Differential Characteristics of the NIST Lightweight Cryptography Standard Ascon: 42 Active S-boxes over 4 Rounds and 68 over 5 Rounds

Nobuo Otoi
BA Cybersecurity
otoi@ba-cybersecurity.com

November 2026

Abstract

This note analyses the permutation of Ascon, the NIST standard for lightweight authenticated encryption and hashing (SP 800-232); it proposes no new construction. We report a 4-round differential characteristic of the Ascon permutation with 42 differentially active S-boxes and a 5-round characteristic with 68 active S-boxes. To the best of our knowledge, the best previously known characteristics had 43 and 72 active S-boxes, respectively, so the minimum number of active S-boxes over 4 rounds now lies between the proven lower bound of 36 and 42. Both characteristics were found with a bit-level SAT model in which the cardinality bound is tightened incrementally under assumptions. We verified every round of both characteristics against the designers' reference implementation and, in addition, with a machine-checked proof in Lean 4. For the 4-round characteristic we also give one concrete pair of states that follows all four rounds, which anyone can check in a minute; for the 5-round characteristic we have no such pair. The characteristics have weight 138 and 226 (probability 2^{-138} and 2^{-226} under the usual independence assumption) and do not threaten the security of Ascon.

Keywords: symmetric-key cryptography, cryptanalysis, differential attack, Ascon, NIST SP 800-232, SAT solver, Lean 4

1 Introduction

Ascon [1] is the family of authenticated encryption and hashing algorithms that NIST selected in 2023 for constrained devices and standardised in SP 800-232 [2]. Its 320-bit permutation acts on five 64-bit words $(x_0, \ldots, x_4)$. Each round adds a constant to x_2, applies a 5-bit S-box S to each of the 64 columns (with x_0 as the most significant bit), where S maps $0, \ldots, 31$ to (in hexadecimal)

```
04 0b 1f 14 1a 15 09 02 1b 05 08 12 1d 03 06 1c
1e 13 07 0e 00 0d 11 18 10 0c 01 19 16 0a 0f 17
```

and then applies the linear layer

$$\Sigma_i(x_i) = x_i \oplus (x_i \ggg r_{i,1}) \oplus (x_i \ggg r_{i,2}), \tag{1}$$

with $(r_{i,1}, r_{i,2}) = (19, 28), (61, 39), (1, 6), (10, 17), (7, 41)$ for $i = 0, \ldots, 4$. An R-round differential characteristic is a sequence $(\Delta X_r, \Delta Y_r)_{r<R}$ of input and output differences of the S-box layer with $\Delta X_{r+1} = \Sigma(\Delta Y_r)$. Writing $a_{r,j}$ and $b_{r,j}$ for column j of ΔX_r and ΔY_r, and $\mathrm{DDT}[a][b] =$

$\big|\{v \in \mathbb{F}_2^5 : S(v) \oplus S(v \oplus a) = b\}\big|$, the characteristic is valid if every $\mathrm{DDT}[a_{r,j}][b_{r,j}] > 0$, and its probability (under the usual independence assumption) and number of active S-boxes are

$$p = \prod_{r=0}^{R-1} \prod_{j=0}^{63} \frac{\mathrm{DDT}[a_{r,j}][b_{r,j}]}{32}, \qquad \# = \sum_{r=0}^{R-1} \big|\{\, j : a_{r,j} \neq 0 \,\}\big|. \tag{2}$$

We call $w = -\log_2 p$ the weight. A differential attack exploits an input difference that leads to a predictable output difference with high probability. Since every active S-box multiplies p by at most $8/32 = 2^{-2}$, lower bounds on # over all non-trivial R-round characteristics are a common way to argue how many rounds resist such attacks, and a concrete characteristic shows how far such a bound can go at most.

For $R \leq 3$ the minimum is known: 1, 4 and 15 [1]. For $R = 4$ the designers reported a characteristic with 44 active S-boxes [1], Makarim and Rohit improved it to 43 [4], and Erlacher, Mendel and Eichlseder proved with SAT solvers that every 4-round characteristic has at least 36 [3]. For $R = 5$ the best known characteristic had 78 active S-boxes [1], improved to 72 [4]. Makarim and Rohit note that a tight bound for 4 to 6 rounds is still not known [4]. El Hirch et al. [5] improved the related bounds on trail weight.

Contribution. We give a 4-round characteristic with 42 active S-boxes (pattern 15–11–9–7) and a 5-round characteristic with 68 active S-boxes (15–11–9–7–26). Table 1 summarises the state of knowledge.

Check it yourself. The 4-round claim does not depend on our search or on trust in us. Take the two states below, P and $P \oplus \Delta$, and run both through the last four rounds of the permutation with the designers' `pyascon`, `ascon_permutation(S, rounds=4)`. The output difference is the one shown, as the characteristic predicts.

```
P   = a69ddcdf1017bede 84a033cdd3354b73 6bcd5a36901c4381 2638430fac79e91d a480735de75e0dc8
D   = b9460e2000000000 0000004080000001 39460c6080000000 0000000000000000 0000000000000001
out= 0000004000080400 8088b24ae4902660 0000000000000000 0000000000000000 8022182264982440
```

The script `verify/check.py` in the repository, which has no dependencies, repeats this round by round and counts the columns in which the two states differ before each S-box layer: 15, 11, 9 and 7, in total 42.

Table 1: Minimum number of differentially active S-boxes of the Ascon permutation.

Rounds	proven lower bound	best known characteristic
3	15 [1]	15
4	36 [3]	44 [1] → 43 [4] → **42** (this note)
5	37 (trivially 36 + 1)	78 [1] → 72 [4] → **68** (this note)

2 Search method

Model. We use a bit-level SAT model of R rounds. Variables X_r and Y_r hold the input and output differences of the S-box layer in round r (5×64 bits each). For each column, the 10-bit relation "$\mathrm{DDT}[a][b] > 0$" between input difference a and output difference b is encoded by blocking all impossible pairs; each blocking clause is shortened greedily while it still excludes no possible pair, which leaves 169 clauses per S-box (the DDT has 317 non-zero entries including $0 \to 0$). The linear layer (1) gives $X_{r+1} = \Sigma(Y_r)$ as 4-variable XOR constraints. An activity variable $A_{r,j}$ is forced to 1 whenever column j of X_r is non-zero. The round constants do not affect differences and are omitted. To remove the rotational symmetry of the state we require column 0 of X_0 to be non-zero.

Cardinality and Matsui-type constraints. The activity variables, in round order, feed one sequential counter [10] encoded in both directions, so that its register $c_{i,j}$ is true exactly when at least j of the first i variables are true. Following Sun, Wang and Wang [8], we add Matsui-type conditions [9]. With B_t a known lower bound for t rounds ($B_0 = 0$), the constraints are

$$\sum_{r=0}^{R-1}\sum_{j=0}^{63} A_{r,j} \le k, \qquad \sum_{r=s}^{e}\sum_{j=0}^{63} A_{r,j} \le k - B_s - B_{R-1-e} \quad (0 \le s \le e \le R-1,\ (s,e) \ne (0,R-1)). \tag{3}$$

We used $B = (0, 1, 4, 15)$, and $B_4 = 36$ for five rounds. Each window constraint is expressed as implications $c_{\mathrm{end}(e),\,t+m+1} \Rightarrow c_{\mathrm{end}(s-1),\,t+1}$ between counter registers at round boundaries, where m is the right-hand side, $\mathrm{end}(r) = 64(r+1)$, $c_{0,t}$ is false for $t \ge 1$, and t ranges over $0 \le t \le k-m$.

Descending search. Asking directly for a 4-round characteristic with at most 44 active S-boxes did not return within one hour. Instead we build the model once for a loose bound k_0 (60 for four rounds) and, each time the solver returns a characteristic with a active S-boxes, solve again under the assumption $\neg c_{N,a}$, i.e. at most $a-1$. The solver keeps its learned clauses between calls. With CaDiCaL 1.9.5 [11] through PySAT [12], on one core of an Apple M1 Max shared with other jobs, the sequence of characteristics found had 59, 58, 52, 49, 47, 46, 45 and 42 active S-boxes; the last one appeared after 2,821 s of wall-clock time. The solver jumped from 45 directly to 42. The model has 16,603 variables and 107,432 clauses. A second run with Glucose 4 reached 45 after 29 s and did not improve on it within the same time. The run that reached 42 was not typical: in three repeat runs that shuffle the clauses and move the symmetry-breaking column (option `--seed`), all reached 45 within three minutes; one reached 43 after 38,458 s, and two remained at 45 when stopped after about 11 h. The result therefore rests on the characteristics themselves, which anyone can check (Section 4), and not on the search being reliably reproducible.

Five rounds. The 5-round characteristic extends the 4-round one forward by one round. Its last S-box layer has 7 active columns, each with 8 possible output differences; enumerating all 8^7 choices and applying Σ gives at least 26 active S-boxes in the fifth round, for a total of 68. Extending backward instead costs 57 S-boxes, because Σ^{-1} is dense.

3 The characteristics

Table 2 lists the 5-round characteristic. Its first four rounds form a 4-round characteristic with 42 active S-boxes and weight 138; the 4-round result of the search differs from it only in ΔY_3, which was re-chosen for the extension. Words are written $x_0, \ldots, x_4$ in hexadecimal, bit 0 is the least significant bit, and S-box inputs are read column-wise with x_0 as the most significant bit, as in the specification.

4 Verification

Against the reference implementation. Independently of the SAT model, we checked each round with the designers' Python implementation `pyascon` [13]. For round r we build a state whose active columns take values v with $S(v) \oplus S(v \oplus a) = b$ for the column differences $a \to b$, fill the other columns at random, apply one full round of the reference permutation (constant addition, S-box layer and linear layer) to the state and to the state XOR ΔX_r, and compare the output difference with ΔX_{r+1} (or with $\Sigma(\Delta Y_4)$ in the last round). All rounds matched in 20 random trials each. We also recovered the S-box table from the reference implementation and checked that it equals the table in the specification.

Table 2: Differences at the input (ΔX_r) and output (ΔY_r) of the S-box layer. #: active S-boxes, w: $-\log_2$ of the probability of the round.

r	ΔX_r $(x_0, \ldots, x_4)$	ΔY_r $(x_0, \ldots, x_4)$	#	w
0	b9460e2000000000	0000000000000000	15	42
	0000004080000001	a000020080000000		
	39460c6080000000	39460c2000000001		
	0000000000000000	0000020000000000		
	0000000000000001	8000024000000000		
1	0000000000000000	0000000000000000	11	39
	a100120481400001	a100024480400001		
	a100120080000001	0000000000000000		
	0000020081000000	0000000000000000		
	a100024480400001	0000104001000000		
2	0000000000000000	0000000000000000	9	36
	2000906081020008	0000000000000000		
	0000000000000000	0000000000000000		
	0000000000000000	0000000000000000		
	2000906081020008	2000906081020008		
3	0000000000000000	0000004000000000	7	21
	0000000000000000	0000104044100440		
	0000000000000000	0000000000000000		
	0000000000000000	0000000000000000		
	0000104044100440	0000100044100440		
4	0000004000080400	0000000000080000	26	88
	8088b24ae4902660	0022082000000000		
	0000000000000000	0088a24880000220		
	0000000000000000	0088a20880000620		
	8022182264982440	8022182264982440		
		rounds 0–3 / rounds 0–4	42 / 68	138 / 226

Machine-checked proof. We transcribed the permutation into Lean 4 (v4.33.1, core library only) from the specification: the S-box layer literally as its bitsliced description, the linear layer and the round constants. The column S-box and its DDT are derived from this definition rather than typed in. All statements below are proved by kernel evaluation (`decide +kernel`); no theorem uses `native_decide`, and each depends at most on the axiom `propext`. (i) The derived S-box equals the table of the specification. (ii) The 12-round permutation maps a fixed input to the output computed with `pyascon`, which ties the Lean definition to the reference implementation. (iii) Both characteristics are valid (every column transition has a non-zero DDT entry, consecutive rounds are linked by Σ, all words are 64-bit values), their first input difference is non-zero, and they have 42 and 68 active S-boxes with the per-round profiles given above. (iv) For every round r an explicit 320-bit state s_r satisfies

$$\mathrm{round}_r(s_r) \oplus \mathrm{round}_r(s_r \oplus \Delta X_r) = \begin{cases} \Delta X_{r+1} & r < R-1, \\ \Sigma(\Delta Y_{R-1}) & r = R-1, \end{cases} \tag{4}$$

where round_r is the full round function including its constant. These pairs realise each round separately. (v) The single pair of the introduction follows all four rounds of the characteristic in Table 2 through rounds 8 to 11, and the active S-boxes counted from the two states alone are 15, 11, 9 and 7 (`pair_follows`, `pair_active`). The whole proof is checked in about 12 s on an Apple M1 Max.

A pair for all rounds. Random trials would need about 2^{138} attempts, but the state values can be chosen directly: a SAT model of the actual values, with the conditions of the characteristic on each column, returned the pair of the introduction in about 3 s (CaDiCaL 1.5.3 through PySAT). The same search returned a pair for each of the 9 windows of four consecutive rounds of the 12-round permutation. This shows that the set of pairs following the characteristic is not empty for these round constants, which the independence assumption in (2) alone does not establish; it does not measure the probability 2^{-138}. For the 5-round characteristic we have only the per-round check above: the same search for one pair following all five rounds gave no answer within 40 minutes, so we do not claim that such a pair exists. This matters: some published collision trails of Ascon have turned out to be invalid [6], and the 5-round characteristic here has not been shown to be free of that problem.

5 Discussion

The new characteristics narrow the 4-round interval for the minimum number of active S-boxes from $[36, 43]$ to $[36, 42]$ and lower the best known 5-round value from 72 to 68. This is a small step: it moves neither the proven lower bound nor the best known weight, which is 107 for 4 rounds and 190 for 5 rounds [5, Table 1], against 138 and 226 here. Fewer active S-boxes do not imply higher probability.

The characteristics do not threaten the security of Ascon. The pair in the introduction relies on free choice of the whole 320-bit state, which no Ascon mode gives an attacker, and the modes use at least 8 rounds of the permutation, while these characteristics cover 4 and 5.

Literature check. "Best known" is limited by what we could read. We looked at the 23 papers citing [4] listed by Google Scholar (October 2026) and searched the Cryptology ePrint Archive and the web again (November 2026); none of what we could read states fewer than 42 or 68 active S-boxes for 4 or 5 rounds. Papers not citing [4] are covered only by those searches, some papers we read only as abstracts, and we could not search Chinese-language databases. Corrections are welcome.

We did not determine the minimum: SAT runs asking directly for at most 41 active S-boxes over 4 rounds (two solvers, about 7.5 h) gave no answer, which says nothing about whether such a characteristic exists.

The search code, the characteristics in machine-readable form, the verification script and the Lean proof are available at `https://github.com/otoinobuo/ascon-trails`.

References

[1] C. Dobraunig, M. Eichlseder, F. Mendel, M. Schläffer. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *Journal of Cryptology* 34(3):33, 2021.

[2] NIST. Ascon-Based Lightweight Cryptography Standards for Constrained Devices. NIST Special Publication 800-232, 2025.

[3] J. Erlacher, F. Mendel, M. Eichlseder. Bounds for the Security of Ascon against Differential and Linear Cryptanalysis. *IACR Transactions on Symmetric Cryptology* 2022(1).

[4] R. H. Makarim, R. Rohit. Towards Tight Differential Bounds of Ascon: A Hybrid Usage of SMT and MILP. *IACR Transactions on Symmetric Cryptology* 2022(3):303–340.

[5] S. El Hirch, S. Mella, A. Mehrdad, J. Daemen. Improved Differential and Linear Trail Bounds for ASCON. *IACR Transactions on Symmetric Cryptology* 2022(4).

[6] G. Liu, S. Li, H. Li, W. Qiu, S. Sun. SAT-Based Space Partitioning and Applications to Ascon-Hash256 Cryptanalysis. Cryptology ePrint Archive, Paper 2025/1542.

[7] Song et al. New Differential-Based Distinguishers for Ascon via Constraint Programming. *IET Information Security*, 2024.

[8] L. Sun, W. Wang, M. Wang. Accelerating the Search of Differential and Linear Characteristics with the SAT Method. *IACR Transactions on Symmetric Cryptology* 2021(1):269–315.

[9] M. Matsui. On Correlation Between the Order of S-boxes and the Strength of DES. EUROCRYPT 1994, LNCS 950.

[10] C. Sinz. Towards an Optimal CNF Encoding of Boolean Cardinality Constraints. CP 2005, LNCS 3709.

[11] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, F. Pollitt. CaDiCaL 2.0. CAV 2024.

[12] A. Ignatiev, A. Morgado, J. Marques-Silva. PySAT: A Python Toolkit for Prototyping with SAT Oracles. SAT 2018.

[13] M. Eichlseder et al. pyascon: Python implementation of Ascon. `https://github.com/meichlseder/pyascon`.