x \oplus t) \oplus c_i = A_i.t$$

this ends the proof of equality of these two sets.

Further Observations. In addition, for any given input difference, $t \in t^i = \{4,12,16,17,19\}$, which are exactly all "strong" differentials in Ascon/Keccak, there are 4 possible output differences u which are disjoint for each of five $t \in t^i = \{4,12,16,17,19\}$. For each DDT$(t,u)$ there are potentially $\binom{8}{2} = 28$ pairs x, x' and out of these for any LSS-11 set of the form $A_{11} \oplus s$, for one u, we have $\binom{6}{2} = 15$ pairs x, x' out of 28, where BOTH

$$x, x' \in A_{11} \oplus s.$$

We see here that linear approximation properties of maximum possible size (LSS-11) and differential properties of maximum possible size (DDT-8) are either disjoint of strongly co-exist in terms of sets and conditional probabilities. These properties are expected to lead to non-Markovian behavior inside Ascon/Keccak and related cryptographic primitives.

Related Work. All the $\delta_{in} \in t^i$ for which we obtain size 6, have the "undisturbed bits" UDB property extensively studied by Tezcan in multiple papers for Ascon [TeAs16,TeDi19] which is also related to the older concept of "linear structures" [MaTe14].

Further Intersections of LSS-11. We observe that our set $H_{s,t}$ are of maximum size 6 if and only if $t \in t^i = \{4,12,16,17,19\}$ furthermore it is of size 4 if and only if $t \in s^i \cup r^i \cup b^i$ which 15 values are **exactly** those lines in DDT table which contain any 4, furthermore it is of size 2 if and only if $t \in q^i \cup a^i$, etc. These intersections are further studied in detail in [CoAmFo24] and below in Section 10.

Applications in Cryptanalysis. Our Thm. 4 opens avenues of research in advanced combined differential and linear cryptanalysis. It says basically that linear and differential properties of maximum strength, which are in fact those which might interest the attacker, are strongly correlated to each other. In older

works differential and linear approximations occur in disjoint remote parts of the cipher and are connected together with so called connectors, see [QSMG17] and [HaDeEi24,TeDi19,TeDiLi23] to name only a few. Here we need to look for new types of differential-linear attacks: where linear and differential properties would be used jointly more or less in each round, and they could work together, an hopefully re-inforce each other in terms of conditional probabilities. We refer to [CoQi20] to see an example of an attack, where differential and multiple linear properties mix together at every round inside one single attack, and where differential probabilities are substantially improved by additional constraints orchestrated by the attacker.

10 Interaction of DDT Sets and Maximum Size LSS

As a complement of Thm. 4 and following [CoAmFo24] here is a short summary of how all known maximum size LSS-11 properties interact with themselves (i.e. their own affine shifts):

Observations on Self Similarity of A_{11}. We found that:

1. The intersection of A_{11} with another variant $A_{11} \oplus x$ is always not empty except for $B_{11} = A_{11} \oplus 0x1A$, when $x = 26 = 0x1A$.
2. The intersection of A_{11} with another variant $A_{11} \oplus x$ is of maximum size 6 in exactly 5 cases where $x \in \{4, 12, 16, 17, 19\}$ which set is sometimes called t, cf. Table 6 which are exactly five of those lines δ_{in} in the DDT of Ascon/Keccak S-box which contain 8, and which are therefore exactly all the input differentials in Ascon of maximum strength 8/32. These 5 δ_{in} also amount to half of 10 well-known "undisturbed bits" UDB properties in [TeAs16,TeDi19]. The relevant DDT sets of size 8 are also studied in Thm. 4.
3. The intersection of A_{11} with another variant $A_{11} \oplus x$ is of size 4 when $x \in \{1, 2, 3, 5, 7, 8, 14, 15, 20, 21, 23, 25, 28, 29, 31\}$ which set was sometimes studied as a union of 3 sets: $s \cup r \cup b$ cf. Table 6 and which 15 values are **exactly** those lines in DDT table which contain any 4 numbers.
4. The intersection of A_{11} with another variant $A_{11} \oplus x$ is of size 2 when $x \in \{6, 9, 10, 11, 13, 18, 22, 24, 27, 30\}$ which set is a.k.a. $q \cup a$ which 10 values are **exactly** those lines in DDT table except for the special case of $x = 26 = 0x1A$ already used in B_{11}.

Remark on A_{11} and Differential Cryptanalysis. Each time when intersection of two A_{11} is of maximum size 6, we obtain a property which concerns 16/32 points and yet captures 6/8 of the points which might interest the attacker the most: those concerned by differentials true with maximum probability of 8/32. This is type of situation where various approximations work jointly or coincide or are amplified when studied on a reduced set of points are also expected to lead new types of differential-linear attacks, guess then determine collision finding attacks, or/and algebraic key recovery attacks.

11 Conclusion

In this paper we studied a variety of linear, differential, rotational, algebraic, space partitioning, and translation self-similarity properties of Ascon and Keccak S-box and we compare it to some other well-known S-boxes. We observed that all quadratic S-boxes have several quite strong properties, and likewise rotation invariant S-boxes frequently also have several quite strong properties, and also that all known S-boxes with high DMI have many undesirable properties. Ascon S-box and Keccak S-box combine all these three highly problematic features.

More precisely we studied two major notions of full-size functional linear approximations for the whole S-box: the LAS with affine spaces and LSS with sets of arbitrary points. This generalizes previous research on combining differential and linear properties in Ascon and Keccak. We also study sets derived from LAT tables and their symmetries. We demonstrated that in many cases these properties are stable by translations [and frequently also by rotations] and they interact very strongly with differential properties. For example we focus on best maximum size LSS properties and show they strongly correlate with essentially every line in the DDT table, which also form space coverings, forming a consistent two-layer super-approximation.

As a potential application, these properties are expected to lead to new types of non-Markovian differential-linear attacks where the attacker should be able to manipulate the probabilities of various differentials to his advantage, possibly similar as in [CoQi20].

11.1 Executive Summary for NIST

1. Every S-box can be linearized to some extent.
2. Ascon/Keccak S-boxes are outliers exhibiting vast quantities of surprisingly large size and highly regular simultaneous linear approximations.
3. LAT, LSS and DDT sets in Ascon/Keccak interact very strongly and exhibit very high levels of regularity cf. Section 6.5, Thm. 4 and Section 10.
4. Large LSS properties form super-structures stable by translation cf. Thm. 2 and Thm. 3 covering the whole space such as 11+11+11+10 cf. Section 9.
5. We don't have an attack (or not yet) but we have a concern.
6. We suggest that Ascon should be **upgraded** and use S-boxes which are **neither rotation symmetric nor quadratic** and have a lower DMI value.
7. This should be possible without significantly impacting Ascon speed and resource needs.

References

[GLL19] Guangpu Gao, Dongdai Lin and Wenfen Liu: A Note on Rotation Symmetric S-boxes Journ. of Syst. Science and Complexity, vol. 32, pp. 1460–1472, 2019.

[Be20] Ward Beullens: *Improved Cryptanalysis of UOV and Rainbow,* At eprint.iacr.org/2020/1343, October 2020.

[BCQ04] Alex Biryukov, Christophe De Cannière, and Michael Quisquater: *On Multiple Linear Approximations,* In eprint.iacr.org/2004/057.

[St8b16] Erik Boss, Vincent Grosso, Tim Güneysu, Gregor Leander, Amir Moradi, and Tobias Schneider: *Strong 8-bit Sboxes with Efficient Masking in Hardware,* https://eprint.iacr.org/2016/647.pdf

[BrChMa04] Luca Breveglieri, Alessandra Cherubini, Marco Macchetti: *On the Generalized Linear Equivalence of Functions over Finite Fields,* In Asiacrypt 2004, LLNCS 3329, pp. 79-91, Springer 2004.

[BoBiSa17] Dušan Božilov, Begül Bilgin and HacıAli Şahin: *A Note on 5-bit Quadratic Permutations' Classification,* In TOSC 2017 Iss. 1, pp. 398-404, March 2017.

[BoFiPe13] Joan Boyar, Magnus Find, and René Peralta: *Four measures of nonlinearity,* In CIAC, LNCS 7878, pp. 61-72, 2013.

[BoPePo00] Joan Boyar, René Peralta, and Denis Pochuev. *On the multiplicative complexity of Boolean functions over the basis (∧,⊕,1).* In Theor. Comput. Sci., 235(1):43–57, 2000.

[BoCa14] Christina Boura and Anne Canteaut: *A new criterion for avoiding the propagation of linear relations through an Sbox,* In FSE 2013, LNCS 8424, pp 585–604, Springer, 2014.

[BIK24] Lilya Budaghyan, José L. Imaña, and Nikolay Kaleyski Low-Complexity Hardware Architecture of APN Permutations Using TU-Decomposition, IEEE Transactions on Circuits and Systems I Regular Papers pp: 1-11, 2024.

[ChDo04] Pascale Charpin: *Normal Boolean functions.* In Journal of Complexity, vol. 20, issues 2–3, 245–265 (2004).

[LFT35] Liyana Chew, Nizam Chew and Eddie Shahril Ismail: *S-box Construction Based on Linear Fractional Transformation and Permutation Function,* In Symmetry 2020, 12(5), 826, MDPI, 2020.

[CR19] Christophe Clavier and Léo Reynaud: *Systematic and Random Searches for Compact 4-Bit and 8-Bit Cryptographic S-Boxes,* eprint.iacr.org/2019/1379, 1 Dec 2019.

[CoNi23] Nicolas T. Courtois: Cryptanalysis of Ascon – An Information Theoretic Perspective – A Position Paper , slides presented by Nicolas Courtois, 22 June 2023, in Lightweight Cryptography Workshop: Day 2, Part 5, at https://www.nist.gov/video/lightweight-cryptography-workshop-day-2-part-5

[CoDe08] Nicolas T. Courtois, Blandine Debraize: *Algebraic Description and Simultaneous Linear Approximations of Addition in Snow 2.0.* In ICICS 2008: pp. 328-344, October 2008.

[CoGo11] Nicolas T. Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* At eprint.iacr.org/2011/626, last revised on 6 Nov. 2015.

[CoAmFo24] Nicolas T. Courtois, Frédéric Amiel, and Alexandre Bonnard de Fonvillars: *On Maximum Size Simultaneous Linear Approximations in Ascon and Keccak and Related Translation and Differential Properties,* In eprint.iacr.org/2024/802, last updated in Oct. 2024.

[CARG19] Nicolas T. Courtois, Matteo Abbondati, Hamy Ratoanina, Marek Grajek: *Systematic Construction of Nonlinear Product Attacks on Block Ciphers,* In ICISC 2019, pp. 20–51, Springer LNCS 11975, 2019.

[CoQi20] Nicolas T. Courtois, Jean-Jacques Quisquater: *Can a Differential Attack Work for an Arbitrarily Large Number of Rounds?,* In ICISC 2020, pp. 157–181, Springer, 2020.

[CoPoSc08] Nicolas Courtois, Maria Bristena Oprisanu, and Klaus Schmeh: *Linear cryptanalysis and block cipher design in East Germany in the 1970s,* In Cryptologia, volume 43, issue 1, pp. 2-22, December 2018.

[DaPhD95] Joan Daemen: *Cipher and Hash Function Design Strategies based on linear and differential cryptanalysis,* PhD thesis, March 1995, reformatted in 2004.

[BoPeTi19] Xavier Bonnetain, Léo Perrin and Shizhu Tian: *Anomalies and Vector Space Search: Tools for S-Box Analysis (Full Version)*, https://eprint.iacr.org/2019/528

[FaAr08] A. Farhadian M.R. Aref: *Efficient method for simplifying and approximating the S-boxes based on power functions,* In IET Information Security, 2008.

[FeHAPa21] Neranga Fernando, Sartaj Ul Hasan, Mohit Pal: *Dembowski-Ostrom polynomials and reversed Dickson polynomials,* arxiv.org/abs/1905.01767, May 2021.

[GJNQSS16] Jian Guo, Jérémy Jean, Ivica Nikolić, Kexin Qiao, Yu Sasaki, Siang Meng Sim: Invariant Subspace Attack Against Midori64 and The Resistance Criteria for S-box Designs. In TOSC 2016, No. 1, pp. 3-56, Springern, 2016.

[HaDeEi24] Hosein Hadipour, Patrick Derbez and Maria Eichlseder: *Revisiting Differential-Linear Attacks via a Boomerang Perspective with Application to AES, Ascon, CLEFIA, SKINNY, PRESENT, KNOT, TWINE, WARP, LBlock, Simeck, and SERPENT,* eprint.iacr.org/2024/255

[HuCu24] Kai Hu: *Improved Conditional Cube Attacks on Ascon AEADs in Nonce-Respecting Settings – with a Break-Fix Strategy,* eprint.iacr.org/2024/743, May 2024.

[MaTe14] Rusydi H. Makarim and Cihangir Tezcan: *Relating Undisturbed Bits to Other Properties of Substitution Boxes,* eprint.iacr.org/2014/855

[MeBi19] Lauren De Meyer, Begül Bilgin: *Classification of Balanced Quadratic Functions,* In TOSC 2019 vol 2, pp. 169–192, June 2019.

[IceP14] Paweł Morawiecki, Kris Gaj, Ekawat Homsirikamol, Krystian Matusiewicz, Josef Pieprzyk, Marcin Rogawski, Marian Srebrny and Marcin Wójcik: *ICEPOLE: High-speed, Hardware-oriented Authenticated Encryption* , eprint.iacr.org/2014/266, April 2014.

[PeTh17] Léo Paul Perrin: *Cryptanalysis, Reverse-Engineering and Design of Symmetric Cryptographic Algorithms,* PhD thesis, University of Luxembourg, April 2017. hdl.handle.net/10993/31195

[PeRE18] Léo Perrin: *S-Box Reverse-Engineering Boolean Functions, American/Russian Standards, and Butterflies,* Slides presented at CECC 2018, who.paris.inria.fr/Leo.Perrin/slides/slides-cecc.pdf 6 June 2018.

[SaGl14] Simona Samardjiska and Danilo Gligoroski: *Linearity Measures for Multivariate Public Key Cryptography,* In Securware 2014, pp. 157–166, 2014.

[ScDa24] Jan Schoone, Joan Daemen: *The state diagram of Chi,* In Designs, Codes and Cryptography, Vol. 92, pp. 1393–1421, Jan 2024.

[QSMG17] Kexin Qiao, Ling Song, Meicheng Liu, and Jian Guo: *New Collision Attacks on Round-Reduced Keccak,* eprint.iacr.org/2017/128.pdf, April 2017. Minor revision w.r.t. Eurocrypt 2017, pp. 216–243, LNCS 10212.

[TeDi19] Cihangir Tezcan: *Distinguishers for Reduced Round Ascon, DryGASCON, and Shamash Permutations,* In NIST Lightweight Cryptography Workshop, 2019. https://www.nist.gov/news-events/events/2019/11/lightweight-cryptography-workshop-2019

[TeAs16] Cihangir Tezcan: Truncated, Impossible, and Improbable Differential Analysis of ASCON, International Conference on Information Systems Security and Privacy, December 2016.

[PaGo98] Jacques Patarin, Louis Goubin, Nicolas T. Courtois: *C*-+ and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai.* In Asiacrypt 1998, LNCS 1514, pp. 35-49, October 1998.

[TeDiLi23] Cihangir Tezcan: *Differential-Linear Cryptanalysis of ASCON: Theory vs. Practice*, slides presented at NIST on 22 June 2023, https://csrc.nist.gov/csrc/media/Presentations/2023/differential-linear-cryptanalysis-of-ascon/images-media/ june-22-tezcan-differential-linear.pdf

[wikiMI] Wikipedia article: *Mutual Information*, en.wikipedia.org/wiki/Mutual_information, consulted 3 May 2024.

A More on Maximum and Near-Maximum Strength Linearization Properties

We recall from [CoAmFo24] that the best possible LSS property in Ascon and Keccak has 11 points and no solution at size 12 exists anymore. There are exactly 32 solutions and that they come in 16 times two pairs of 11 disjoint points, forming 16 dual linear approximations of Ascon spanning 22 out of 32 points.

- $A_{11} = s^i + t^i + \{0\} = 0, 3, 4, 12, 16, 17, 19, 20, 21, 29, 31$
- $B_{11} = r^i + q^i + \{26\} = 5, 7, 9, 10, 11, 14, 15, 22, 25, 26, 30$

where $B_{11} = A_{11} \oplus 0x1A$, which is the same as shifting by 26 in decimal.

Translation Invariance. Following Thm. 2 which holds for all quadratic S-boxes, but in general, each pair of LSS-11 properties in Ascon is invariant by arbitrary input-side translation by a constant. These properties span uniformly the whole space, and for different configurations of points there will be typically several way to approximate them using these properties. All solutions at size 11 are isomorphic by a translation with a constant. Solutions at size 10 or smaller are no longer all isomorphic and come in 5 distinct classes, see Appendix A.3 below.

A.1 Remarkable Identities

There exist a plethora of remarkable identities between sets of size 11 and two distinguished sets of 16 points forming a space of dimension 4, see Section 9 pages 32-37 in [CoAmFo24]. For example we have:

$$a^o \cup s^o \cup y^o = t^o \oplus t^o \oplus y^o \qquad b^o \cup q^o \cup z^o = r^o \oplus r^o \oplus z^o$$
$$a^o \cup t^o \cup y^o = s^o \oplus s^o \oplus y^o \qquad b^o \cup r^o \cup z^o = q^o \oplus q^o \oplus z^o$$
$$a^i \cup r^i \cup z^i = t^i \oplus t^i \oplus z^i \qquad s^i \cup q^i \cup y^i = b^i \oplus b^i \oplus y^i$$
$$a^i \cup t^i \cup z^i = r^i \oplus r^i \oplus z^i \qquad s^i \cup b^i \cup y^i = q^i \oplus q^i \oplus y^i$$
$$A^i_{11} \cup s^i \cup t^i = B^i_{11} \oplus z^i \qquad B^i_{11} \cup q^i \cup r^i = A^i_{11} \oplus z^i$$
$$A^i_{11} \oplus A^i_{11} = F_2^5 \setminus 26 \qquad B^i_{11} \oplus B^i_{11} = F_2^5 \setminus 26$$
$$A^o_{11} \oplus A^o_{11} = a^o \cup s^o \cup t^o \cup y^o \qquad B^o_{11} \oplus B^o_{11} = a^o \cup s^o \cup t^o \cup y^o$$
$$A^o_{11} \cup a^o = A^o_{11} \oplus A^o_{11} = B^o_{11} \oplus B^o_{11} \qquad B^o_{11} \cup b^o = B^o_{11} \oplus a^o = A^o_{11} \oplus B^o_{11}$$

A.2 Further Study of LSS-11 Properties Based on A_{11}

We have verified that for all affine shifts of our LSS-11 property, which are always of the form $A_{11} \oplus x$, the matrices are invertible in 30/32 cases for all $x \neq 0$ and $x \neq 26$. In 2/32 cases which are exactly A_{11} and B_{11} for which the matrices are shown above, the rank drops to 4. It is a bit surprising to discover that the right kernel space of each matrix is the same and contains exactly one non-zero element which is in both cases the same and equal to 26.

Constant parts inside these approximations are highly biased. In [CoAmFo24] we discover that in half or 16/32 of these approximations of type $A_{11} \oplus x$, the constant part on 5 bits is the same and equal to $0x4 = 00100$.

A.3 On Almost-Maximal size LSS-10 Properties

It is easy to see that the number of LSS$-m$ properties increases as m goes down. For example, any subset of 10 inside an LSS-11 property form a valid albeit trivial LSS-10 property. In addition we found there exists exactly 5 **non-trivial** LSS-10 properties. They form 5 distinct classes or orbits w.r.t. translations of LSS-10 properties of Ascon. Each these 5 classes generates 32 distinct translations of size 10 following Thm. 2. We list these remarkable sets of 10 here together with a XOR of all elements at output side, which is a translation invariant for all sets (true because 10 is even). Overall our 16 properties are freely translatable by translations and rotations. We have exactly 5 classes with 32 affine shifts each:

- 0,1,3,4,5,9,16,19,20,24 $\sum_{in}(V)$=5
- 0,3,4,15,16,17,19,20,21,28 $\sum_{in}(V)$=7
- 0,4,8,11,17,21,23,24,25,27 $\sum_{in}(V)$=14
- 0,2,3,4,12,14,16,19,20,31 $\sum_{in}(V)$=15
- 0,2,4,12,14,16,17,20,21,29 $\sum_{in}(V)$=25

The set of 5 sums $\sigma_{ou}(V)$ is exactly $r^i = \{5, 7, 14, 15, 25\} \subset B_{11}$. Four out of five from these classes are used in space coverings of type 11+11+11+10.

11. Joachim Strömbergson, February 7, 2025

Hello,

Will the draft be updated with test vectors? Or are (will) they be available in a separate document? I'm working on a HW implementation for which I would be happy to present results. But currently I can't complete thet work since I don't know if the implementation is functionally correct.

BR
Joachim Strömbergson - Assured AB.

12. Canadian Centre for Cyber Security, February 7, 2025

Comments on NIST SP 800-232 IPD

Section 2: Table 2

The definition for 'digest' could be made more concrete. For instance, replacing the current definition 'Hash value' with 'Output of a hash function'

The so-called 'nonce-uniqueness requirement' is used in the definition of both nonce-misuse and nonce-respecting. A concrete definition of this should be added to the Table for clarity.

In the definition of the 'width', the definition could be clearer if the size unit (bits) is specified: "The state size, in bits, of the underlying permutation".

Section 2: Table 3

In the definition of s(i, j) the comma separating the ranges for i and j is italicized. To be consistent with formatting elsewhere in the document (e.g. compare to $Si[j]$), that comma should not be italicized, and should be outside of a mathematical formatting environment.

Section 2.1: Auxiliary Functions

It should be made more clear/explicit that, in the case that |X| mod r = 0, the final block is empty, but not null. i.e. A length-0 final block is not the absence of a block, and eventually will result in a full block of padding.

Section 3.2: Constant-Addition Layer

The authors could comment on the original rationale for the choice of specific constants, and, in particular, why they are chosen to have such low entropy (which is commented on in Refs 1-3).

Section 4.1.1: Ascon-AEAD128 – Encryption

In subsection 2 on the processing of initial data, it should again, be made clear, that if it turns out that the associated data aligns with a block-boundary, that the Am block needs to be a full padding block (i.e. 0x10000…)

In Algorithm 3, there is a syntax error in setting the last block of the ciphertext. The index [0,ℓ-1] incorrectly uses a comma, when it should be [0:ℓ-1].

Algorithm 4: Ascon-AEAD128.dec

The authors could consider specifying more clearly what happens in the final stage of processing ciphertext, when ℓ = 0. Specifically, if ℓ = 0, then ($\widetilde{P_n}$) then is empty. While this is certainly true based on the slice definition of the state, this edge case could be made clearer in the algorithm specification.

In the same section of the algorithm, there is a syntax error for the internal state update after the XOR operation that recovers the final (partial) plaintext block. The current equation has the indices

[ℓ,127], which should be [ℓ:127] (replace comma with colon). A similar error is made on the next line with index [0,ℓ-1].

Section 4.1.2: Section 3 – Processing ciphertext

On lines 442-445 it would be clearer if some commentary was provided on what happens if the $(\widetilde{C_n})$ block is empty. I.e. if $(\widetilde{C_n})$ ĩs empty than so is $(\widetilde{P_n})$ ãnd the state is obtained by XORing with the full padding (0x1000…) block

Section 4.2.2

The wording in the last paragraph (lines 471-474) could be made more precise. Specifically, while the authors provide a reference for the context-commitment portion of the nonce-masking caveat, they should elaborate on what is meant by "related-key security" or provide a reference.

The final paragraph (lines 471-474) could also be broken up into two sentences. I would suggest something like: "…and related-key security are note concerns. This is because the encryption of Ascon-AEAD128 with nonce masking will output the same (C, T) pair for two different input tuples (K||K', N, A, P) and (K||K'', N', A, P) whenever N + K' = N' + K''."

Section 4.4

Regarding Lines 502-503, more recent analysis in ePrint 2024/1969 provides additional insight on security bounds, and further points out an error in one of the bounds used for security under nonce-misuse authenticity in the current reference 17.

Section 5.1

On Line 556, the authors could clarify here that, unlike the AEAD mode, which uses the p[12] and p[8] permutations for the initialization and processing phases (respectively), the hash mode uses the p[12] permutation for both. Furthermore, the authors should mention that, unlike the AEAD mode, the rate is smaller (64 vs 128) in the Ascon-Hash modes.

On Line 556, there is a grammatical error in the last sentence before Figure 7: modify to "Note that L, the length of the output, and is 256 bits for Ascon-Hash256 and L > 0 for Ascon-XOF128."

On Line 568, similar to the AEAD section, clarify that the final block can be empty if the message is block-aligned would be an improvement. For instance, updating the line to: "The partial (or empty) block $(\widetilde{M_n})$

Section 5.2

On Line 602, it would be clearer to state that the absorbing phase is the same as in Ascon-Hash256, rather than it is "similar to the AEAD". If the authors prefer to still compare it to AEAD however, the text 'AEAD' should be replaced with 'Ascon-AEAD128' and it should again be noted how it differs (the rate is changed, and the permutation p[8] is changed to p[12])

Section 5.3

On Lines 639-642, the authors should explicitly state that, once the processed customization string is prepended to the message blocks, the concatenation of the two is treated equivalently to the message data in Ascon-XOF128

13. John Mattsson, February 7, 2025

[attachment begins on next page]

ERICSSON

Date: February 7, 2024

Ericsson AB
Group Function Technology
SE-164 80 Stockholm
SWEDEN

Comments on SP 800-232 Ascon-Based Cryptography

Dear NIST,

Thanks for your continuous efforts to produce well-written, user-friendly, and open-access security documents. We welcome NIST's plans to standardize Ascon-based cryptography.

Our primary concerns with the initial public draft are the absence of an Ascon-based Key Derivation Function (KDF) and the proposed constraints for truncated tags. We would prefer that NIST publish Ascon-AEAD128 alongside an Ascon-based KDF that can be used to key it. Additionally, we believe SP 800-232 should not be published with the currently suggested constraints for truncated tags. These constraints should be significantly revised, or truncation should be prohibited entirely.

Please find below our comments on the Initial Public Draft of SP 800-232:

General:

- The specification should clarify that all modes of Ascon are quantum-resistant and specify the security category [1] each mode provide. It would be interesting to know whether nonce masking increases the security level of Ascon-AEAD128 as suggested in [2]. Note that nonce masking does for example not increase the complexity of Grover's algorithm applied to AES in counter mode. Rather than the conventional approach of searching for K such that $AES_K(0^{128}) \oplus P = C$, an attacker can search for K such that $\mathrm{AES}_K^{-1}(P_0 \oplus C_0) \oplus \mathrm{AES}_K^{-1}(P_1 \oplus C_1) = 0^{127} \,||\, 1$. This approach remains effective even with nonce masking, see pages 28–29 of [3]. Similarly, many types of nonce hiding does not increase the complexity of nonce-collision attacks [4].

- We recommend that NIST enhance the introduction by providing a clear, user-friendly explanation of AEAD, hash functions, and XOFs, along with their practical applications. As demonstrated in [5], many developers, end-users, and individuals without a background in cryptography often find these fundamental concepts challenging to grasp.

- *"The family is developed to offer a viable alternative when the Advanced Encryption Standard (AES) may not perform optimally."*

  We think the specification should provide a detailed comparison of the specific benefits and drawbacks of Ascon relative to AES, focusing on aspects such as area, gate count, energy consumption, power efficiency, performance metrics, memory usage, code size, latency, security properties, and key agility. This would likely be valuable to a majority of readers. Although Ascon was designed for efficiency in constrained environments, our understanding is that it provides improved theoretical and practical security properties compared to many modes of AES-128 and SHA-256.

- The specification should explain that Ascon is designed with side-channel resistance in mind, highlighting the critical importance of side-channel protection. Side channels have resulted in numerous exploitable implementation vulnerabilities, both publicly known and undisclosed, and should be a requirement for all new cryptography.

- We welcome many of the changes NIST has made from the Ascon submission v1.2 [6], including allowing 16 rounds, updated initial values, little-endianness, truncation, and nonce masking.

- The terms width and capacity are never used in the specification. We suggest that the term width is removed as it does not add any value over the term state size. The specification should explain the actual numbers for state size, rate, and capacity in the beginning of the specification.

- *"The secret key K and the nonce-masking key K' (if available) shall be generated following the recommendations for cryptographic key generation specified in SP 800-133 [14] and using an approved random bit generator that supports at least a 128-bit security strength."*

  That there is no Ascon-based Key Derivation Function (KDF) as suggested in [7] severely limits the use cases for Ascon. As NIST states, the main feature of Ascon is that it can be used for multiple functionalities, which results in a compact implementation. Without an approved Ascon-based KDF, implementations will most likely use Ascon-CXOF128 as a KDF in a proprietary way. While such proprietary KDFs are likely secure, they may create interoperability challenges in the future. It seems unlikely that constrained systems will also implement Keccak, SHA-2, or AES just to derive keys for Ascon-AEAD128. We would prefer that NIST simultaneously publish specifications for Ascon-AEAD128 and an Ascon-based KDF to key the AEAD. The approach of first publishing Keccak-based hash functions and XOFs in FIPS 202 and then releasing a Keccak-based KDF a year later in SP 800-185 only worked because FIPS 202 exclusively contains unkeyed algorithms.

  The Ascon-based KDF should be secure even when the input secret is non-uniformly distributed, allow variable-length output, and offer collision and pre-image resistance even when the input secret is known [8]. We do not think that a separate Ascon-based DRBG is needed. A KDF has stronger security properties than a PRF, which in turn has stronger security properties than a DRBG or MAC. A single well-designed KDF should be enough for all use cases, but additional functions such as a MAC could be specified if there are significant performance benefits.

- The specification should make it clear that the APIs in Algorithms 3, 4, 5, 6, and 7 are just examples, and that Ascon can be implemented with streaming APIs where the length of the input (A, P, C, M, Z) and the output L are not necessarily available before Ascon is called.

- Appendix B describes how the 64-bit IV is constructed from the parameters in Table 12. We note that as a, b, t, and $r/8$ are uniquely determined by v, which is also included in the IV, there is little value in including a, b, t, and $r/8$ in the IV. Changing this could potentially lead to slightly more optimized implementations.

- The CCM specification SP 800-38C states that a protocol or application should protect against replay attacks, and it has been suggested that NIST should strengthen the recommendations [8]. We think replay protection should be a strong requirement unless careful analysis of the whole system shows that replay protection is not needed in some specific part. Users and developers expect replay protection and higher layer protocols are often designed with the expectation that the security protocol provides replay protection. Systems lacking replay protection are often vulnerable to unexpected attacks and challenging to analyze. If an upper layer was designed with the expectation of replay protection in a lower layer, using a security protocol without replay protection in the lower layers can compromise confidentiality, integrity, and availability in the higher layer, i.e., the whole infosec CIA triad. Practical and serious vulnerabilities due to the lack of replay protection have been common in both standardized and proprietary systems. The specification should strongly recommend that replay protection is used with Ascon-AEAD128.

AEAD:

- We welcome NIST's plan to only standardize a single AEAD based on Ascon.

- *"This section provides an option to implement Ascon-AEAD128 using a 256-bit key, mainly to maintain the 128-bit security strength of Ascon-AEAD128 in a multi-key setting"*

  We think it should be explained already in this section that the 256-bit key does not provide a 256-bit security strength in any setting.

- The specification should state that the nonce masking mechanism is not a nonce hiding transform [9] and state that the masked nonce $N \oplus K'$ shall be secret. A reader might otherwise believe that $N \oplus K'$ can be used as a public field in a security protocol.

- *"Nonce shall be distinct for each encryption operation for a given key to ensure that identical plaintexts encrypted multiple times produce different ciphertext."*

  The specification should describe the security of Ascon-AEAD128 with random nonces. Ascon-AEAD128 with 128-bit random nonces and a fixed key provide the same security as Ascon-AEAD128 with random 128-bit keys and a fixed nonce. This is true even if nonce hiding transforms are used [4].

- In addition to integrity strength, the specification should also describe if Ascon provides reforgeability resistance. This is an essential property, especially for short tags.

- We welcome that NIST uses the term "multi-key setting" instead of the outdated term "multi-user".

- *"The plaintext confidentiality of Ascon-AEAD128 is lost when a nonce is repeated with the same secret key."*

  This makes sense since Ascon-AEAD128 operates like a synchronous stream cipher on the first data block.

  "In the u-key setting, Ascon-AEAD128 with a λ-bit tag provides (min{128−log2(u), λ})-bit security strengths of confidentiality and integrity when a (nonce, associated data) pair is never repeated for two encryptions with each of u keys and the number of nonce repetitions per key for encryption is limited to 2^8"

  We don't see how this is true, considering that confidentiality is lost as soon as a nonce is repeated.

- *"In this scenario, the security strengths of Ascon-AEAD128 are summarized in Table 7."*

  Our interpretation of Table 7 is that it demonstrates the confidentiality against active attackers, which aligns with the integrity strength. However, we would also like to see confidentiality against passive attackers. for a more comprehensive analysis. In practical applications, passive and active attacks are often very different. For audio encryption applications, forgeries may not affect confidentiality.

- *"In the single-key setting, the attacker focuses on a specific key that is shared by one or more users. In contrast, in the multi-key setting with u keys, the attacker aims to compromise any of the u keys used by the users."*

  Multi-key does not imply more users than single-key. A single user can use many keys for protection at rest, and two users can use many keys for protection in transit. In fact, many practical use cases involve one key per encryption invocation, and security protocols between two users following best practices rekey often resulting in a large number of keys, and two users typically have many protocol sessions between them over time. Single-key is mostly a theoretical simplification that does not have much to do with practical security. One suggestion:

  "In the single-key setting, the attacker focuses on a specific key that is shared by one or more users. In contrast, in the multi-key setting with u keys, the attacker aims to compromise any of the u keys used by one or more user."

  But we would instead strongly suggest toning down the single-key scenario in general. Single-key seldom occur in practice and the single-key thinking should be phased out. Attackers will go for most bang-for-the-buck and will do whatever gives most benefit for the least amount of cost. The multi-key setting where the attacker tries to attack a single key is also a theoretical simplification. A multi-key setting (potentially with precomputation) where the attacker aims to compromise a large number of keys, is what most closely corresponds to actual practical attacks. The AT model appears to be a more realistic approach compared to other memory models [10]. Precomputation

can be seen as the cost of designing the VLSI chip. Ignoring the difference in value between various data, the important real-world property is the average cost per recovered plaintext byte.

- *"When the tag bit length is λ, $64 \leq \lambda \leq 128$, the maximum number of decryption failures for a fixed key shall be at most $2^{\lambda-64}$"*

  We welcome that the specification allows 64-bit tags, which is a strong requirement in many constrained use cases. However, availability is an essential part of any security systems (the A in the CIA triad) and enforcing a small limit of the number of decryption failures create denial-of-service problems and implies very low availability and robustness [11–12]. A low limit on decryption failures makes Ascon with 64-bit tags completely unusable in most applications. An application might derive a new key for each invocation, but then the requirement should be 1 invocation rather than 1 decryption failure.

  If Ascon with 64-bit tags is used for file encryption of several files, a single decryption failure would mean that the application is forbidden to decrypt any of the other files. It is hard to understand why NIST allows a constrained application to encrypt things with AES-CTR or AES-CBC but effectively forbids from using Ascon with 64-bit tags.

  We do not think SP 800-232 should be published with the current constraints for truncated tags.

- *"The key shall be updated to a new one when the total number of input data blocks or the number of decryption failures reach their respective limits or if the nonce uniqueness requirement is violated."*

  Why and how does the implementation keep track of nonce uniqueness requirements? On the encryption side the implementation should avoid nonce reuse instead of detecting nonce reuse and then rekeying. On the decryption side, the implementation should perform replay protection, allowing it to simply ignore nonce reuse.

  Doing rekeying too early before the confidentiality or integrity of the algorithm decreases significantly faster than linear typically increases the multi-key advantage for the session. The exact multi-key advantage depends on the algorithm but can be as much as u times its single-key advantage where u is the number of keys [11]. Does the rekeying based on decryption failures improve practical security in any way?

  Enforcing rekeying based on single-key advantages transform the setting to a multi-key setting, invalidating the single-key advantages [11]. The specification does not list the confidentiality advantages against passive attackers, integrity advantages in the single-key setting, or how a successful forgery affects the probability of subsequent forgeries. It is therefore unclear how NIST motivates the requirements and how they are supposed to improve practical security. Earlier NIST requirements for rekeying based on decryption failures have been based on the incorrect assumption that tag length is a good measure of security and the incorrect assumption that rekeying improves practical security [8].

  We do not think SP 800-232 should be published with the current rekeying requirements based on decryption failures and nonce uniqueness requirement.

Hash Functions and XOFs:

- We welcome that NIST standardizes a customized XOF. We believe that Ascon-CXOF128 is far more useful than Ascon-Hash256 and Ascon-XOF128, but our understanding is that Ascon-Hash256 and Ascon-XOF128 provide small performance advantages.

- *"This draft standard outlines the technical specifications of Ascon-AEAD128, Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128, and provides their security properties."*

  Our understanding is that Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 provide security against length-extension attacks as well as indifferentiability from a random oracle. The specification should describe these important security properties. Length-extension attacks have resulted in numerous exploitable implementation vulnerabilities.

- Table 9 lists the security strengths of Ascon-Hash256, Ascon-XOF128, and Ascon-CXOF128 as a function of the output length L. The security strength is also a function of the variable length message M. A random function whose input length is len(M) bits cannot provide more than len(M) security against preimage attacks. The preimage security is bounded by the Shannon entropy of the message M. If the message length is known or likely to be short, the preimage security is less than suggested in Table 9. The specification should explain this.

John Preuß Mattsson,
Expert Cryptographic Algorithms and Security Protocols

References

[1] IR 8547 Initial Public Draft
https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

[2] Ascon Team Comments on NIST IR 8547
https://csrc.nist.gov/files/pubs/ir/8547/ipd/docs/nist-ir-8547-ipd-comments-received.pdf

[3] The Multi-User Security of Authenticated Encryption: AES-GCM in TLS 1.3
https://eprint.iacr.org/2016/564.pdf

[4] Collision Attacks on Galois/Counter Mode (GCM)
https://eprint.iacr.org/2024/1111.pdf

[5] PQC Forum email discussion
https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/4_gSpmFccg8/m/eHzw9tkABgAJ

[6] Ascon submission v1.2
https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/round-2/spec-doc-rnd2/ascon-spec-round2.pdf

[7] Proposals for Standardization of the Ascon Family
https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/03-proposals-for-standardization-of-ascon-family.pdf

[8] Ericsson Comments on SP 800-38C
https://csrc.nist.gov/csrc/media/Projects/crypto-publication-review-project/documents/initial-comments/sp800-38c-initial-public-comments-2024.pdf

[9] “Nonces Are Noticed: AEAD Revisited”
https://eprint.iacr.org/2019/624.pdf

[10] Nice Attacks - but What is the Cost? Computational Models for Cryptanalysis
https://hal.science/hal-02306912v2/document

[11] Hidden Stream Ciphers and TMTO Attacks on TLS 1.3, DTLS 1.3, QUIC, and Signal
https://eprint.iacr.org/2023/913.pdf

[12] Robust Channels
https://eprint.iacr.org/2020/718.pdf

14. Arne Padmos, February 7, 2025

Dear NIST LWC team,

Thank you again for the great work in organising the lightweight crypto competition (aka "competition" or competition-like process) as well as the subsequent standards drafting process. Please find couple of comments below on the initial public draft of SP 800-232, focused around the topics identified in the draft as having changed from the Ascon v1.2 spec.

Regards,

Arne

I think it's good to have a new IV format and to have Ascon-p defined up to 16 rounds for potential functionality extensions. In NIST's last LWC presentation, the additional functionalities are flagged as 'PRF, MAC, KDF, DRBG etc.'. I understand that SP 800-232 won't contain a specific list of functionality to be standardised in the future, but it would be useful to get some idea of what's on the backlog and how the standardisation of that functionality will be approached. For example, for the SP 800-38 series there is an open continuing call for proposals and there is also the accordion mode effort currently ongoing (note that some of the suggested modes are instantiable with Ascon-XOF128).

As also noted in my comments on the decision proposal to revise SP 800-185, a two-pass mode for key wrapping and some way to support sessions would be interesting avenues for further exploration. In the meantime, one thing that should be included in SP 800-232 is a note that keys can be derived using Ascon-based hash and XOF variants. The SP 800-232 initial public draft makes reference to SP 800-133 Rev. 2, which references the withdrawn SP 800-108 for approved methods for symmetric key derivation from a pre-existing key. The updated SP 800-108r1-upd1 specifies different KDFs using CMAC as a PRF as well as a KDF using KMAC and a KDF using HMAC. As is, implementations of Ascon-AEAD128 might be hampered by having to implement HMAC for key derivation, let alone the overhead of CMAC or KMAC.

Relatedly, while I understand not wanting to spend large parts of the encoding space of IVs for user customisation, I still think that some kind of 'private IV space' would be useful – such as by making the upper 4 bits of 'v' configurable or by reserving another subset of the unique algorithm identifier for tailored variants (e.g. 0x80 to 0xFF). This can help reduce computational overhead and reduce the need for separate key derivation, storage, etc. etc. for situations like bi-directional channels. Also, this can make domain separation explicit between standard and tailored implementations. For example, implementations that decide to include zero padding and/or that make use of bit-interleaved input for better performance on 32-bit microcontrollers. Where needed, it would also allow implementers to clearly separate the output for different tag lengths, as well as allowing a distinction between raw and hashed customisation strings.

The switch from big-endian to little-endian is a logical one. Appendix A on implementation considerations and endianness is a great addition providing relevant background information. It might be helpful to more clearly highlight how multi-word values are read from or written to memory, such as a 128-bit key or a precomputed state. For figure 10, it could be useful to specify that the byte

values correspond to memory address offsets. Additionally, consider including references to implementation optimisation techniques like bit interleaving and state precomputation. Also, the standard should clarify how multi-word strings like hash and XOF outputs are to be printed such that a single canonical display format can be put in place. Even if it's not a full KAT suite, having at least a modicum of test vectors included in the standard – e.g. in a new appendix C – would help to make all of this more concrete for implementers.

It is wonderful to see that the XOFs are included in the draft standard. I do find it somewhat baffling that the hash variant is also included. The distinction between the hash versus XOFs is an artificial one. Adding the hash means more variants without good reason. Note that the Ascon-AEAD128 variant used with longer and shorter tags also doesn't get different IVs. If the problem is that previous NIST publications make reference to a hash as a function with a fixed output, then Ascon-Hash256 could simply be defined as a synonym of Ascon-XOF128 with a fixed output length in a given context, just like how Ascon-AEAD128 with different tag lengths is disambiguated through the use of different keys. Some guidance can be included to append the output length where explicit domain separation is necessary. (While reference could also be made to Ascon-CXOF128, as it's currently defined, Ascon-CXOF128 is less efficient than it could be for customisation strings less than or equal to 6 bytes. Also, the implementation complexity of Ascon-CXOF128 may by greater than simply xoring a domain separation bit to the state or prepending/appending a fixed-length string to the message. See how the Ascon-CXOF128 variant remains 'to be implemented' in the pyascon reference implementation.)

SP 800-232 ipd notes that 'the supported security strength [of Ascon-XOF128] is up to 128 bits' with 'Ascon-Hash256 [...] offering a security strength of 128 bits', but it doesn't mention that the mode of Ascon-Hash and Ascon-(C)XOF provides 192-bit preimage resistance (for output sizes greater than or equal to 192 bits) as detailed by Lefevre and Mennink in 'SoK: Security of the Ascon Modes' (https://eprint.iacr.org/2024/1969.pdf#subsection.8.3). It would be useful to understand the concrete security of Ascon-XOF128, especially because of the decision to go for the XOF variant with the larger security margin. Provided that the concrete security level is sufficient, the 'Preimage' column in table 9 can be updated with the values '192', 'min(L,192)', and 'min(L,192)'. If insufficient information is available to determine concrete security levels, then the caption of table 9 should be changed to 'Target security strengths'.

Nonce-masking is a very useful feature. The related warning regarding context commitment is appropriate, but it should be accompanied by a note that the context commitment of Ascon-AEAD128 is limited to a 64-bit security level. A higher level of 96 bits can be achieved through tailored zero padding with 64 zero bits, while a 128-bit security level can only be achieved using the Ascon-128 variant with 128-bit zero padding combined with fixing half of the nonce to zero or adding an initial key derivation step. More details can be found in 'Committing security of Ascon' by Naito et al. (https://doi.org/10.46586/tosc.v2023.i4.420-451).

There is a clear demand for 64-bit tags, even with all of the pitfalls involved. As to the limitations on the maximum number of decryption failures, it might be useful to make it very concrete what these

limitations mean for tags of length 64, 96, and 128 bits. Also, the requirement to update the key once 'the total number of input data blocks or the number of decryption failures reach their respective limits' is clear. However, how will the part of 'or if the nonce uniqueness requirement is violated' be enforced? Are implementations supposed to keep a tally of all nonces that have been generated?

15. Bishwajit, Mridul, Soumit, Thomas and Quan Quan, February 7, 2025

Dear NIST LwC committee,

We want to congratulate you on completing the first Public draft [1] on standardising ASCONAEAD128. Please find our comments on the draft below.

ASCONAEAD128.

In our opinion, it is a fair and clear winner of the LwC competition, and we believe that the construction will meet strong security requirements, which are much needed in modern lightweight devices. A large number of analyses on Ascon are available in the literature. We want to start by pointing out the pros and cons of ASCONAEAD128.

Pros:

1. As shown in [2], ASCONAEAD128 provides 64-bit CMT security, which can be extended with the use of injective padding on the message.

2. As shown in [3,4,5,6], the dual full-key binding property provides ASCONAEAD128 with a much higher AEAD security than any other sponge-type construction. Further, it resists authentication and key-recovery attacks even when an internal state is recovered.

Cons:

1. As discussed in [4] and also recognised in your draft, due to its small key size, ASCONAEAD128 is vulnerable when used in multi-user applications.

In [4], the authors also proposed a construction they called Ascon-256, which addresses the above-mentioned limitation of ASCONAEAD128. However, it has a con that it requires a separate dedicated construction and cannot be instantiated using ASCONAEAD128.

mu-ASCONAEAD128.

In your recent draft, recognising the limitation mentioned above, you have prescribed the use of a 256-bit key in the multi-user applications and also proposed a user instance on how to process these additional key bits while keeping the ASCONAEAD128 API as it is. Since you have not given any specific name for this user instance, henceforth, we refer to it as mu-ASCONAEAD128. With this nomenclature, we observe the Pros and cons of mu-ASCONAEAD128.

Pros:

1. It is fully compatible with ASCONAEAD128 API.

2. The AEAD security of mu-ASCONAEAD128 directly follows that of ASCONAEAD128, and hence, it maintains the same level of AEAD security guarantees.

3. Due to increased key size, the multi-user security of mu-ASCONAEAD128 overcomes the limitations of ASCONAEAD128.

Cons:

1. As observed in your draft, mu-ASCONAEAD128 has no committing security.

2. We have observed that the construction, although using a 256-bit key, only uses half of it for the "key binding", and thus, there is no security enhancement against key recovery under state recovery attacks. More specifically, one can still recover the whole 256-bit key in time 2^{128} given some internal state is recovered. Hence, the security essentially falls down to the hardness of recovering a state which has time complexity at most 2^{192} with exhaustive search. Here, we would like to point out that, with a large number of users using a large number of keys, the total data complexity can reach high values (even when the data limit per key is only 2^{50}). In such scenarios, the time complexity of recovering a single internal state of some user might become very low and thus, in the multi-user scenario, recovering the key of one of the users may become much smaller than 2^{192}. However, we don't have any concrete results on the overall complexity.

Our Proposed Variants.

Keeping the pros and cons of mu-ASCONAEAD128 in mind, we would like to propose two user instantiations that are compatible with Ascon API.

Both our instantiations are generated by tweaking the Ascon-256 construction in order to use its pros while eliminating the cons.

1. Our first construction, which we call Ascon-256.v2, like mu-ASCONAEAD128, uses a preprocessing step on (K, N, A, M) to generate the input of ASCONAEAD128.

2. Our second construction is called Ascon-256.v3, uses a preprocessing of (K, N, A, M) to generate the input of ASCONAEAD128 and also a post-processing of the ASCONAEAD128 output and can only be instantiated where the user has the control over modifying the ASCONAEAD128 output.

Both our instantiations use only a single extra permutation call in comparison to the mu-ASCONAEAD128. With this negligible extra cost in performance, we achieve the following advantages.

1. Both our constructions achieve the same security guarantees of mu-ASCONAEAD128.

2. Both our constructions are CMT-4 secure.

3. Ascon-256.v3 achieves full 256-bit key binding property.

Please find the draft attached to this email for more concrete and detailed explanations of all the above-mentioned observations.

References:

[1] Meltem Sonmez Turan, Kerry McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. Technical report, National Institute of Standards and Technology, 2024. Publication number 800 NIST SP 800-232 ipd

[2] Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Committing security of ascon: Cryptanalysis on primitive and proof on mode. IACR Transactions on Symmetric Cryptology, 2023(4):420–451, 2023.

[3] Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Exact security analysis of ASCON. In Jian Guo and Ron Steinfeld, editors, Advances in Cryptology – ASIACRYPT 2023, Part III, volume 14440 of Lecture Notes in Computer Science, pages 346–369, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8727-6_12

[4] Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Tight multi-user security of ascon and its large key extension. Lecture Notes in Computer Science, pages 57–76. Springer, Cham, Switzerland, December 1–3, 2024. doi:10.1007/978-981-97-5025-2_4.

[5] Charlotte Lefevre and Bart Mennink. Generic security of the ascon mode: On the power of key blinding. Cryptology ePrint Archive, Paper 2023/796, 2023. URL: https://eprint.iacr.org/2023/796.

[6] Charlotte Lefevre and Bart Mennink. SoK: Security of the Ascon Modes. Cryptology ePrint Archive, Paper 2024/1969, 2024. URL: https://eprint.iacr.org/2024/1969.

[attachment begins on next page]

AsconAEAD128 Revisited in the Multi-user Setting

Bishwajit Chakraborty[1], Mridul Nandi[2], Soumit Pal[2],
Thomas Peyrin[1], Quan Quan Tan[1]

[1] Nanyang Technological University, Singapore
{bishwajit.chakrabort@,thomas.peyrin@,quaanquan001@e.}ntu.edu.sg
[2] Indian Statistical Institute, Kolkata, India
{mridul.nandi, soumitpal378 }@gmail.com

Abstract. After more than half a decade since its initiation, NIST declared Ascon as the winner of the LwC competition. In the first public draft of AsconAEAD128, NIST recognized that Ascon has limitations when used in multi-user applications. To mitigate this, NIST prescribed the use of a 256-bit key in multi-user applications and produced an instantiation on how to process this extra key size in the current AsconAEAD128 API. While doing so, they identified a limitation of this new scheme (which we refer to as mu-Ascon in this document): mu-Ascon is vulnerable to committing attack and hence cannot be used in cases where committing security is required. On the other hand, the full key-binding property in Ascon, which separated it from other sponge-type constructions, has been used to show that Ascon is much stronger in the sense that it presents a key recovery resistance even in the case where some intermediate state is recovered. We remark that the current mu-Ascon has the limitation that only a partial key is bound during initialization and finalization. In this work, we propose some alternative instantiations of AsconAEAD128 API for multi-user applications. In comparison with the current mu-Ascon proposal, our first construction Ascon-256.v2 guarantees CMT-4 committing security up to 64 bits, and our second construction Ascon-256.v3 leads to both CMT-4 committing security and full 256-bit key binding. Structurally, our instantiations use only an extra-permutation call to provide these extra security features compared to mu-Ascon, which has a negligible overhead in terms of performance (given the lightweight nature of the Ascon permutation).

Keywords: Ascon, Multi-user Security, 256-bit Key, AEAD , Tight Security, Lightweight Cryptography

1 Introduction

Authenticated Encryption with Associated Data (AEAD) schemes are a fundamental class of symmetric-key encryption schemes that have been extensively studied. Over the past two decades, the demand for AEAD schemes suitable for

lightweight devices without compromising security has increased dramatically. To address this need, in 2018, the National Institute of Standards and Technology (NIST) issued a call for proposals [17] (known as the NIST Lightweight Cryptography (LwC) competition) for a lightweight AEAD scheme suitable for standardization. The first round received approximately 56 submissions, and after nearly half a decade of rigorous research and evaluation, NIST selected the Ascon-128a protocol as the winner. In [21], NIST renamed the scheme to AsconAEAD128 and published an initial public draft of it for comment. Ascon's security has been extensively analyzed for over a decade since its participation in the CAESAR competition, where it also won in the resource-constrained use case.

1.1 Existing Analysis on Ascon

In the provable security domain, most studies on Ascon have focused on conventional AEAD security, namely privacy and authenticity. Initially, Ascon was primarily considered a Duplex-type construction. Chakraborty et al. [2] and Lefevre et al. [10] independently were among the first to treat Ascon AEAD as a dedicated mode when proving security. They demonstrated that the double key binding present in both the initialization and finalization states provides Ascon AEAD with significantly stronger security compared to the Duplex construction. The authors in [2] showed that these additional key bindings provide Ascon AEAD security up to $D \ll 2^c, T \ll 2^c$ in the ideal permutation model, where D and T represent the data and time complexity of the attacker, respectively, and c is the rate size. Independently, Lefevre et al. [10] showed that, unlike the general Duplex construction, Ascon maintains authenticity security even under state-recovery attacks; that is, recovering some intermediate state does not lead to forgery or key recovery. They derived a tight security bound against forgery under state-recovery in nonce-misuse settings. Furthermore, they provided bounds for Ascon in the multi-user setting, which were later improved by Chakraborty et al. [3]. A dominant term appearing in both [10] and [3] in the multi-user setting is of the form $\mu T/2^\kappa$, where μ is the number of users and κ is the key size. Since the key size of both Ascon-128 and Ascon-128a is 128 bits, achieving a security level of $T = 2^{112}$ does not allow for a large number of users (μ). To address this limitation, the authors in [3] proposed a 256-bit key variant of Ascon, named Ascon-256, and demonstrated its security even with a large number of users.

Besides conventional AEAD security, modern AEAD schemes offer additional security guarantees such as related-key security (RKA), key-dependent message security (KDM), and context-committing security (CMT). Farshim et al. [8] introduced the concept of key-committing security, where an adversary aims to find two keys $K \neq K'$ and a ciphertext-tag pair (C, T) such that (C, T) is valid under both keys. It was quickly shown that standard AEAD security does not imply key-committing security, and popular schemes like GCM [9,6], GCM-SIV [12], CCM [7,13], and ChaCha20-Poly1305 [9,16] were found to be vulnerable. These attacks often had significant practical implications, making

committing security a critical issue. Bellare and Hoang [1] proposed generalized security notions where the adversary's goal is to find two different contexts $(K, N, A, M) = (K', N', A', M')$ that produce the same ciphertext-tag pair. They defined these notions as CMT-1, CMT-3, and CMT-4, where CMT-1 represents conventional key-committing security, CMT-3 requires $(K, N, A) \neq (K', N, A')$, and CMT-4 requires $(K, N, A, M) \neq (K', N', A', M')$. They also showed that CMT-3 is equivalent to CMT-4 and is strictly stronger than CMT-1. Consequently, designing CMT-4 secure AEAD schemes has become a crucial research area. In fact, NIST's workshop on updating block-cipher modes [18] explicitly mentioned that commitment security will be a mandatory feature for updated block ciphers. Recently, Naito et al. [15] studied the context-committing security of AsconAEAD128 and proved that it achieves CMT-4 security up to the birthday bound in tag size. They also suggested a method to enhance this security by padding the message with zeros.

1.2 AsconAEAD128 in Multi-User Applications

Recently, in their first public draft [21] for AsconAEAD128, NIST acknowledged its limitations in multi-user settings due to the $\mu T/2^{\kappa}$ term in the security bound. To mitigate this, NIST proposed using a 256-bit key in multi-user scenarios, with the key processed according to the initialization procedure described by Dobraunig et al. [5].

In Section 5.2, we demonstrate that this multi-user instance of AsconAEAD128 (hereafter referred to as mu-Ascon) is susceptible to complete key recovery under state-recovery attacks in $T = 2^{\kappa/2}$ and hence the primary time/data complexity of this key-recovery attack is determined by the complexity of the intermediate state-recovery attack. The current key-recovery security guarantees against this attack for mu-Ascon rely solely on the time/data complexity restrictions specified in [17,21]. Our current attack does not violate these restrictions. However, if future advanced attacks reduce state-recovery attack complexities below these limits, complete key recovery will become a real concern. The "power of key-binding" (as termed in [10]) of mu-Ascon will then be limited by the strength of the lower half of the key.

Here we remark that the Ascon-256 construction is resistant to this attack due to the full 256-bit key binding. Unfortunately, the current Ascon-256 mode cannot be instantiated with the AsconAEAD128 mode API.

Another disadvantage of mu-Ascon, is that it is only suitable for applications where committing security is not required. This was acknowledged in the draft itself [21] showing a trivial committing attack. While committing security was not an initial requirement in the original NIST call [17], we must argue, that it is a crucial security property for any modern and future-proof standardized AEAD scheme.

1.3 Our Contributions

Motivated by the newly introduced mu-Ascon with a 256-bit key, as described in the recent NIST public draft [21], this paper analyzes this new use case from a provable security perspective.

More specifically, in this draft, we deal with the two weaknesses exhibited by the current mu-Ascon instantiation. Namely,

1. having only partial security against key-recovery under state recovery attacks.

2. having no commitment security.

In this regard with small tweaks in the Ascon-256 construction by Chakraborty et al.[3], we generate two schemes which we call Ascon-256.v2 and Ascon-256.v3 for the multi-user settings, which are compatible with the AsconAEAD128 API. Ascon-256.v2 is able to resist CMT-4 attacks, while Ascon-256.v3 offers protection against both CMT-4 attacks and key-recovery under state recovery attacks. These improvements only come with a minimal performance cost: an additional Ascon permutation.

Table 1: Table comparing our provable security results in the ideal cipher model mu-Ascon, Ascon-256.v2 and Ascon-256.v3 use-cases in the nonce-respecting-multi-user settings. **Sr-Kr**:=Key-recovery under state recovery ; T is measured in the unit of number of permutation calls.

security-type	**Variant**	Adversarial-advantage
Sr-Kr	mu-Ascon/ Ascon-256.v2	$\geq \frac{T}{2^{128}}$ [section 5.2]
CMT-4	mu-Ascon	1 [21]
CMT-4	Ascon-256.v2	$\leq \frac{T^2}{2^{128}}$ [section 5.1].
CMT-4	Ascon-256.v3	$\leq \frac{T^2}{2^{128}}$ [section 5.1].

To prove the committing security we simply reuse the committing security of AsconAEAD128 result by Naito et al. [15]. Given the lightweight nature, since our Ascon-256.v2 and Ascon-256.v3 use cases require only one extra permutation call per query, hence an efficiency difference between it and the current use-case is negligible compared to the amount of extra security guarantees it provides.

2 Preliminaries

2.1 Notations

Let $\{0,1\}^n$ represent the set of bit strings of length n, and $\{0,1\}^+$ denote the set of bit strings of arbitrary length. The empty string is denoted by λ, and we define $\{0,1\}^* = \{\lambda\} \cup \{0,1\}^+$. For any integers $a \leq b \in \mathbb{N}$, $[b]$ and $[a,b]$ denote

the sets $\{1, 2, \ldots, b\}$ and $\{a, a+1, \ldots, b\}$, respectively. For $n, k \in \mathbb{N}$ with $n \geq k$, the falling factorial is defined as $(n)_k := n(n-1)\cdots(n-k+1)$. It's worth noting that $(n)_k \leq n^k$.

For any bit string $x = x_1 x_2 \cdots x_k \in \{0,1\}^k$ of length k, and for $n \leq k$, we use $\lceil x \rceil_n := x_1 \cdots x_n$ (and $\lfloor x \rfloor_n := x_{k-n+1} \cdots x_k$) to denote the most (and least) significant n bits of x. The bit concatenation operation is denoted by $\|$. The notation $(x_1, \ldots, x_r)$ is also used to represent the bit concatenation operation $x_1 \| \cdots \| x_r$, where $x_i \in \{0,1\}^*$. For instance, if $V := x \| z := (x, z) \in \{0,1\}^r \times \{0,1\}^c$, then $\lceil V \rceil_r = x$ and $\lfloor V \rfloor_c = z$. The bitwise XOR operation is denoted by $\oplus$.

For a finite set $\mathcal{X}$, $\mathsf{X} \xleftarrow{\$} \mathcal{X}$ denotes the uniform and random sampling of X from $\mathcal{X}$, and $\mathsf{X} \xleftarrow{\mathsf{wor}} \mathcal{X}$ denotes sampling without replacement of X from $\mathcal{X}$.

Padding and Parsing a Bit String. Let $r > 0$ be an integer and $X \in \{0,1\}^*$. Let $d = |X| \bmod r$ (the remainder while dividing $|X|$ by r).

$$\mathsf{pad}(X) = X \| 1 \| 0^{r-1-d}.$$

Given $X \in \{0,1\}^*$, let $x = \lceil \frac{|X|+1}{r} \rceil$. We define $(X_1, \ldots, X_x) \xleftarrow{r}_* X$ as $X_1 \| \cdots \| X_x = X$, $|X_1| = \cdots = |X_{x-1}| = r$ and

$$X_x = \begin{cases} \lambda & \text{if } |X| = r(x-1) \\ \lfloor X \rfloor_{|X|-r(x-1)} & \text{otherwise} \end{cases}.$$

For $N \geq 4,\ n = \log_2 N$, we define

$$\mathsf{mcoll}(q, N) = \begin{cases} 3 & \textit{if } 4 \leq q \leq \sqrt{N} \\ \frac{4 \log_2 q}{\log_2 \log_2 q} & \textit{if } \sqrt{N} < q \leq N \\ 5n \left\lceil \frac{q}{nN} \right\rceil & \textit{if } N < q. \end{cases}$$

2.2 Authenticated Encryption with Associated Data: Definition and Security Model

An authenticated encryption scheme with associated data functionality, abbreviated as AEAD, is characterized by a tuple of algorithms $\mathsf{AE} = (\mathsf{Enc}, \mathsf{Dec})$. These algorithms, referred to as the encryption and decryption algorithms, operate over the *key space* $\mathcal{K}$, *nonce space* $\mathcal{N}$, *associated data space* $\mathcal{A}$, *message space* $\mathcal{M}$, *ciphertext space* $\mathcal{C}$, and *tag space* $\mathcal{T}$. The functionalities are defined as follows:

$$\mathsf{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C} \times \mathcal{T} \quad \text{and} \quad \mathsf{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\mathsf{rej}\}.$$

Here, rej signifies that the tag-ciphertext pair is invalid and consequently rejected. Additionally, the correctness condition is imposed:

$$\mathsf{Dec}(K, N, A, \mathsf{Enc}(K, N, A, M)) = M \text{ for any } (K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}.$$

For a key $K \in \mathcal{K}$, we use $\mathsf{Enc}_K(\cdot)$ and $\mathsf{Dec}_K(\cdot)$ to denote $\mathsf{Enc}(K, \cdot)$ and $\mathsf{Dec}(K, \cdot)$, respectively. In this paper, we consider $\mathcal{K} = \{0,1\}^\kappa, \mathcal{N} = \{0,1\}^\nu, \mathcal{T} = \{0,1\}^\tau$, and $\mathcal{A}, \mathcal{M} = \mathcal{C} \subseteq \{0,1\}^*$.

AEAD Security in the Random Permutation Model.
Let $\mathsf{Perm}(b)$ denote the set of all permutations over $\{0,1\}^b$ and $\mathsf{Func}(\mathcal{N} \times \mathcal{A} \times \mathcal{M}, \mathcal{M} \times \mathcal{T})$ denote the set of all functions from (N, A, M) to (C, T) such that $|C| = |M|$. We consider the AEAD security in the multi-user (mu) setting, parameterized by the number of users μ. Let:

- $\Pi \xleftarrow{\$} \mathsf{Perm}(b)$ (we use the superscript $\pm$ to denote bidirectional access to Π),
- $\Gamma_1, \ldots, \Gamma_\mu \xleftarrow{\$} \mathsf{Func}(\mathcal{N} \times \mathcal{A} \times \mathcal{M}, \mathcal{M} \times \mathcal{T})$,
- rej denotes the degenerate function from $(\mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$ to $\{\mathsf{rej}\}$, and
- $K_1, \ldots, K_\mu \xleftarrow{\$} \mathcal{K}$.

We have the following definition:

Definition 1. *Let AE_Π be an AEAD scheme based on the random permutation Π, defined over $(\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{T})$. The mu-AEAD advantage of an adversary $\mathscr{A}$ against AE_Π is defined as*

$$\mathbf{Adv}^{\mathsf{mu-aead}}_{\mathsf{AE}_\Pi}(\mathscr{A}) := \left| \Pr_{\substack{(\mathsf{K_i})_{i=1}^{\mu} \xleftarrow{\$} \mathcal{K} \\ \Pi^{\pm}}} \left[\mathscr{A}^{(\mathsf{Enc}_{\mathsf{K_i}}, \mathsf{Dec}_{\mathsf{K_i}})_{i=1}^{\mu}, \Pi^{\pm}} = 1 \right] - \Pr_{\substack{(\Gamma_i)_{i=1}^{\mu} \\ \Pi^{\pm}}} \left[\mathscr{A}^{(\Gamma_i)_{i=1}^{\mu}, \mathsf{rej}, \Pi^{\pm}} = 1 \right] \right|.$$

Here $\mathscr{A}^{\mathsf{Enc}_{\mathsf{K_i}}, \mathsf{Dec}_{\mathsf{K_i}}, \Pi^{\pm}}$ denotes $\mathscr{A}$'s response after its interaction with $\mathsf{Enc}_{\mathsf{K_i}}$, $\mathsf{Dec}_{\mathsf{K_i}}$, and $\Pi^{\pm}$ (i.e., both forward and backward queries to Π) respectively. Similarly, $\mathscr{A}^{\Gamma_i, \mathsf{rej}, \Pi^{\pm}}$ denotes $\mathscr{A}$'s response after its interaction with Γ_i, rej, and $\Pi^{\pm}$ respectively.

In this paper, we assume that the adversary is adaptive. This means that the adversary neither issues duplicate queries nor requests information for which the response is already known due to some previous query. Let q_e, q_d, and q_p represent the number of queries made across all $\mathsf{Enc}_{\mathsf{K_i}}$, all $\mathsf{Dec}_{\mathsf{K_i}}$, and $\Pi^{\pm}$, respectively. Furthermore, let σ_e and σ_d denote the sum of input lengths (including associated data and message) across all encryption and decryption queries, respectively. Additionally, let $\sigma := \sigma_e + \sigma_d$ represent the combined resources for construction queries.

Remark 1. Here σ corresponds to the online or data complexity, and q_p corresponds to the offline or time complexity of the adversary. An adversary adhering to the specified resource constraints is referred to as an $(q_p, \sigma_e, \sigma_d)$-adversary.

2.3 H-coefficient Technique

Consider an adversary $\mathscr{A}$, which is deterministic and computationally unbounded, attempting to distinguish between the real oracle, denoted as $\mathcal{O}_{\mathrm{re}}$, and the ideal oracle, denoted as $\mathcal{O}_{\mathrm{id}}$. The interaction of $\mathscr{A}$ with its oracle is captured by the query-response tuple denoted as ω. In certain scenarios, after the query-response phase of the game, the oracle may choose to reveal additional information to

the distinguisher. In such cases, the extended definition of the transcript may include that additional information. Let Θ_{re} (respectively, Θ_{id}) represent the random transcript variable when $\mathscr{A}$ interacts with $\mathcal{O}_{\text{re}}$ (respectively, $\mathcal{O}_{\text{id}}$). The probability of realizing a specific transcript ω in the security game with an oracle $\mathcal{O}$ is referred to as the *interpolation probability* of ω with respect to $\mathcal{O}$. Given the determinism of $\mathscr{A}$, this probability depends solely on the oracle $\mathcal{O}$ and the transcript ω. A transcript ω is considered *realizable* if $\Pr[\Theta_{\text{id}} = \omega] > 0$. In this paper, $\mathcal{O}_{\text{re}} = (\mathsf{Enc_K}, \mathsf{Dec_K}, \Pi^{\pm})$, $\mathcal{O}_{\text{id}} = (\Gamma, \mathsf{rej}, \Pi^{\pm})$, and the adversary aims to distinguish $\mathcal{O}_{\text{re}}$ from $\mathcal{O}_{\text{id}}$ in an AEAD sense.

Proposition 1 (H-coefficient technique [19,20]). *Let Ω be the set of all realizable transcripts. For some $\epsilon_{\mathsf{bad}}, \epsilon_{\mathsf{ratio}} > 0$, suppose there is a set $\Omega_{\mathsf{bad}} \subseteq \Omega$ satisfying the following:*

- $\Pr[\Theta_{\text{id}} \in \Omega_{\mathsf{bad}}] \leq \epsilon_{\mathsf{bad}}$;
- *For any $\omega \notin \Omega_{\mathsf{bad}}$,*

$$\frac{\Pr[\Theta_{\text{re}} = \omega]}{\Pr[\Theta_{\text{id}} = \omega]} \geq 1 - \epsilon_{\mathsf{ratio}}.$$

Then for any adversary $\mathscr{A}$, we have the following bound on its AEAD distinguishing advantage:

$$\mathbf{Adv}^{\mathsf{aead}}_{\mathcal{O}_{\text{re}}}(\mathscr{A}) \leq \epsilon_{\mathsf{bad}} + \epsilon_{\mathsf{ratio}}.$$

A proof of Proposition 1 can be found in multiple papers including [20,4,14].

2.4 Revisiting Committing Security of Ascon Modes

In this section we informally revisit the commitment security of Ascon mode as proved by Naito et al [15].

for any non-negative integer z, with the initialization/finalization permutation P_1 and intermediate permutation P_2 define

$$\mathsf{AsconAEAD128}^{[P_1,P_2]}_{ZP}.\mathsf{Enc}(K,N,A,M) = \mathsf{AsconAEAD128}^{[P_1,P_2]}.\mathsf{Enc}(K,N,A,M\|0^z).$$

Theorem 1. *[15] Let P_1 and P_2 be independent random permutations. For any CMT-4 adversary making a total of T queries to P_1,P_1^{-1}, P_2, or P_2^{-1}, we have*

$$\mathbf{Adv}^{\text{cmt}-4}_{\mathsf{Ascon}_{ZP}} \leq \left(\frac{11T^2}{2^c} + \frac{5T^2}{2^{n-\nu}} + \frac{0.5T^2}{2^{\tau+z}} + \frac{0.5T^2}{2^{n+\tau-\kappa-\nu}}\right) \cdot \left(1 - \frac{0.5T^2}{2^n}\right).$$

where, ν, κ, τ, c, n denote the nonce-size, key-size, tag-size , capacity-size and permutation state-size respectively.

Assuming $T < 2^{n/2}$, the term $\left(1 - \frac{0.5T^2}{2^n}\right)$ is $\mathcal{O}(1)$. Then assuming $\kappa \geq \tau$, the above bound shows that Ascon_{ZP} is CMT-4 secure as long as $T \ll \min\{2^{\frac{c}{2}}, 2^{\frac{\tau+z}{2}}, 2^{\frac{n+\tau-k-\nu}{2}}\}$, ensuring $\min\{\frac{c}{2}, \frac{\tau+z}{2}, \frac{n+\tau-k-\nu}{2}\}$ bit CMT-4 security.

Corollary 1. *In the random permutation model,*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\mathsf{AsconAEAD128}} = \mathcal{O}\left(\frac{T^2}{2^{128}}\right).$$

Proof. The CMT-4 security of AsconAEAD128 follows from theorem 1 by plugging in the actual parameter sizes and the observation that AsconAEAD128 is $\mathsf{AsconAEAD128}_{ZP}$ with $z = 0$.

3 Existing Ascon Proposals in the Multi-User Settings

Consider the AsconAEAD128 design in the multi-user settings. due to the term $\frac{\mu T}{2^\kappa}$ appearing in the security bound [10,3], the standard key size of 128-bits doesn't leave room for a large μ (number of users). In this section we revisit existing Ascon-based constructions which are tailor-made to deal with large number of users. More specifically we revisit the modes of operations defined in [21,3].

3.1 mu-Ascon

In the recently published public draft [21], NIST acknowledged the issue of having a small key size for the use case where there are a large number of users. To tackle this they used the masked nonce initialization in the Duplex paradigm introduced by Dobraunig et al. [5]. More specifically in the multi-user setting each user has a 256-bit key and given a encryption tuple (N, A, M), an user uses its key $K := K_1 \| K_2$ in the AsconAEAD128 construction to get the ciphertext-tag pair as follows:

$$\mathsf{mu\text{-}Ascon}(K_1 \| K_2, N, A, M) := \mathsf{AsconAEAD128.Enc}(K_1, N \oplus K_2, A, M)$$

The advantages of this specific use instance is that it can be instantiated directly with theAsconAEAD128 API and doesn't compromise the performance.

The disadvantage of this construction is that although it doesn't violet the security requirement prescribed in [17], as shown in section 5.2 this use case is prone to key-recovery under state recovery attack and in no case provide > 192-bit key-recovery security.

Furthermore, it doesn't provide any committing security. For any two pair $(K, N) \neq (K', N')$ such that $K \oplus 0^{128} \| N = K' \oplus 0^{128} \| N'$ and any (A, M) this outputs same (C, T) pair. Here we argue that, although the committing security was not put in as an initial requirement in [17], it is a must-have security for a future ready standardized AEAD scheme.

3.2 Ascon-256

Another construction that is tailor-made for multi-user settings with a large number of users was introduced by Chakraborty et al. [3] called the Ascon-256 construction. On the high level the construction is as follows:

INITIALIZATION : $V_0 = P(K\|IV) \oplus 0^{64}\|K$.
DATA PROCESSING : given nonce N and associated data A, and message M define $A' := N\|A$, Process A', M using the AsconAEAD128 associated data and message processing protocols.
FINALIZATION : in the finalization process xor $K\|0^{64}$ to the input of the permutation and add $\lfloor K \rfloor_\tau$ o the output of the permutation to generate the tag.

The multi-user settings construction security follows the bounds achieved in [3].

The disadvantage of Ascon-256 is that in it's present form, it cannot be directly instantiated using the AsconAEAD128 API due to the addition of full 256 bit K binding to the output (resp. input) of the initialization (resp. finalization) permutation.

Remark 2. The commitment security of Ascon-256 is still an open problem but a quick read of the proof in [15], we think it should follow in the same line as AsconAEAD128. Since our objective is to come-up with a construction which is compatible to AsconAEAD128 API, we declare it as out of scope of this paper.

4 Ascon-256.v2, Ascon-256.v3: Alternate Proposals for Multi-User Settings

In this section we modify the Ascon-256 construction so as to make it compatible with the AsconAEAD128 API. Our main objective is to construct use-cases of AsconAEAD128 which can resists against the weaknesses of mu-Ascon i.e. they should be resistant to

- CMT-4 attack
- key recovery against state-recovery attack.

In this respect we define two new use-instances which we call Ascon-256.v2 and Ascon-256.v3 constructions depending on how the AsconAEAD128 API is processed. More specifically,

- we define Ascon-256.v2 for the use-case when the user can only pre-processes her input before feeding it in the AsconAEAD128.
- we define Ascon-256.v3 for the use-case when the user can also process the AsconAEAD128 output before releasing it.

We present our constructions using the AsconAEAD128 construction as a black-box. We start by defining two functions,

$$F_{pre} : \{0,1\}^{256} \times \{0,1\}^{128} \times \{0,1\}^* \to \{0,1\}^{128} \times \{0,1\}^{128} \times \{0,1\}^*$$

$$(K_1\|K_2, N, A) \mapsto (K_2, K_1, A_1)$$

where $A_1 := (N \oplus \lceil K_1 \rceil_{128})\|A$.

$$F_{post} : \{0,1\}^{128} \times (\{0,1\}^* \cup \perp) \rightarrow (\{0,1\}^* \cup \perp)$$

$$(K, \perp) \mapsto \perp$$

$$(K, M) \mapsto (M \oplus 0^{|M|-128} \| K) \text{ if } |M| \geq 128$$

$$(K, M) \mapsto ((M \oplus \lceil K \rceil_{|M|})) \text{ if } |M| \leq 128$$

Definition 2. *(Ascon-256.v2)*

$\mathsf{Ascon\text{-}256.v2.Enc}(K_1 \| K_2, N, A, M) = \mathsf{AsconAEAD128.Enc}(F_{pre}(K_1 \| K_2, N, A), M)$

$\mathsf{Ascon\text{-}256.v2.Dec}(K_1 \| K_2, N, A, C, T) = \mathsf{AsconAEAD128.Dec}(F_{pre}(K_1 \| K_2, N, A), C, T)$

Definition 3. *(Ascon-256.v3)*

$$\mathsf{Ascon\text{-}256.v3.Enc}(K_1 \| K_2, N, A, M) = (F_{post}(K_1, C), T)$$

where $(C, T) = \mathsf{AsconAEAD128.Enc}(F_{pre}(K_1 \| K_2, N, A), F_{post}(K_1, M))$.

$$\mathsf{Ascon\text{-}256.v3.Enc}(K_1 \| K_2, N, A, M) = (F_{post}(K_1, C), T)$$

where $(C, T) = \mathsf{AsconAEAD128.Enc}(F_{pre}(K_1 \| K_2, N, A), F_{post}(K_1, M))$.

$$\mathsf{Ascon\text{-}256.v3.Dec}(K_1 \| K_2, N, A, C, T) = F_{post}(K_1, M)$$

where $M = \mathsf{AsconAEAD128.Dec}(F_{pre}(K_1 \| K_2, N, A), F_{post}(K_1, C), T)$

In the following section we compare different security notions for AsconAEAD128, Ascon-256 and our new variants Ascon-256.v2, Ascon-256.v3.

5 Comparative Security Analysis of mu-Ascon, Ascon-256.v2 and Ascon-256.v3

In this section, we first try to do a comparative security analysis of mu-Ascon, Ascon-256.v2 and Ascon-256.v3 in the nonce-respecting multi-user settings under different security notions. More specifically the CMT-4 security, key-recovery under state recovery security of these constructions. To the best of our knowledge the AEAD security of mu-Ascon has never been formally explored. In their draft NIST [21] justify the AEAD security of mu-Ascon with reference to the results in [5,3]. In this respect we will argue that Ascon-256 in [3], has a full key-binding rather than the partial key binding in mu-Ascon and the analysis in [5] was explored on the Duplex rather than Ascon mode. Hence, for the sake of completeness later in section 7 we do a revisit of the proofs in [3] with the necessary adjustment needed in the analysis of mu-Ascon, Ascon-256.v2 and Ascon-256.v3.

5.1 CMT-4 Security of Ascon-256.v2 and Ascon-256.v3

In this section, we show that unlike mu-Ascon, our variants namely Ascon-256.v2 and Ascon-256.v3 both enjoy commitment security.

Theorem 2. *For any CMT-4 adversary $\mathscr{A}$,*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\textit{Ascon-256.v2}}(\mathscr{A}) \leq \mathbf{Adv}^{\mathrm{cmt}-4}_{\textit{AsconAEAD128}}(\mathscr{A}).$$

Proof. Suppose there exists an adversary $\mathscr{A}$ who can break the CMT-4 security of Ascon-256.v2. Let $\mathscr{A}$ outputs $(K_1\|K_2, N, A, M) \neq (K_1'\|K_2', N', A', M')$ such that

$$\textsf{Ascon-256.v2.Enc}(K_1\|K_2, N, A, M) = \textsf{Ascon-256.v2.Enc}(K_1'\|K_2', N', A', M').$$

Note that F_{pre} is an injective function. Hence

$$(F_{pre}(K_1\|K_2, N, A), M) \neq (F_{pre}(K_1'\|K_2', N', A'), M')$$

but

$$\textsf{AsconAEAD128.Enc}((F_{pre}(K_1\|K_2, N, A), M)) = \textsf{AsconAEAD128.Enc}((F_{pre}(K_1'\|K_2', N', A'), M')).$$

Hence the adversary $\mathscr{A}$ breaks the CMT-4 security of AsconAEAD128 by outputting $(F_{pre}(K_1\|K_2, N, A), M) \neq (F_{pre}(K_1'\|K_2', N', A'), M')$.

Corollary 2. *In the random permutation model,*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\textit{Ascon-256.v2}} = \mathcal{O}\left(\frac{T^2}{2^{128}}\right).$$

Proof. The corollary follows from theorems 1 and 2.

Theorem 3. *For any CMT-4 adversary $\mathscr{A}$ in the random permutation model,*

$$\mathbf{Adv}^{\mathrm{cmt}-4}_{\textit{Ascon-256.v3}} = \mathcal{O}\left(\frac{T^2}{2^{128}}\right).$$

Proof. The proof of this theorem can be proved with a similar approach as taken by Naito et al. to prove the CMT-4 security of AsconAEAD128. We provide the proof details in section 6.

5.2 Key-Recovery from State-Recovery Attack on mu-Ascon and Ascon-256.v2 in Nonce-Respecting Multi-User Settings

We start by revisiting the security model defined by Lefevre et al. [11] which they call the authenticity under state recovery. The authors formally define the security model as follows.

Definition 4. *[11] Consider an adversary $\mathscr{A}$ with access to two learning oracles $\mathcal{LE}$ and $\mathcal{LD}$, which are defined as $\mathcal{E}$ and $\mathcal{D}$ but that additionally leak all input/output values of the evaluations of the inner permutations. We say that $\mathscr{A}$ wins if it ever makes a query to one of its learning decryption oracles that is successful and that is not the result of an earlier encryption query. This leads to the following model:*

$$\mathbf{Adv}^{\mathrm{sr-Auth}}_{\mathit{Ascon}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathcal{LE},\mathcal{LD}} \textit{ forges}\right].$$

In this section we extend the security model defined by Lefevre et al. [11] and call it key-recovery under state recovery. The formal definition of the security model in the multi-user settings is a simple extension of definition 4 and is as follows.

Definition 5. *Consider an adversary $\mathscr{A}$ with access to two learning oracles $\mathcal{LE}_u$ and $\mathcal{LD}_u$ which are defined as $\mathcal{E}_u$ and $\mathcal{D}_u$ for each of the users u such that for some user u, they additionally leak all input/output values of the evaluations of their inner permutations. We say that $\mathscr{A}$ wins if it can recover the entire secret-key for one such user u:*

$$\mathbf{Adv}^{\mathrm{mu-sr-kr}}_{\textit{mu-Ascon},\textit{Ascon-256.v2},\textit{Ascon-256}}(\mathscr{A}) = \Pr\left[\mathscr{A}^{\mathcal{LE},\mathcal{LD}} \textit{recovers the key for some user } u\right].$$

Now we explore the Key-recovery under state-recovery attack on mu-Ascon and Ascon-256.v2. As the definition suggests the objective of the adversary is to recover the entire secret-key of one of the users given an intermediate state has been recovered for some encryption query made to the user. Note that, since the intermediate states are processed using the duplex construction hence if for some query with σ_e blocks of data for an user any of the intermediate state is recovered then the state after the initialization and state-before the finalization can be recovered in time complexity σ_e.

Theorem 4. *Given some and thus all intermediate states of some mu-Ascon encryption query by one of the users is know. Then there exists a nonce-respecting adversary $\mathscr{A}$ who can recover the full 256-bit key of that user in $q << 2^{129}$ queries. i.e.*

$$\mathbf{Adv}^{\mathrm{mu-sr-kr}}_{\textit{mu-Ascon}}(\mathscr{A}) \geq \frac{q}{2^{129}}$$

Proof. The attack is straight forward. But for the sake of completeness we define how such an adversary $\mathscr{A}$ can work.

- $\mathscr{A}$knows the X_1, Y_l, and T where X_1 is the full intermediate state after the initialization, Y_l is the last intermediate state before the finalization and T is the tag .
- $\mathscr{A}$makes 2^{128} many permutation queries of the form $P2(Y_l \oplus 0^{128} \| K_{i,1} \| 0^{64})_{i \in [2^{128}]}$ and receives responses of the form Z_i.

– For each Z_i, $\mathscr{A}$checks if $\lfloor Z_i \rfloor_{128} \oplus K_{i,1} = T$.
– for all Z_i such the $\mathscr{A}$gets a matching, it makes one inverse permutation query of the form $(P2)^{-1}(X_l \oplus 0^{192} \| K_{i,2})_{i \in [2^{128}]}$ and receives responses of the form W_i and checks if $\lfloor W_i \rfloor_{192} = IV \| K_{i,2}$.
– If the above check is successful the adversary define $K_{i,1} = N \oplus \lceil W_i \rceil_{128}$.
– $\mathscr{A}$then verifies if $K_{i,1} \| K_{i,2}$ is the key for the user by making an empty data query to the user and matching the tag received with the tag computed by herself using $K_{i,1} \| K_{i,2}$ as the key.

If the number of $K_{i,2}$ such that $\lfloor Z_i \rfloor_{128} \oplus K_{i,1} = T$ is t. Then the number of total permutation and inverse permutation queries required is at most $2^{128} + t \ll 2^{129}$.

Theorem 5. *Given some and thus all intermediate states of some Ascon-256.v2 encryption query by one of the users is know. Then for any nonce-respecting adversary can recover the full 256-bit key of that user in $q << 2^{129}$ queries. i.e.*

$$\mathbf{Adv}^{\mathrm{mu-sr-kr}}_{\textit{Ascon-256.v2}}(\mathscr{A}) \geq \frac{q}{2^{129}}$$

Proof. Consider an adversary $\mathscr{A}$ which recovers the intermediate states for some user. For that user,

– $\mathscr{A}$ knows the X_1 and Y_l and τ where X_1 is the full intermediate state after the initialization, Y_l is the last intermediate state before the finalization and T is the tag .
– $\mathscr{A}$ makes T many permutation queries of the form $P2(Y_l \oplus 0^{128} \| K_{i,2} \| 0^{64})_{i \in [2^{128}]}$ and receives responses of the form Z_i.
– For each Z_i, $\mathscr{A}$checks if $\lfloor Z_i \rfloor_{128} \oplus K_{i,1} = \tau$.
– Once a $K_{i,2}$ is identified it makes T inverse permutation queries of the form $(P2)^{-1}(X_l \oplus (N \oplus K_{i,2}) \| 0^{64} \| K_{i,2})_{i \in [2^{128}]}$ and receives responses of the form W_i and checks if $W_i = K_{i,1} \| IV \| K_{i,2}$.
– $\mathscr{A}$If the above check is successful, then $\mathscr{A}$ verifies if $K_{i,1} \| K_{i,2}$ is the key for the user by making an empty data query to the user and matching the tag received with the tag computed by herself using $K_{i,1} \| K_{i,2}$ as the key.

If we assume $P2$ to be ideal random permutation then the expected number of $K_{i,2}$ such that $\lfloor Z_i \rfloor_{128} \oplus K_{i,1} = T$ is 1. Then the number of total permutation and inverse permutation queries required is at most 2^{129}.

6 Proof of Theorem 3

Suppose the adversary makes q permutation queries (forward/backward) to the ideal world oracles $\mathcal{O}$. The oracle $\mathcal{O}$ chooses a random permutations P. Each query (forward/backward) is responded by the oracle $\mathcal{O}$ using this random permutation. Let $\mathcal{F} := \{(X_i, Y_i) \mid P(X_i) = Y_i; i \in [q]\}$ denotes the list generated by the adversarial queries. We define the ordering $\prec$ on $\mathcal{F}$ such that given any

pair $\{(X_i, Y_i) \neq (X_j, Y_j)\} \in \mathcal{F}$, $(X_i, Y_i) \prec (X_j, Y_j)$ if and only if the query corresponding to X_i, Y_i occurred before the query corresponding to (X_j, Y_j). We start by defining a Bad event generated due to $\mathcal{F}$.

We say the adversary forms a "full sequence" $S = \{(X_i, Y_i) \mid i \in [0, t_S]\} \subseteq \mathcal{F}$ if it represents the input/output states of the underlying permutation calls of some valid decryption query

$$\mathsf{Ascon\text{-}256.v3.Dec}((K_1 \| K_2, N, A, C, T)) \neq \perp$$

Given any sequence S define;

$$K_S := \lceil X_0 \rceil_{128}; T_S := \lceil Y_t \rceil_{128} \oplus K_S;$$

It is easy to note that for any full sequence S, the following equations holds.

(1) $\lfloor X_0 \rfloor_{64} = IV$.
(2) $\forall i \in [1, t_S - 1]$, $\lceil Y_i \rceil_{192} \oplus \lceil X_{i+1} \rceil_{192} \in \{0^{192}, 0^{191} \| 1\}$.
(3) $\lceil Y_0 \rceil_{192} \oplus \lceil X_1 \rceil_{192} = 0^{64} \| K_S$.
(4) $\lceil Y_{T_S - 1} \rceil_{192} \oplus \lceil X_{T_S} \rceil_{192} \in \{K_S \| 0^{64}, K_S \| 0^{63} \| 1\}$.

Proposition 2. *Suppose an adversary breaks the CMT-4 security game against Ascon-256.v3. Then there exists two distinct full sequences S, S' such that $T_S = T_{S'} = T$ and they both correspond to the same cipher text.*

Given any full sequence $S = \{(X_i, Y_i) \mid i \in [0, t_S]\} \in \mathcal{F}$, we call $S_{in} = \{(X_i, Y_i) \mid i \in [0, t_S - 1]\}$ the internal sequence of S. We define some bad events while generating some internal sequence S_{in} using $\mathcal{F}$. For any $(X_i, Y_i) \in \mathcal{F}$, define

$$\Delta_i := \{0^{192}, 0^{64} \| \lceil X_i \rceil_{128}, 0^{191} \| 1, \}.$$

`Fcon` : There exists $(X_i, Y_i) \prec (X_j, Y_j) \in \mathcal{F}$, such that (X_j, Y_j) is a forward query and $\lceil Y_j \rceil_{192} \oplus \lceil X_i \rceil_{192} \in \Delta_j$.
`Bcon` : There exists $(X_i, Y_i) \prec (X_j, Y_j) \in \mathcal{F}$ such that (X_j, Y_j) is a backward query and $\lceil Y_i \rceil_{192} \oplus \lceil X_j \rceil_{192} \in \Delta_i$.
`Bend` : There exists $(X_i, Y_i) \prec (X_j, Y_j) \in \mathcal{F}$, both backward queries $\lfloor X_j \rfloor_{64} = IV$ and $\lceil X_j \rceil_{128} = \lceil Y_j \oplus X_i \rceil_{128}$.

Next consider the following events due to two distinct internal sequences.

`Scoll`: There exists two internal sequence $S_{in} \neq S'_{in}$ in $\mathcal{F}$ and integers $i \in [0, t_S - 2], j \in [0, T_{S'} - 2]$ such that $S_{in}[i] \neq S'_{in}[j]$ but $S_{in}[i+1] = S'_{in}[j+1]$.

Define `SBAD` := `Fconnect` $\cup$ `Bconnect` $\cup$ `Bend` $\cup$ `Scoll`

Lemma 1.

$$\Pr[\texttt{SBAD}] \leq \frac{7q^2}{2^{193}}.$$

Proof. Proof of the lemma follows from the observation that $\mathcal{F}$ is generated using a random permutation Π.

Proposition 3. *If* SBAD *doesn't occur, then the following holds in* $\mathcal{F}$*.*

– *for all internal sequences* S_{in} *in* $\mathcal{F}$

$$S_{in}[0] \prec \cdots \prec S_{in}[t_S - 1]$$

– *For any two internal sequences* $S_{in} \neq S'_{in}$ *in* $\mathcal{F}$

$$S_{in}[t_S - 1] \neq S'_{in}[t_{S'} - 1]$$

Corollary 3. *If* SBAD *doesn't occur in* ($\mathcal{F}$)*, then there exists at most* q *distinct internal sequences in* $\mathcal{F}$*.*

Lemma 2. *If* SBAD *doesn't in* $\mathcal{F}$*. Then,*

$$\mathbf{Adv}^{\text{cmt-4}}_{\textit{Ascon-256.v3}} \leq \frac{q^2}{2^{129}} + \frac{3q^2}{2^{193}} + \frac{q^4}{2^{514}} + \frac{q^3}{2^{322}}.$$

Proof. Let S, S' denote the two full sequences in $\mathcal{F}$ generated by the CMT-4 adversary. Let S_{in}, S'_{in} denote the internal sequences of S, S'. First suppose $S_{in} = S'_{in}$ and C, T denote the corresponding ciphertext. Then if $|C| = d \mod 128$, from the construction of Ascon-256.v3 we must have,

$$\lfloor X_{t_S} \oplus X_{t_{S'}} \rfloor_{128} := \lfloor K'_S \oplus K'_{S'} \rfloor_d \| 0^{128-d}$$

where $S_{in}[0] = IV \| K'_S \| K_S$ and $S'_{in}[0] = IV \| K'_{S'} \| K_{S'}$ are fixed given any two S_{in}, S'_{in}.

With these observations we consider the following cases.

CASE1: $S_{in} = S'_{in}$. But then we have

$$X_{t_S} = X_{t_{S'}}$$

and hence $S = S'$.

CASE2: $S_{in} \neq S'_{in}, (X_{t_S}, Y_{t_S}) \prec (X_{t_S-1}, Y_{t_S-1})$. In this case without loss of generality we consider the following sub-cases.

- $(X_{t_S}, Y_{t_S}) \prec (X_0, Y_0)$. Note that this implies the case that given any X, Y in $\mathcal{F}$ one constructs a sequence S_{in} such that $\lceil Y_{t_S-1} \rceil_{192} \oplus 0^{64} \| \lceil X_0 \rceil_{128} = \lceil X \rceil_q$. Further since X_{t_S-1}, Y_{t_S-1} must be a forward query, Hence probability this happens is bounded by $\frac{q}{2^{192}}$. varying over all possible S_{in} we have probability of this case is bounded by $\frac{q^2}{2^{193}}$.
- $(X_0, Y_0) \prec (X_{t_S}, Y_{t_S}) \prec (X_{t_S-1}, Y_{t_S-1})$. Note that in this case given (X_0, Y_0) there exists a unique forward query of the form X_{t_S-1}, Y_{t_S-1}. Further since $(X_{t_S}, Y_{t_S}) \prec (X_{t_S-1}, Y_{t_S-1})$ hence $\lceil X_{t_S} \rceil_{192}$ is fixed. Hence in this case probability that $\lceil Y_{t_S-1} \rceil_{192} \oplus 0^{64} \| \lceil X_0 \rceil_{128} = \lceil X_{t_S} \rceil_q$ is again bounded by $\frac{1}{2^{192}}$ varying over all $(X_0, Y_0) \prec (X_{t_S}, Y_{t_S})$ we have probability of this case is bounded by $\frac{q^2}{2^{193}}$.

CASE3: $S_{in} \neq S'_{in}$, $(X_{t_S-1}, Y_{t_S-1}) \prec (X_{t_S}, Y_{t_S})$ and $(X_{t_{S'}-1}, Y_{t_{S'}-1}) \prec (X_{t_{S'}}, Y_{t_{S'}})$. In this case without loss of generality suppose $(X_{t_S}, Y_{t_S}) \prec (X_{t_{S'}}, Y_{t_{S'}})$. Note that in this case $K_S, K'_S, K_{S'}, K'_{S'}$ are fixed. We divide the case into the following sub-cases.

- Both are backward queries. in this case the adversary has full control over T. Hence varying over all S, S',probability that this case occurs is bounded by

$$\frac{q^2}{2} \times \sum_{(X,Y)\neq(X',Y')} \Pr\left[\begin{array}{c} \lceil X\rceil_{192}\oplus 0^{64} \| K_S = \lceil Y_{T_S-1}\rceil_{192} \\ \bigwedge \\ \lceil X'\rceil_{192}\oplus 0^{64} \| K_{S'} = \lceil Y_{T_{S'}-1}\rceil_{192} \\ \bigwedge \\ \lfloor X\oplus X'\rfloor_{128} := \lfloor K'_S \oplus K'_{S'}\rfloor_d \| 0^{128-d} \end{array}\right] \leq \frac{q^4}{2^{514}}.$$

- (X_{t_S}, Y_{t_S}) forward query and $(X_{t_{S'}}, Y_{t_{S'}})$ backward query. In this case T is fixed. Hence the probability of this case is bounded by

$$\frac{q^2}{2} \times \sum_{(X,Y)\neq(X',Y')} \Pr\left[\begin{array}{c} \lceil X_{S'}\rceil_{192}\oplus 0^{64} \| K_{S'} = \lceil Y_{T_{S'}-1}\rceil_{192} \\ \bigwedge \\ \lfloor X_{t_S}\oplus X_{t_{S'}}\rfloor_{128} := \lfloor K'_S \oplus K'_{S'}\rfloor_d \| 0^{128-d} \end{array}\right] \leq \frac{q^3}{2^{322}}.$$

- (X_{t_S}, Y_{t_S}) backward query and $(X_{t_{S'}}, Y_{t_{S'}})$ forward query. Probability that this case occurs is bounded by

$$\frac{q^2}{2} \times \sum_{(X,Y)\neq(X',Y')} \Pr\left[\begin{array}{c} \lceil X_S\rceil_{192}\oplus 0^{64} \| K_S = \lceil Y_{T_S-1}\rceil_{192} \\ \bigwedge \\ \lceil Y_{t_S}\rceil_{128}\oplus\lceil Y_{t_{S'}}\rceil_{128} := K_S \oplus K_{S'} \end{array}\right] \leq \frac{q^3}{2^{322}}.$$

- Both are forward queries. This case is bounded by

$$\frac{q^2}{2} \times \Pr\left[\lceil Y_{t_S}\rceil_{128} \oplus \lceil Y_{t_{S'}}\rceil_{128} := K_S \oplus K_{S'}\right] \leq \frac{q^2}{2^{129}}.$$

7 AEAD Security of Ascon-256.v2 and Ascon-256.v3 in the Nonce-Respecting Multi-User Settings

In this section we analyse the AEAD security of our newly proposed schemes namely Ascon-256.v2 and Ascon-256.v3. In this regard we would like to remark that as constructions which use AsconAEAD128 as an internal API, the security of these constructions can be derived directly from the security proofs shown in [2][3]. Nonetheless we provide a modular proof sketch which mostly describes the extra bad events that appear due to the tweaks in our constructions.

7.1 AEAD Security proof for Ascon-256.v2

Theorem 6 (Ascon-256.v2). *Consider a nonce-respecting AEAD adversary $\mathscr{A}$ making q_p permutation queries, q_e encryption queries with a total number of σ_e data blocks, q_d decryption queries with a total number of σ_d data blocks. Define*

$\sigma := \sigma_e + \sigma_d$. *Then, the upper bound of the AEAD advantage of* $\mathscr{A}$ *against* Ascon-256.v2 *is the following:*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{mu-AEAD}}_{\textit{Ascon-256.v2}}(\mathscr{A}) \leq\ & \frac{2q_d}{2^\tau} + \frac{\sigma_e^2}{2^b} + \frac{\sigma_d(q_p+\sigma_d)}{2^b} + \frac{\mathsf{mcoll}(\sigma_e, 2^r)\times(\sigma_d+q_p)}{2^c} \\ & + \frac{\mathsf{mcoll}(q_e, 2^\tau)q_d}{2^c} + \frac{\mu^2}{2^\kappa} + \frac{\mu(q_p+\sigma)}{2^\kappa} \\ & + \frac{q_d^2+q_e^2+q_eq_d+(q_e+q_d)(\sigma+q_p)}{2^b} \\ & + \frac{\mathsf{mcoll}(q_e, 2^{b-\kappa_2})(\sigma+q_p)}{2^{\kappa_2}} + \frac{q_e(\sigma+q_p)}{2^b} \\ & + \frac{\mathsf{mcoll}(\sigma+q_p, 2^\tau)q_d}{2^{\kappa_2}}\end{aligned}$$

Proof. The proof of this theorem directly follows from the works of Chakraborty et al. [2], [3]. Thus we omit the proof for this construction.

7.2 AEAD Security Proof for Ascon-256.v3

Theorem 7 (Ascon-256.v3). *Consider a nonce-respecting AEAD adversary* $\mathscr{A}$ *making* q_p *permutation queries,* q_e *encryption queries with a total number of* σ_e *data blocks,* q_d *decryption queries with a total number of* σ_d *data blocks. Define* $\sigma := \sigma_e + \sigma_d$. *Then, the upper bound of the AEAD advantage of* $\mathscr{A}$ *against* Ascon-256.v3 *is the following:*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{mu-AEAD}}_{\textit{Ascon-256.v3}}(\mathscr{A}) \leq\ & \frac{2q_d}{2^\tau} + \frac{\sigma_e^2}{2^b} + \frac{\sigma_d(\sigma_d+q_p)}{2^b} + \frac{\mathsf{mcoll}(\sigma_e, 2^r)\times(\sigma_d+q_p)}{2^c} + \frac{\mu^2}{2^\kappa} \\ & + \frac{\mu(q_p+\sigma)}{2^\kappa} + \frac{3q_d^2+3q_e^2+3q_eq_d+(q_e+q_d)(\sigma+q_p)}{2^b} \\ & + \frac{\mathsf{mcoll}(\sigma+q_p, 2^r)\times(q_e+q_d)}{2^c} + \frac{\mathsf{mcoll}(\sigma+q_p, 2^\tau)q_d}{2^{\kappa_2}} \\ & + \frac{\mathsf{mcoll}(q_e, 2^{b-\kappa_2})(\sigma+q_p)}{2^{\kappa_2}} + \frac{q_e(\sigma+q_p)}{2^b}\end{aligned}$$

Description of the Real World for Ascon-256.v3

Sample Keys $\mathsf{K}_1, \ldots, \mathsf{K}_\mu \leftarrow_\$ \{0,1\}^\kappa$.
Sample Random Permutation $\Pi \leftarrow_\$ \mathsf{Perm}(b)$.
Online Transcript All the queries are responded honestly following the Ascon AEAD and Π. Set:

$$\Theta_{real,online} := \left((\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i, \mathsf{C}_i, \mathsf{T}_i)_{i\in[q_e]}, (\mathsf{u}'_i, \mathsf{N}'_i, \mathsf{A}'_i, \mathsf{C}'_i, \mathsf{T}'_i, \mathsf{M}'_i)_{i\in[q_d]}, \mathsf{P}\right)$$

Complete Transcript Set the extended partial function to be $\mathsf{P}_{\mathsf{fin}}$ and Set:

$$\Theta_{real} := \left((\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i, \mathsf{C}_i, \mathsf{T}_i)_{i \in [q_e]}, (\mathsf{u}'_i, \mathsf{N}'_i, \mathsf{A}'_i, \mathsf{C}'_i, \mathsf{T}'_i, \mathsf{M}'_i)_{i \in [q_d]}, \mathsf{P}_{\mathsf{fin}} \right)$$

Probability Computation For any real world realizable transcript

$$\theta := \left((u_i, N_i, A_i, M_i, C_i, T_i)_{i \in [q_e]}, (u'_i, N'_i, A'_i, C'_i, T'_i, M'_i)_{i \in [q_d]}, P_{\mathsf{fin}} \right)$$

we obtain that,

$$\Pr[\Theta = \theta] = \Pr[\mathsf{P}_{\mathsf{fin}} \subset \Pi] = \frac{1}{(2^b)_{|\mathsf{P}_{\mathsf{fin}}|}}$$

Description of the Ideal World for Ascon-256.v3

We present the Ideal World Sampling along with the bad events by the help of 6 algorithms. We list the algorithms here:

- Algorithm 1: It provides the Online Transcript obtained in the Ideal World.
- Algorithm 2: It sets the internal states of Encryption Queries.
- Algorithm 3: Sets the internal states of the Decryption Queries.
- Algorithm 4: Samples the key and sets the initial states.
- Algorithm 5: It sets the final two states of the construction.
- Algorithm 6: Finally, at this step the tag consistency is checked.

Algorithm 1: Ideal World: Online Transcript Ascon-256.v3

Online Phase
Input: q_e encryption, q_d decryption, q_p primitive queries
Output: Ideal world responses
for *encryption query* $i \in [q_e]$ *with* $(\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i)$ **do**
  $\mathsf{C}_i \xleftarrow{\$} \{0,1\}^{|\mathsf{M}_i|}$
  $\mathsf{T}_i \xleftarrow{\$} \{0,1\}^{\tau}$
  return $\boxed{(\mathsf{C}_i, \mathsf{T}_i)}$
end
for *decryption query* $i \in [q_d]$ *with* $(\mathsf{u}'_i, \mathsf{N}'_i, \mathsf{A}'_i, \mathsf{C}'_i, \mathsf{T}'_i)$ **do**
  return $\boxed{\mathsf{rej}}$
end
for *primitive query* $i \in [q_p]$ *with* $(\mathsf{Q}_i, \mathsf{dir}_i)$ **do**
  if $\mathsf{dir}_i = +1$ **then**
    $\mathsf{U}_i \leftarrow \mathsf{Q}_i$
    $\mathsf{V}_i \xleftarrow{\$} \{0,1\}^b \setminus \mathsf{range}(\mathsf{P})$
    $\mathsf{P} \leftarrow \mathsf{P} \cup \{(\mathsf{U}_i, \mathsf{V}_i)\}$
    return $\boxed{\mathsf{V}_i}$
  else
    $\mathsf{V}_i \leftarrow \mathsf{Q}_i$
    $\mathsf{U}_i \xleftarrow{\$} \{0,1\}^b \setminus \mathsf{domain}(\mathsf{P})$
    $\mathsf{P} \leftarrow \mathsf{P} \cup \{(\mathsf{U}_i, \mathsf{V}_i)\}$
    return $\boxed{\mathsf{U}_i}$
  end
end
return

$$\boxed{\Theta_{\text{ideal,online}} \leftarrow \big((\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i, \mathsf{C}_i, \mathsf{T}_i)_{i\in[q_e]}, (\mathsf{u}'_i, \mathsf{N}'_i, \mathsf{A}'_i, \mathsf{C}'_i, \mathsf{T}'_i, \mathsf{rej})_{i\in[q_d]}, \mathsf{P}\big)}$$

if $\exists (i,j) \in [q_e] \times [q_d]$ *with later* $i > j$ *and* $(\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{C}_i, \mathsf{T}_i) = (\mathsf{u}'_j, \mathsf{N}'_j, \mathsf{A}'_j, \mathsf{C}'_j, \mathsf{T}'_j)$ **then**
  $\boxed{\mathsf{bad}_1 \leftarrow 1}$
end

Algorithm 2: Ideal World (Offline Phase for Ascon-256.v3): Encryption Query Internal States

for $i \in [q_e]$ **do**
  $(\mathsf{A}_{i,1}, \ldots, \mathsf{A}_{i,a_i}) \xleftarrow{\text{r}} \mathsf{pad}_1(\mathsf{N}_i, \mathsf{A}_i)$; $(\mathsf{M}_{i,1}, \ldots, \mathsf{M}_{i,m_i}) \xleftarrow{\text{r}} \mathsf{M}_i$; $(\mathsf{C}_{i,1}, \ldots, \mathsf{C}_{i,m_i}) \xleftarrow{\text{r}} \mathsf{C}_i$
  $t_i = a_i + m_i$, $d_i = |\mathsf{M}_{i,m_i}|$
  $\mathsf{V}_{i,0}, \ldots, \mathsf{V}_{i,a_i-1} \xleftarrow{\$} \{0,1\}^b$, $\mathsf{Z}_{i,a_i+1}, \ldots, \mathsf{Z}_{i,t_i-1} \xleftarrow{\$} \{0,1\}^c$, $\delta_i^* \xleftarrow{\$} \{0,1\}^{r-d_i}$
  if $a_i > 0$ **then**
    for $j = 1$ **to** a_i **do**
      $\mathsf{U}_{i,j} = \mathsf{V}_{i,j-1} \oplus (\mathsf{A}_{i,j} \parallel 0^c)$
    end
    $\mathsf{V}_{i,a_i} = (\mathsf{C}_{i,1} \oplus \mathsf{M}_{i,1}) \parallel \mathsf{Z}_{i,a_i}$
  end
  if $m_i \geq 2$ **then**
    $\mathsf{U}_{i,a_i+1} = \mathsf{C}_{i,1} \parallel (\mathsf{Z}_{i,a_i} \oplus 0^{c-1}1)$
    for $j = 2$ **to** $m_i - 2$ **do**
      $\mathsf{U}_{i,a_i+j} = \mathsf{C}_{i,j} \parallel \mathsf{Z}_{i,a_i+j-1}$
    end
    for $j = 1$ **to** $m_i - 3$ **do**
      $\mathsf{V}_{i,a_i+j} = (\mathsf{C}_{i,j+1} \oplus \mathsf{M}_{i,j+1}) \parallel \mathsf{Z}_{i,a_i+j-1}$
    end
    $\mathsf{V}_{i,t_i-1} = (\mathsf{C}_{i,m_i} \oplus \mathsf{M}_{i,m_i}) \parallel \delta_i^* \parallel \mathsf{Z}_{i,t_i-1}$
  end
end
return $\boxed{\mathsf{P}_E = \{(\mathsf{U}_{i,j}, \mathsf{V}_{i,j}) : i \in [q_e], j \in [t_i - 1]\}}$
if $\exists (i,j) \neq (i',j')$ *with* $\mathsf{U}_{i,j} = \mathsf{U}_{i',j'}$ *or* $\mathsf{V}_{i,j} = \mathsf{V}_{i',j'}$ **then**
  $\boxed{\mathsf{bad}_2 \leftarrow 1}$
end

Algorithm 3: Ideal World (Offline Phase for Ascon-256.v3): Decryption Query Internal States (Extension of P)

for $i \in [q_d]$ **do**
  $\left(\mathsf{A}'_{i,1}, \ldots, \mathsf{A}'_{i,a_i}\right) \leftarrow \mathsf{pad}_1(\mathsf{N}'_i, \mathsf{A}'_i)$; $\left(\mathsf{C}'_{i,1}, \ldots, \mathsf{C}'_{i,c_i}\right) \leftarrow \mathsf{C}'_i$
  if $\nexists j \in [q_e]$ *with* $\mathsf{u}_j = \mathsf{u}'_i$ **then**
    $p_i \leftarrow -1$
  else if $\nexists j \in [q_e]$ *with* $(\mathsf{u}_j, \mathsf{N}_j) = (\mathsf{u}'_i, \mathsf{N}'_i)$ **then**
    $p_i \leftarrow 0$
  else
    $p_i \leftarrow$ LCP of $\left(\mathsf{A}'_{i,1}, \ldots, (\mathsf{A}'_{i,a'_i}, *), \mathsf{C}'_{i,1}, \ldots, \mathsf{C}'_{i,c_i-2}\right)$ and $\left(\mathsf{A}_{j,1}, \ldots, (\mathsf{A}_{j,a_j}, *), \mathsf{C}_{j,1}, \ldots, \mathsf{C}_{j,m_j-2}\right)$
  end
end
for $i \in [q_d]$ *with* $p_i = -1$ **do**
  if $(\mathsf{u}'_i, \mathsf{N}'_i) = (\mathsf{u}_j, \mathsf{N}_j)$ *for some* $j \in [i-1]$ **then**
    $\mathsf{V}'_{i,0} \leftarrow \mathsf{V}_{j,0}$
  else
    $\mathsf{V}'_{i,0} \xleftarrow{\$} \{0,1\}^b$
  end
  if $a'_i > 0$ **then**
    Run $\mathsf{xorRand\_Extn}^P(\mathsf{V}'_{i,0}, (\mathsf{A}'_{i,1}, \ldots, \mathsf{A}'_{i,a'_i}))$
  end
  if $c_i > 1$ **then**
    Run $\mathsf{Rand\_Extn}^P(\mathsf{V}'_{i,a'_i} \oplus 0^*1, \mathsf{C}'_{i,1} \| \ldots \| \mathsf{C}'_{i,c_i-2})$
  end
end
for $i \in [q_d]$ *with* $0 \leq p_i \leq a'_i$ **do**
  $\mathsf{V}'_{i,p_i} \leftarrow \mathsf{V}_{j,p_i}$ where $(\mathsf{u}'_i, \mathsf{N}'_i) = (\mathsf{u}_j, \mathsf{N}_j)$
  if $a'_i > p_i$ **then**
    Run $\mathsf{xorRand\_Extn}^P(\mathsf{V}'_{i,p_i}, (\mathsf{A}'_{i,p_i+1}, \ldots, \mathsf{A}'_{i,a'_i}))$
  end
  if $c_i > 1$ **then**
    Run $\mathsf{Rand\_Extn}^P(\mathsf{V}'_{i,a'_i} \oplus 0^*1, \mathsf{C}'_{i,1} \| \ldots \| \mathsf{C}'_{i,c_i-2})$
  end
end
for $i \in [q_d]$ *with* $a'_i < p_i < t_i - 1$ **do**
  $\mathsf{V}'_{i,p_i} \leftarrow \mathsf{V}_{j,p_i}$ where $(\mathsf{u}'_i, \mathsf{N}'_i) = (\mathsf{u}_j, \mathsf{N}_j)$
  if $p_i < t_i - 2$ **then**
    Run $\mathsf{Rand\_Extn}^P(\mathsf{V}'_{i,p_i} \oplus 0^*1, \mathsf{C}'_{i,p_i-a'_i+1} \| \ldots \| \mathsf{C}'_{i,c_i-2})$
  end
end
return $\boxed{\mathsf{P}_1 = \{(\mathsf{U}'_{i,j}, \mathsf{V}'_{i,j})\}_{i \in [q_d], j \in [t_i-1]}}$
if P_1 *not injective* **then**
  $\boxed{\mathsf{bad}_3 \leftarrow 1}$
end
if $\mathsf{domain}(\mathsf{P}_1) \cap \mathsf{domain}(\mathsf{P}_\mathsf{E}) \neq \emptyset \ \vee \ \mathsf{range}(\mathsf{P}_1) \cap \mathsf{range}(\mathsf{P}_\mathsf{E}) \neq \emptyset$ **then**
  $\boxed{\mathsf{bad}_4 \leftarrow 1}$
end
Extend: $\boxed{\mathsf{P}_2 \leftarrow \mathsf{P}_E \sqcup \mathsf{P}_1}$

Algorithm 4: Ideal World (Offline Phase for Ascon-256.v3): Key Sampling and Initialization

$\mathsf{K}_1, \ldots, \mathsf{K}_\mu \xleftarrow{\$} \{0,1\}^\kappa$ where $\mathsf{K}_i = \mathsf{K}_{i,1} \parallel \mathsf{K}_{i,2}$
$\mathcal{J} \leftarrow \{j \in [q_d] : \mathsf{u}'_j \neq \mathsf{u}_i \ \forall i\}$
for $i \in [q_e]$ **do**
  $\mathsf{I}_i \leftarrow IV \parallel K_{i,2} \parallel K_{i,1}$
  $\mathsf{O}_i \leftarrow \mathsf{V}_{i,0} \oplus (\mathsf{K}_{i,1} \parallel 0^{b-\kappa} \parallel K_{i,2})$
end
for $j \in \mathcal{J}$ **do**
  $\mathsf{I}'_j \leftarrow IV \parallel K_{j,2} \parallel K_{j,1}$
  $\mathsf{O}'_j \leftarrow \mathsf{V}'_{j,0} \oplus (\mathsf{K}_{i,1} \parallel 0^{b-\kappa} \parallel K_{i,2})$
end
for $j \in [q_d] \setminus \mathcal{J}$ **do**
  $\mathsf{I}'_j \leftarrow \mathsf{I}_i$
  $\mathsf{O}'_j \leftarrow \mathsf{O}_i$
end
return $\boxed{\mathsf{P}_{\mathsf{init}} \leftarrow \big((\mathsf{I}_i, \mathsf{O}_i)_{i \in [q_e]}, (\mathsf{I}'_j, \mathsf{O}'_j)_{j \in \mathcal{J}}\big)}$
if $\exists i \neq j \in [\mu]$ *with* $\mathsf{K}_i = \mathsf{K}_j$ **then**
  $\boxed{\mathsf{bad}_5 \leftarrow 1}$
end
if $\mathsf{domain}(\mathsf{P}_2) \cap \mathsf{domain}(\mathsf{P}_{\mathsf{init}}) \neq \emptyset \ \vee \ \mathsf{range}(\mathsf{P}_2) \cap \mathsf{range}(\mathsf{P}_{\mathsf{init}}) \neq \emptyset$ **then**
  $\boxed{\mathsf{bad}_6 \leftarrow 1}$
end
Extend : $\boxed{\mathsf{P}_3 \leftarrow \mathsf{P}_2 \sqcup \mathsf{P}_{\mathsf{init}}}$

Algorithm 5: Ideal World (Offline Phase for Ascon-256.v3): Finalization- Last Two Blocks

for *encryption query* $i \in [q_e]$ **do**
  $m_i \leftarrow \lceil \frac{|\mathsf{M}_i|+1}{r} \rceil$
  $d_i \leftarrow |\mathsf{M}_i| \bmod r$
  if $m_i \geq 2$ **then**
    $\mathsf{V}_{i,a_i+m_i-2} \leftarrow \left(\mathsf{C}_{i,m_i-1} \oplus \mathsf{M}_{i,m_i-1} \oplus (0^{d_i} \,\|\, \lceil \mathsf{K}_{i,1} \rceil_{r-d_i})\right) \|\, \mathsf{Z}_{a_i+m_i-2}$
    $\mathsf{F}_{i,0} \leftarrow \left(\mathsf{C}_{i,m_i-1} \oplus (0^{d_i} \,\|\, \lceil \mathsf{K}_{i,1} \rceil_{r-d_i})\right) \|\, \lfloor \mathsf{V}_{a_i+m_i-2} \rfloor_c$
    $\mathsf{V}_{i,a_i+m_i-1} \leftarrow (\mathsf{C}_{i,m_i} \oplus \mathsf{M}_{i,m_i} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}) \,\|\, \delta_i^* \,\|\, \mathsf{Z}_{a_i+m_i-1}$
    $\mathsf{F}_{i,1} \leftarrow (\mathsf{C}_{i,m_i} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}) \,\|\, \delta_i^* \,\|\, \lfloor \mathsf{V}_{a_i+m_i-1} \rfloor_c$
  end
  if $m_i = 1$ **then**
    $\mathsf{F}_{i,1} \leftarrow (\mathsf{C}_{i,1} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}) \,\|\, \delta_i^* \,\|\, (\mathsf{Z}_{i,a_i} \oplus 0^{c-1}1)$
    $\mathsf{F}_{i,0} \leftarrow \mathsf{U}_{i,a_i}$
  end
end
for *decryption query* $i \in [q_d]$ **do**
  if $c_i \geq 2$ **then**
    $\mathsf{F}'_{i,0} \leftarrow \left(\mathsf{C}'_{i,c_i-1} \oplus (0^{d_i} \,\|\, \lceil \mathsf{K}_{i,1} \rceil_{r-d_i})\right) \|\, \lfloor \mathsf{V}'_{a'_i+c_i-2} \rfloor_c$
    $\mathsf{V}'_{i,a'_i+c_i-1} \leftarrow \left(\mathsf{C}'_{i,c_i} \oplus \mathsf{M}'_{i,m'_i} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}\right) \|\, \delta_i'^* \,\|\, \mathsf{Z}'_{a'_i+c_i-1}$
    $\mathsf{F}'_{i,1} \leftarrow \left(\mathsf{C}'_{i,c_i} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}\right) \|\, 10^* \,\|\, \lfloor \mathsf{V}'_{a'_i+c_i-1} \rfloor_c$
  end
  if $c_i = 1$ **then**
    $\mathsf{F}'_{i,1} \leftarrow \left(\mathsf{C}'_{i,c_i} \oplus \lfloor \mathsf{K}_{i,1} \rfloor_{d_i}\right) \|\, 10^* \,\|\, \left(\lfloor \mathsf{V}'_{a'_i+c_i-1} \rfloor_c \oplus 0^{c-1}1\right)$
    $\mathsf{F}'_{i,0} \leftarrow \mathsf{U}'_{i,a_i}$
  end
end
return $\boxed{\mathsf{P}_F \leftarrow \{(\mathsf{F}_{i,0}, \mathsf{V}_{i,t_i-1})\}_{i\in[q_e]} \cup \{(\mathsf{F}'_{i,0}, \mathsf{V}'_{a'_i+c_i-1})\}_{i\in[q_d]}}$
if P_F *not injective* **then**
  $\boxed{\mathsf{bad}_7 \leftarrow 1}$
end
if $\mathsf{domain}(\mathsf{P}_F) \cap \mathsf{domain}(\mathsf{P}_3) \neq \emptyset \ \vee\ \mathsf{range}(\mathsf{P}_F) \cap \mathsf{range}(\mathsf{P}_3) \neq \emptyset$ **then**
  $\boxed{\mathsf{bad}_8 \leftarrow 1}$
end
if $\exists i \neq j \in [q_e]$ *with* $\mathsf{F}_{i,1} = \mathsf{F}_{j,1}$ ***or*** $\exists i \neq j \in [q_d]$ *with* $\mathsf{F}'_{i,1} = \mathsf{F}'_{j,1}$ **then**
  $\boxed{\mathsf{bad}_9 \leftarrow 1}$
end
if $\exists (i,j) \in [q_e] \times [q_d]$ *with* $(\mathsf{F}'_{i,1}, \mathsf{T}'_i) = (\mathsf{F}_{j,1}, \mathsf{T}_j)$ **then**
  $\boxed{\mathsf{bad}_{10} \leftarrow 1}$
end
Extend : $\boxed{\mathsf{P}_4 \leftarrow \mathsf{P}_3 \sqcup \mathsf{P}_F}$

Algorithm 6: Ideal World (Offline Phase for Ascon-256.v3): Finalization- Tag Consistency

for *encryption query* $i \in [q_e]$ **do**
  $\mathsf{X}_i \leftarrow \mathsf{F}_i \oplus 0^{b-\kappa_2} \parallel \mathsf{K}_{u_i,2}$
  $\mathsf{Y}_i \leftarrow \alpha_i \parallel (\mathsf{T}_i \oplus \mathsf{K}_{u_i,2})$ where $\alpha_i \xleftarrow{\$} \{0,1\}^{b-\tau}$
end
return $\boxed{\mathsf{P}_{\mathsf{tag}} \leftarrow (\mathsf{X}_i, \mathsf{Y}_i)_{i\in[q_e]}}$
if $\mathsf{domain}(\mathsf{P}_{\mathsf{tag}}) \cap \mathsf{domain}(\mathsf{P}_4) \neq \emptyset \ \vee \ \mathsf{range}(\mathsf{P}_{\mathsf{tag}}) \cap \mathsf{range}(\mathsf{P}_4) \neq \emptyset$ **then**
  $\boxed{\mathsf{bad}_{11} \leftarrow 1}$
end
Extend : $\boxed{\mathsf{P}_5 \leftarrow \mathsf{P}_4 \sqcup \mathsf{P}_{\mathsf{tag}}}$
for *decryption query* $i \in [q_d]$ **do**
  $\mathsf{X}'_i \leftarrow \mathsf{F}'_i \oplus 0^{b-\kappa_2} \parallel \mathsf{K}_{u'_i,2}$
  if $\mathsf{X}'_i \in \mathsf{domain}(\mathsf{P}_5)$ **then**
    $\mathsf{Y}'_i \leftarrow \mathsf{P}_4(\mathsf{X}'_i)$
    if $\lfloor \mathsf{P}_4(\mathsf{X}'_i) \rfloor_\tau \oplus \mathsf{K}_{u'_i,2} = \mathsf{T}'_i$ **then**
      $\boxed{\mathsf{bad}_{12} \leftarrow 1}$
    end
  else
    $\mathsf{Y}'_i \xleftarrow{\$} \{0,1\}^b$
    if $\lfloor \mathsf{Y}'_i \rfloor_\tau \oplus \mathsf{K}_{u'_i,2} = \mathsf{T}'_i$ **then**
      $\boxed{\mathsf{bad}_{13} \leftarrow 1}$
    end
  end
end
Extend : $\boxed{\mathsf{P}_{\mathsf{fin}} \leftarrow \mathsf{P}_5 \sqcup \{(\mathsf{X}'_i, \mathsf{Y}'_i)_{i\in[q_d]}\}}$
Ideal World Transcript :
$\boxed{\Theta_{\text{ideal}} \leftarrow \big((\mathsf{u}_i, \mathsf{N}_i, \mathsf{A}_i, \mathsf{M}_i, \mathsf{C}_i, \mathsf{T}_i)_{i\in[q_e]}, (\mathsf{u}'_i, \mathsf{N}'_i, \mathsf{A}'_i, \mathsf{C}'_i, \mathsf{T}'_i, \mathsf{M}'_i)_{i\in[q_d]}, \mathsf{P}_{\mathsf{fin}}\big)}$

Bad Analysis

Let us define,

$$\mathsf{bad}_* = \bigcup_{i=1}^{6} \mathsf{bad}_i \vee \bigcup_{j=10}^{13} \mathsf{bad}_j$$

Note that, bad_* is essentially same with the bad events defined in the earlier works of [2,3]. Thus, the similar bounds should hold true. We summarize the result in the following Lemma.

Lemma 3.

$$\begin{aligned}\Pr\left[\mathsf{bad}_* = 1\right] \leq \frac{q_d}{2^\tau} + \frac{\sigma_e^2}{2^b} + \frac{\sigma_d(\sigma_d + q_p)}{2^b} + \frac{\mathsf{mcoll}(\sigma_e, 2^r) \times (\sigma_d + q_p)}{2^c} + \frac{\mu^2}{2^\kappa} + \frac{\mu(q_p + \sigma)}{2^\kappa} \\ + \frac{q_d^2 + q_e^2 + q_e q_d + (q_e + q_d)(\sigma + q_p)}{2^b} + \frac{\mathsf{mcoll}(\sigma + q_p, 2^\tau) q_d}{2^{\kappa_2}} \\ + \frac{\mathsf{mcoll}(q_e, 2^{b-\kappa_2})(\sigma + q_p)}{2^{\kappa_2}} + \frac{q_e(\sigma + q_p)}{2^b} + \frac{\mathsf{mcoll}(\sigma + q_p, 2^\tau) q_d}{2^{\kappa_2}} + \frac{q_d}{2^\tau}\end{aligned}$$

Proof. Proof of this lemma directly follows from the works of Chakraborty et al. [2,3].

Hence we establish upper bounds on the probability of newly encountered bad events, i.e., $\mathsf{bad}_7, \mathsf{bad}_8$ and bad_9 (See Algorithm 5) dedicated to Ascon-256.v3.

Lemma 4. $\Pr\left[\mathsf{bad}_7 = 1\right] \leq \dfrac{(q_e + q_d)^2}{2^b}$

Proof. From the construction of P_F we have that $\mathsf{domain}(\mathsf{P}_F) \leq (q_e + q_d)$ and $\mathsf{range}(\mathsf{P}_F) \leq (q_e + q_d)$. For any two values $x, y \in \mathsf{domain}(\mathsf{P}_F)$ (or $x, y \in \mathsf{range}(\mathsf{P}_F)$) we have that $x = y$ with probability $\dfrac{1}{2^b}$. Hence, by union bound we derive the lemma.

Lemma 5. $\Pr\left[\mathsf{bad}_8 = 1\right] \leq \dfrac{\mathsf{mcoll}(\sigma + q_p, 2^r) \times (q_e + q_d)}{2^c}$

Proof. Let ρ_1 (and ρ_2) denote the multicollision on the values of $\lceil x \rceil_r$ for all $x \in \mathsf{domain}(\mathsf{P}_3)$ (and for all $x \in \mathsf{range}(\mathsf{P}_3)$, respectively). Then, by randomness of randomised extension process and xor randomised extension process we have that

$$\Pr(\mathsf{bad}_8 = 1 \mid \max\{\rho_1, \rho_2\} = \rho) \leq \frac{\rho \times (q_e + q_d)}{2^c}$$

Hence by using the expectation on ρ_1 we obtain,

$$\Pr(\mathsf{bad}_8 = 1) \leq \frac{\mathsf{mcoll}(\sigma + q_p, 2^r) \times (q_e + q_d)}{2^c}$$

Lemma 6. $\Pr\left[\mathsf{bad}_9 = 1\right] \leq \dfrac{q_e^2 + q_d^2}{2^b}$

Proof. The proof is straightforward.

Good Analysis

Let θ be a good transcript (no bad events occur). Note that we sample either inputs or outputs of $\mathsf{P}_{\mathsf{fin}} \setminus \mathsf{P}$ uniformly. Thus,

$$\Pr\left[\Theta_{ideal} = \theta\right] = \Pr\left[P \subseteq \Pi\right] \times 2^{-b(|\mathsf{P}_{\mathsf{fin}}|-|\mathsf{P}|)} \leq \frac{1}{(2^b)_{|\mathsf{P}_{\mathsf{fin}}|}} = \Pr\left[\Theta_{real} = \theta\right]$$

Thus by H-Coefficient Technique we conclude the proof of this theorem.

8 Conclusion

In this draft, we revisit the mu-Ascon instantiation proposed by NIST for multi-user applications, which only provides partial security against key-recovery under state recovery attacks and doesn't provide any commitment security. Dealing with these weaknesses, we generate two schemes which we call Ascon-256.v2 and Ascon-256.v3 for the multi-user settings, which are compatible with the AsconAEAD128 API. Due to the usage of a single extra Ascon permutation, Ascon-256.v2 and Ascon-256.v3 have a very negligible performance difference in comparison to that of mu-Ascon. On the other hand we show that in compensation to these negligible expense in performance, Ascon-256.v2 resists CMT-4 attacks and Ascon-256.v3 resist CMT-4 attacks and at the same time also enjoys the full key-binding property.

References

1. Mihir Bellare and Viet Tung Hoang. Efficient schemes for committing authenticated encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 845–875, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07085-3_29.
2. Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Exact security analysis of ASCON. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part III*, volume 14440 of *Lecture Notes in Computer Science*, pages 346–369, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore. doi:10.1007/978-981-99-8727-6_12.
3. Bishwajit Chakraborty, Chandranan Dhar, and Mridul Nandi. Tight multi-user security of ascon and its large key extension. Lecture Notes in Computer Science, pages 57–76. Springer, Cham, Switzerland, December 1–3, 2024. doi:10.1007/978-981-97-5025-2_4.
4. Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014. Proceedings*, pages 327–350, 2014.
5. Christoph Dobraunig and Bart Mennink. Generalized initialization of the duplex construction. In Christina Pöpper and Lejla Batina, editors, *ACNS 24: 22nd International Conference on Applied Cryptography and Network Security,*

Part II, volume 14584 of *Lecture Notes in Computer Science*, pages 460–484, Abu Dhabi, UAE, March 5–8, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-54773-7_18.

6. Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 155–186, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-96884-1_6.
7. Morris Dworkin. Nist special publication 800-38b recommendation for block.
8. Pooya Farshim, Claudio Orlandi, and Răzvan Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Transactions on Symmetric Cryptology*, 2017(1):449–473, 2017. doi:10.13154/tosc.v2017.i1.449-473.
9. Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 66–97, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-63697-9_3.
10. Charlotte Lefevre and Bart Mennink. Generic security of the ascon mode: On the power of key blinding. Cryptology ePrint Archive, Paper 2023/796, 2023. URL: https://eprint.iacr.org/2023/796.
11. Charlotte Lefevre and Bart Mennink. Generic security of the ascon mode: On the power of key blinding. Cryptology ePrint Archive, Report 2023/796, 2023. URL: https://eprint.iacr.org/2023/796.
12. Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In *30th USENIX security symposium (USENIX Security 21)*, pages 195–212, 2021.
13. Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart. Context discovery and commitment attacks - how to break CCM, EAX, SIV, and more. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 379–407, Lyon, France, April 23–27, 2023. Springer, Cham, Switzerland. doi:10.1007/978-3-031-30634-1_13.
14. Bart Mennink and Samuel Neves. Encrypted davies-meyer and its dual: Towards optimal security using mirror theory. In *Advances in Cryptology - CRYPTO 2017. Proceedings, Part III*, pages 556–583, 2017.
15. Yusuke Naito, Yu Sasaki, and Takeshi Sugawara. Committing security of ascon: Cryptanalysis on primitive and proof on mode. *IACR Transactions on Symmetric Cryptology*, 2023(4):420–451, 2023.
16. Yoav Nir and Adam Langley. Chacha20 and poly1305 for ietf protocols. Technical report, 2015.
17. NIST. Submission requirements and evaluation criteria for the Lightweight Cryptography Standardization Process, 2018. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf.
18. 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.
19. Jacques Patarin. *Etude des Générateurs de Permutations Pseudo-aléatoires Basés sur le Schéma du DES*. PhD thesis, Université de Paris, 1991.
20. Jacques Patarin. The "coefficients H" technique. In *Selected Areas in Cryptography - SAC 2008. Revised Selected Papers*, pages 328–345, 2008.

21. Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. Technical report, National Institute of Standards and Technology, 2024.

16. Atsec Information CST lab, February 11, 2025

#	Line #	Comment (Include rationale for comment)	Suggested change. (if applicable)
1	Algorithm 4	In Algorithm 4, the absorption of the associated data blocks is described slightly different from Algorithm 3. Considering that equation 18 and 36 are identical (except the parentheses), it would be clearer if the steps of the algorithms are identical too.	Make Algorithm 3 and Algorithm 4 consistent with relation to the absorption of A.
2	483	"ciphertext" should be "ciphertexts"	Change to "ciphertexts"
3	Figure 10	In the "Word value (bitstring)" table, fourth row, third column, a space is missing	Change "10001011" to "1000 1011"
4	Figure 10/Table 11	The placement of Figure 10 and Table 11 causes the "Domain Separation Bit" paragraph (and the word "implemented") to break across page 32 and 34. This is very confusing.	Start the "Domain Separation Bit" paragraph on page 34.
5	Appendix A	Algorithm 5 Initialization step: This step describes an operation whose outcome is deterministic. Similar to the version 1.2 of the original Ascon paper, Appendix A should provide the pre-computed values of this step to allow implementors to skip this step and initialize the Ascon state with the pre-computed values.	Add the 5 constants in proper little-endian notation. see ASCON_HASH_IV* in the Ascon reference implementation https://github.com/ascon/asconc/blob/main/src/constants.h#L50-L54
6	Appendix A	Algorithm 6 Initialization step: See comment to Algorithm 5 Initialization step.	Add the 5 constants in proper little-endian notation. see ASCON_XOF_IV* in the Ascon reference implementation https://github.com/ascon/asconc/blob/main/src/constants.h#L62-L66
7	Appendix A	Algorithm 7 Initialization step: See comment to Algorithm 5 Initialization step.	Add the 5 constants in proper little-endian notation.
8	Section 2.1 Padding rule	The description of the padding rule is not clear that the bit-notation also changed such that the bit order is reversed (which is a fact that is not implied with little endian). See https://github.com/ascon/ascon-c/issues/19 for the discussion.	Please add a reference from the padding rule description to Table 11 explaining the bit-reversal.