A corrected six-round differential optimum and checked eight- and nine-round bounds for the NSA block cipher ARADI, and Ascon characteristics with fewer active S-boxes

Nobuo Otoi

ba-cybersecurity.com, Tokyo, Japan

Abstract

Tables of trail weights are increasingly filled by automated tools, and a value printed as optimal is only as reliable as the evidence behind it. We re-examine such a table and argue for a simple reporting practice: state for every claim what an independent checker has verified and what is trusted beyond that. *(i)* For ARADI, a block cipher designed by the NSA, Table 4 of the automated CLAASP study, in its ePrint version, prints 84 in bold as the optimal weight of six-round single-key differential trails. A trail of weight **82** exists in the same CLAASP model, checked by fixing it in CLAASP's own models, so 84 is not optimal. We also show that no trail of weight at most 81 exists, with 945 subproblems, each with a DRAT proof checked by `drat-trim`; the clauses are CLAASP's trail clauses with a weight counter, window clauses and a split of our own, and the window bounds are DRAT-checked over CLAASP's complete CNF. With a rotational symmetry of ARADI that we argue on paper and test numerically, the same procedure gives the lower bound 122 for eight rounds (8,770 DRAT-checked subproblems), so the table's entry 122 is optimal. For linear trails, with window bounds that rest on our encoding and on the symmetry, it shows that the eight-round entry 62 is optimal and that the nine-round entry 71, not marked as optimal in the table, is optimal too; a ten-round trail of weight 74 fills an empty entry. Each lower bound holds under the dependences that we list. *(ii)* For the Ascon permutation we give 4- and 5-round differential characteristics with **42** and **68** active S-boxes, against 43 and 72 published, checked against a reference implementation and in Lean 4. They are not proofs of the minimum, and the 4-round one has lower probability than the best known 4-round characteristic. None of the results yields an attack.

Keywords: differential cryptanalysis, trail bounds, SAT solving, DRAT proofs, proof assistant, linear cryptanalysis, ARADI, Ascon

1 Introduction

Differential cryptanalysis [BS91] rates a primitive by the weight (negative binary logarithm of the probability) of its best differential trail, or by the number of differentially active S-boxes, over a reduced number of rounds. These quantities fill the tables that justify round numbers, and they are increasingly produced by automated tools: SAT/SMT and MILP models that search for low-weight trails and, less often, prove that none lighter exists [SWW21, AK18]. Between the solver and the table there is usually no independent check.

Email: otoi@ba-cybersecurity.com (Nobuo Otoi)

Date of this document: 2026-11-18.

A SAT-based lower bound can be given such a check. A modern SAT solver can emit a DRAT proof [WHJ14] of each unsatisfiability it claims, and a separate checker re-derives the empty clause from that proof without trusting the solver. Proofs of this kind are established practice for large combinatorial results [HKM16]. When the clauses fed to the solver are a primitive's trail-validity clauses, a checked UNSAT is evidence that no trail below a given weight exists—evidence whose remaining trusted base (the generator of the clauses, any bounds used for pruning, the checker itself) can be listed explicitly. This paper applies that idea to two primitives and lists the trusted base of each claim.

Contributions.

- **ARADI: a published optimum is not optimal.** Table 4 of [BFG+24], in its ePrint version, prints 84 in bold as the six-round single-key differential weight of ARADI, the block cipher published by the NSA for memory encryption [GMW24]. We exhibit a six-round trail of weight 82 that is valid in the CLAASP model used by [BFG+24] (Table 2); this alone shows that 84 is not optimal. The trail has 41 active S-boxes, all of weight 2, so it also contradicts the statement of Baudrin et al. [BHNS26] that the six-round optimum exceeds twice the minimum number of active S-boxes. Their count of 41 implies the lower bound 82; we certify that bound with an unsatisfiability argument of 945 DRAT-checked subproblems whose trusted base we state in full (Section 4). The same procedure, with a rotational symmetry of ARADI and 8,770 DRAT-checked subproblems, confirms the eight-round entry 122 (Section 4.3). For linear trails the same procedure, with window bounds that we certify for every start index with our own encoding and the symmetry, shows that the eight-round entry 62 and the nine-round entry 71 are optimal, the latter not marked optimal in the table; a ten-round trail of weight 74 is the best we found, and that nothing lighter exists rests on one unproved solver run (Section 4.4).
- **Ascon: two characteristics with fewer active S-boxes.** A 4-round characteristic with 42 active S-boxes and a 5-round one with 68, against 43 and 72 in [MR22], both printed in full and checked in Lean 4 (Section 5).
- **A reporting practice.** For each claim we separate what a DRAT checker or a proof-assistant kernel has verified from what is trusted in addition (Table 1). We do not claim the practice is new in itself; we claim that applying it makes the limits of our own results visible. The corrected six-round entry itself rests on a trail alone.

Limitations. None of the results yields an attack on a full cipher; each concerns a reduced-round trail under the standard Markov and independent-round-key model. The ARADI lower bound is not a proof over CLAASP's clauses alone: the weight counter, the window clauses and the split of the six-round instance are our own code (Section 4). The eight-round bound in addition rests on a rotational-symmetry argument that is written out and tested numerically, not checked by a proof checker. The linear bounds of Table 3 rest on our encoding in the same way, and from four rounds on on the same argument for masks, which we tested numerically only outside CLAASP. The ten-round linear lower bound rests on one solver run, without a proof. The Ascon result improves a count, not a proven bound, and not a probability. The contribution is a corrected table entry, certified ones (differential, eight rounds; linear, eight and nine rounds), a new ten-round linear trail, two characteristics with fewer active S-boxes, and the evidence for them; the techniques are standard.

Related work. The models follow the SAT trail-search method of Sun, Wang and Wang [SWW21], Matsui's bounding conditions [Mat95], the sequential-counter encoding of

Sinz [Sin05], cube-and-conquer [HKWB12], and DRAT proofs [WHJ14]. The gap between trail weights and the probability of differentials is studied in [AK18]. Primitive-specific prior work is [GMW24, BFG+24, BGG+24] for ARADI and [DEMS21, EME22, MR22, HMMD22] for Ascon. Baudrin et al. [BHNS26] compute the exact minimum number of active S-boxes of ARADI for two to twelve rounds with a SAT solver (41 for six rounds, without proofs); since every active S-box of ARADI has weight at least 2, this gives the lower bound 82 for six rounds. They compare it with Table 4 of [BFG+24] and state that for six rounds the optimal weight exceeds twice the minimum number of active S-boxes, that is, they take 84 as the optimum. Our weight-82 trail shows that the bound is tight. Other analyses of ARADI [MMRS25, GMNP25] concern key-recovery attacks with trails over eight and more rounds and, to our knowledge, do not recompute the six-round optimum. Apart from earlier versions of the present preprint [Oto26], we found no earlier published correction of the six-round entry.

Use of AI tools. Generative AI tools were used substantially in this work; Section 7 states how.

2 Preliminaries

Differential trails and weight. For a round function with independent round keys, XOR differences pass key additions unchanged, so a differential characteristic (or trail) is a sequence of XOR differences through the non-linear layers. Under the Markov assumption [LMM91] the probability of a trail is the product of its per-step transition probabilities, and its *weight* is $w = -\log_2 p$. For an S-box-based primitive one also counts *active S-boxes*. We write B_R for the minimum weight (or minimum active-S-box count, where stated) over all non-trivial R-round trails.

Linear trails. A linear trail is a sequence of masks through the rounds, with each S-box mapping a non-zero input mask to a non-zero output mask with non-zero correlation c. Its weight in this paper, as in Table 4 of [BFG+24], is the sum over active S-boxes of $-\log_2 |c|$; for ARADI's S-box each active S-box contributes 1 or 2.

SAT model of a trail. A trail is encoded bit by bit. Difference variables hold the input and output differences of each non-linear layer; the valid transitions of each S-box (its difference distribution table, DDT) or of modular addition (the Lipmaa–Moriai condition [LM01]) are expressed as clauses that block every impossible local transition. Linear layers become XOR constraints.

Weight counting. Per-step weight variables are summed by a sequential counter [Sin05]; a unit clause on its final register bounds the total by k. For a lower bound the encoding must be *complete*: every valid trail of weight at most k must extend to a satisfying assignment. Completeness is a property of the generator and is part of the trusted base of every lower bound below.

Window bounds. Following Matsui [Mat95] and its SAT formulation [SWW21], if every trail over a window of ℓ consecutive rounds has weight at least b, then in a trail of total weight at most k the rounds outside the window have weight at most $k - b$. Clauses stating this for many windows prune the search. They remove no trail of weight at most k *provided the bounds b are correct*, so a lower bound obtained with them is conditional on those bounds.

Splitting. A hard instance is partitioned into subproblems ("cubes") by fixing partial weights [HKWB12]. If the cubes together cover every assignment and each is unsatisfiable, the instance is unsatisfiable. Coverage is a separate obligation.

DRAT proofs. Each unsatisfiable subproblem is solved by a solver emitting a binary DRAT proof [WHJ14], which `drat-trim` re-checks. We accept a run only on the checker's `VERIFIED` verdict, and we say "DRAT-checked" only in that case. `drat-trim` is not itself formally verified; re-checking with a verified checker such as `cake_lpr` [THM21] is possible in principle and was not done.

Tools. The solvers are `kissat` 4.0.4 [B+25]; for Ascon, `CaDiCaL` 1.9.5 [BFFH20] through PySAT [IMMS18]; and for the ten-round ARADI linear search, also the stand-alone `CaDiCaL` 3.0.1. The proof checker is `drat-trim` [WHJ14]. For ARADI the trail-validity clauses come from the CLAASP framework [BGG+24] at commit `f2239d6` (v3.2.0).

3 Procedure and trusted base

The case studies follow one procedure, of which not every step applies to every case:

1. **Model.** Build a CNF of the R-round trail relation with a weight bound.
2. **Upper bound.** Find a trail by descending search and print it. Check it outside the search model.
3. **Lower bound** (ARADI). Assert "weight $\leq k-1$" and show unsatisfiability, splitting into cubes if needed.
4. **Coverage** (ARADI). Show that the cubes omit no assignment.
5. **Proof checking.** Emit and check a DRAT proof for each unsatisfiable instance, and say which instances have none.
6. **Trusted base.** List what the result depends on beyond the checked proofs.

Table 1 gives the outcome of step 6 for every claim in the paper. The Ascon result has no lower-bound part, so steps 3–5 do not apply to it; its evidence is the printed characteristics and their check in a proof assistant.

4 ARADI

ARADI is a 128-bit block cipher with a 256-bit key, designed by Greene, Motley and Weeks of the NSA for memory encryption [GMW24]. Bellini et al. [BFG+24] analysed it automatically with CLAASP [BGG+24]. Their Table 4 lists, for one to ten rounds, the weights of the best differential trails found in the single-key setting with the solver ParKissat-RS. Its caption says "Optimal weights are highlighted in bold", and the text states that the differential trails in the table are optimal up to ten rounds, a trail being optimal if no trail of smaller weight exists. The six-round entry is 84. We use the ePrint version of [BFG+24] as revised on 29 August 2024, the latest at the time of writing. All statements about Table 4 in this paper refer to that version. The proceedings version, whose abstract reports results beyond those of the ePrint version, was not accessible to us. Baudrin et al. [BHNS26] cite the proceedings version, where the table is Table 3, and state that its six-round differential weight exceeds twice their count of 41 active S-boxes, that is, exceeds 82; so that table, too, does not give 82.

Models. CLAASP offers two cipher objects for ARADI. In the *S-box model* (class `AradiBlockCipherSBox`) the weight of a trail is

$$w_{\mathrm{S}} = \sum_{i=1}^{6} \sum_{j=1}^{32} \bigl(4 - \log_2 \mathrm{DDT}(a_{ij}, b_{ij})\bigr),$$

where each active S-box contributes 2 or 3; this is the model of [BFG+24, §6.1, Listing 1.6]. In the *bit-sliced model* (`AradiBlockCipher`) the weight counts active AND gates. All statements below are for the single-key setting, as in Table 4. We do not know which CLAASP version produced Table 4. At the commit we use, CLAASP reproduces the designers' test vector, and its S-box model gives the one-to-five-round optima $2, 8, 14, 32, 50$, which agree with Table 4, and trails of weight 106 and 122 for seven and eight rounds, which equal the Table 4 values. We confirm the optimality of 122 in Section 4.3; for seven rounds, 106 remains an upper bound for us.

4.1 A trail of weight 82: 84 is not optimal

Table 2 gives a six-round trail of weight 82. It was checked in two ways that do not depend on the search that found it: (a) with its seven state differences fixed in CLAASP's unmodified S-box model, the model is satisfiable at weight bound 82 and unsatisfiable at 81; (b) its weight is recomputed as 82 from the round function, without any solver. CLAASP's own search (`SatXorDifferentialModel` with `kissat`) also returns trails of weight 82 directly, for instance at bound 82. Hence 84 is not the optimal six-round weight in the S-box model. This statement does not depend on any lower-bound computation; it does assume that CLAASP's model at our commit is the model of [BFG+24].

The bit-sliced model. In CLAASP's bit-sliced model the same trail also has weight 82, with the same per-round weights: with its state differences fixed, CLAASP's bit-sliced CNF is satisfiable at bound 82 and unsatisfiable at 81. So 84 is not the optimum in that model either. We make no claim about the bit-sliced optimum. A lower-bound computation in a compressed model also gave 82, but it rests on a model-equivalence argument that no tool has checked, and nothing in this paper relies on it.

Why was 84 printed? We do not know. A trail of weight 84 exists, so 84 is a valid upper bound. One candidate explanation—that 84 is the optimum of the bit-sliced model—is excluded by the previous paragraph. We did not reproduce the setup of [BFG+24] (their solver, hardware, or CLAASP version), and we make no claim about the cause.

4.2 No trail of weight 81 or less

The minimum number of active S-boxes reported in [BHNS26], 41 for six rounds, already implies this bound, but as an uncertified solver result. The argument below checks the weight bound directly, with DRAT proofs and the dependences listed in Table 1.

Instance. The CNF has three parts. (1) The component clauses that CLAASP emits for the six-round S-box model, without CLAASP's weight counter. They contain 128 weight variables per round, and the weight of a round is the number of them that are set. (2) A sequential counter over the 768 weight variables $x_1, \ldots, x_{768}$, written by us and encoded in both directions: the clauses state $c_{i,j} \leftrightarrow c_{i-1,j} \vee (x_i \wedge c_{i-1,j-1})$ for every register, so $c_{i,j}$ holds exactly when at least j of the first i weight variables are set. The counter is therefore a definitional extension: every assignment of CLAASP's variables extends to the registers in exactly one way, and the counter by itself removes no trail. The unit clause

$\neg c_{768,82}$ bounds the total weight by 81. (3) Window clauses, written by us. For every window of rounds $[s, e]$ other than the whole range they state that the weight inside the window is at most $81 - p - q$, where p is a lower bound on the weight of the s rounds before the window and q a lower bound on the weight of the $5 - e$ rounds after it. Since the registers are exact, every trail of weight at most 81 satisfies these clauses whenever p and q are valid bounds.

Part (1) was produced by CLAASP's clause generator, called from a short driver script of ours and running under a minimal SageMath shim instead of a full SageMath installation. We checked the output by enumerating the S-box and by comparing, for two rounds, the complete set of 39,168 trails of weight 8 with that of an independently written model; the driver and the shim remain part of the trusted base.

Window bounds. ARADI's linear layer depends on the round index modulo 4, so the minimum weight of an ℓ-round window may depend on its starting index s mod 4. For each $s \in \{0, 1, 2, 3\}$ and $\ell \in \{1, \ldots, 5\}$ we show that every ℓ-round trail starting at index s has weight at least b_ℓ, with $(b_1, \ldots, b_5) = (2, 8, 14, 32, 50)$. Each of these twenty statements is one unsatisfiable instance at weight bound $b_\ell - 1$, with a DRAT proof checked by `drat-trim`. The instances are CLAASP's complete CNF for the ℓ-round cipher, including CLAASP's own weight counter, which enforces a weight of at most the bound (at bound 83, for example, CLAASP's six-round model returns a trail of weight 82). Our counter, window and cube code is not involved; the driver and the shim are. For $s = 0$ the cipher is CLAASP's class unchanged. For $s \in \{1, 2, 3\}$ it is a subclass whose only override passes the round index plus s to CLAASP's linear-layer method; in the single-key setting the key difference is zero, so the round index used by the key schedule plays no role. Two checks support the subclass. For every ℓ, the CNFs for $s = 0, 1, 2, 3$ are pairwise different, so the override takes effect, while shifts of 4 and 8 give a CNF byte-identical to that of the unchanged class and a shift of 5 one identical to $s = 1$. And for every s and ℓ the instance at bound b_ℓ is satisfiable, so the bounds are tight. The twenty instances took 200 seconds to solve and 185 seconds to check in total; the largest proof is 63 MB. The six-round instance uses nine of the twenty bounds: $(s, \ell) = (0, 1), \ldots, (0, 5)$ for the rounds before a window, and $(1, 1), (3, 3), (2, 4), (1, 5)$ for the rounds after it.

Split and coverage. Let w_0 be the weight of round 0 and w_{01} the weight of rounds 0 and 1. A cube fixes both exactly, by unit clauses on the registers at the two round boundaries: $c_{128,a}$, $\neg c_{128,a+1}$, $c_{256,b}$, $\neg c_{256,b+1}$ state $w_0 = a$ and $w_{01} = b$. In any non-trivial trail every round has an active S-box, so $w_0 \geq 2$ and $w_{01} \geq w_0 + 2$. A trail of total weight at most 81 has $w_0 \leq 81 - b_5 = 31$, by the five-round bound at start index 1, and $w_{01} \leq 81 - b_4 = 49$, by the four-round bound at start index 2. The integer pairs with $2 \leq w_0 \leq 31$ and $w_0 + 2 \leq w_{01} \leq 49$ number **945**, one cube each (Figure 1). Coverage thus follows by arithmetic from the window bounds; for six rounds we did not discharge it by a separate DRAT-checked formula.

Result. All 945 cubes are unsatisfiable, each with a DRAT proof checked by `drat-trim` (945 `VERIFIED`, no failure). The campaign took 22.47 CPU-hours (11.88 solving and proof generation, 10.59 checking) and 1 h 54 min of wall time; the hardest cube took 286 s to solve and 261 s to check, with a 307 MB proof. We did not pursue the unsplit instance.

What this does and does not establish. The checked proofs establish that each cube is unsatisfiable. That this means "no six-round trail of weight at most 81 in CLAASP's S-box model" depends further on (i) our code for the counter, the window clauses and the cubes having the properties stated above, which we argued on paper but did not machine-check, (ii) the three-line subclass used for the window bounds at $s \neq 0$, and (iii)

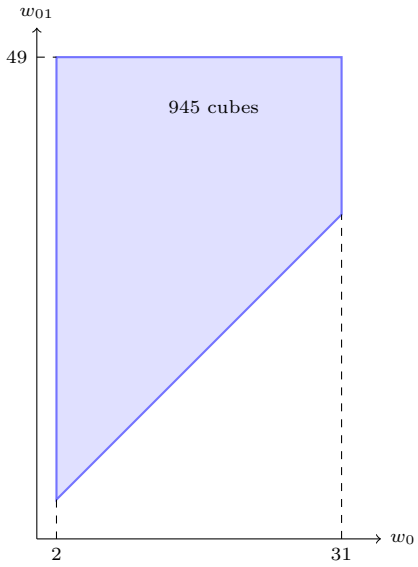


Figure 1: The split of the six-round weight-≤ 81 instance by (w_0, w_{01}). Each integer point of the shaded region is one cube.

the driver and the shim. As a guard against an over-constrained encoding, the instance generated in the same way for bound 82 is satisfiable and returns a trail of weight 82; this is a sanity check, not a proof. Dependence (i) could be removed by splitting CLAASP's complete six-round CNF directly, without our counter; we did not attempt this.

4.3 Eight rounds: a lower bound of 122

Table 4 of [BFG+24] also prints 122 in bold for eight rounds. The count of 60 active S-boxes in [BHNS26] implies only the lower bound 120. We show with the same procedure that 122 is optimal in the S-box model: no eight-round trail of weight at most 121 exists. A trail of weight 122 is returned by CLAASP's own search and is checked as in the six-round case.

Instance. As in Section 4.2, the CNF consists of CLAASP's component clauses for eight rounds, our exact sequential counter over the 1,024 weight variables with the unit clause bounding the total by 121, and weight-window clauses. The window bounds are the twenty DRAT-checked values b_ℓ, $\ell \leq 5$, for every start index; the six-round bound 82 at start index 0 from Section 4.2; and values derived from these by superadditivity, without a solver. In addition, a second exact counter bounds the number of active S-boxes by 60, which removes no trail of weight at most 121 because every active S-box has weight at least 2. Its inputs are 256 of CLAASP's own weight variables: CLAASP encodes the weight of an S-box as a vector whose set bits number its weight, and one bit of that vector is set exactly when the S-box is active. The bounds below use this equivalence in both directions, so it is part of the trusted base; we checked it on the decoded trails, not for every S-box transition. Activity windows use the bounds $1, 4, 7, 16, 25$ for one to five rounds at every start index; each of these twenty bounds is again an unsatisfiable instance at the bound minus one, with a DRAT proof checked by `drat-trim`.

Rotational symmetry. Write each 32-bit word as two 16-bit halves and let σ_t, $0 \leq t \leq 15$, rotate both halves of all four words left by t. Each linear layer L_i of ARADI is

built from XOR and rotations of the two halves, so $L_i \circ \sigma_t = \sigma_t \circ L_i$. The map σ_t applies the same bit permutation to every word, so it permutes the S-box columns and commutes with the S-box layer. In the single-key setting the key difference is zero. Hence rotating every difference of a trail by σ_t gives a trail with the same weight and the same number of active S-boxes in every round. In CLAASP's numbering, columns 0–15 hold the bits of the upper halves and columns 16–31 those of the lower halves; the permutation acts transitively on each of the two sets, so every non-trivial trail is equivalent to one in which column 0 of its first difference is active (class 0), or in which columns 0–15 are inactive and column 16 is active (class 16). The same holds for a window that starts at any index, which is how the class clause is used in the range bounds below. The argument is a statement about the cipher, while the proofs are about CLAASP's CNF; the link between the two is not machine-checked. We tested it numerically: CLAASP's rotation amounts equal the designers'; L_i and σ_t commute on all 32 unit vectors, which suffices by linearity, and on random words; the S-box layer and the full round commute with σ_t on random states; and a weight-122 trail rotated by each t keeps weight 122, both by recomputation from the round function and in CLAASP's fixed-trail model.

Split and coverage. A cube fixes w_0 and w_{01} as in Section 4.2 and adds the clause of one symmetry class. A trail of weight at most 121 has $w_0 \le 121 - 84 = 37$ and $w_{01} \le 121 - 82 = 39$, given that every seven-round trail starting at index 1 has weight at least 84 and every six-round trail starting at index 2 weight at least 82. We obtain these two bounds from activity: with the activity windows and the symmetry split, at most 41 active S-boxes over seven rounds from index 1, and at most 40 over six rounds from index 2, are unsatisfiable (four instances, each DRAT-checked). The 666 pairs with $2 \le w_0 \le 37$ and $w_0 + 2 \le w_{01} \le 39$, in two classes, give 1,332 cubes. That the pairs omit no assignment satisfying the two bounds is stated by a separate formula, unsatisfiable with a checked DRAT proof. Of the 340 cubes with $w_{01} \ge 35$, 107 were solved unsplit; the 233 that had not started seven hours into the run were split further by fixing w_{012}, the weight of rounds 0–2, to each value from $w_{01} + 2$ to $121 - 50 = 71$; the upper end is the five-round bound at start index 3. This gives 7,671 sub-cubes, whose coverage is a second DRAT-checked formula. A split cube counts as refuted only when all its sub-cubes are.

Result. All 1,099 unsplit cubes and all 7,671 sub-cubes are unsatisfiable, each with a DRAT proof checked by `drat-trim` (8,770 `VERIFIED`, no failure), and so are the 26 auxiliary instances: twenty activity bounds, four range bounds and two coverage formulas. The campaign took 583 CPU-hours (324 solving and proof generation, 259 checking; the sum of the wall times of the individual jobs) and 52 h 43 min of wall time on one 12-core machine, with 12 parallel jobs for the first seven hours and 16 afterwards, while single-run searches for linear trails shared the machine. The proofs total 586 GB; the largest is 1.92 GB and took 4,521 s to produce and 3,377 s to check. As a guard against an over-constrained encoding, one cube of the instance generated in the same way for bound 122 ($w_0 = 6$, $w_{01} = 8$, class 0) is satisfiable within 13 seconds and returns a trail of weight 122; this tests one cube only. Together with the trail of weight 122, the eight-round optimum in the S-box model is 122. The Table 4 entry is therefore correct. Given the count of 60 active S-boxes in [BHNS26], which has no proof, the eight-round optimum is larger than twice that count, as they state; the reason they give, that optimal trails must contain sub-optimal transitions, does not apply to every weight-122 trail, since the trail returned by the guard above has 61 active S-boxes, all of weight 2.

Dependences. Beyond the checked proofs, the eight-round result depends on items (i)–(iii) of Section 4.2, now including the activity counter and the activity windows; on CLAASP's encoding of activity described above; on the rotational-symmetry argument

and its link to CLAASP's CNF; and on the six-round bound, which enters as a window bound.

4.4 Linear trails: lower bounds for eight and nine rounds

Table 4 of [BFG+24] also lists the weights of linear trails and marks them optimal up to eight rounds: $1, 4, 7, 16, 25, 41, 53, 62$ for one to eight rounds, 71 for nine rounds without a claim of optimality, and no entry for ten rounds. Section 6.1 of [BFG+24] defines the linear weight through the squared correlation, but the values of Table 4 are on the scale of the absolute correlation (its Table 6 gives correlation 2^{-71} for the nine-round trail); we use the latter scale throughout. We ran CLAASP's SAT linear model in its default setting, which removes the key schedule from the model (independent round keys), with a non-zero plaintext mask; we take this to be the model behind Table 4, since it reproduces every value of its linear column. On this scale an active S-box has weight 1 or 2. Every trail found was checked by recomputing its weight from ARADI's linear approximation table and the transposed linear layers, without CLAASP.

Instance. We apply the procedure of Section 4.3 to this model. The CNF consists of CLAASP's component clauses for the linear model, our exact sequential counter over CLAASP's weight variables with the unit clause bounding the total, and weight-window clauses; there is no activity counter. CLAASP encodes the weight of a linear S-box transition as a vector of four bits whose set bits number its weight; its clauses fix two of the four to zero, so the counter runs over the other two, 64 variables per round, and counts exactly CLAASP's weight. For a non-zero start index we use the three-line subclass of Section 4.2; with the key schedule removed, the round index plays no role for the round keys. The same two hash checks hold for the linear model: for one to six rounds the CNFs for start indices 0–3 are pairwise different, and shifts of 4 and 8 (5) give a CNF byte-identical to that of index 0 (1).

Rotational symmetry. The argument of Section 4.3 carries over to masks. Mask propagation through L_i uses its transpose $L_i^\top$, which is built from XOR and the inverse rotations of the two halves, so $L_i^\top \circ \sigma_t = \sigma_t \circ L_i^\top$. The S-box layer is handled as before, now with the linear approximation table. With independent round keys, the mask of a round key equals the state mask at its XOR and rotates with it, and a non-zero plaintext mask stays non-zero. So the same two classes cover every non-trivial linear trail. The numerical tests are weaker than in the differential case: $L_i^\top$, as implemented in our checker, and σ_t commute on all unit vectors, and each of the nine optimal trails for seven, eight and nine rounds, rotated by each t, keeps its weight and active count in every round, recomputed without CLAASP. We did not repeat the comparison of rotation amounts or the fixed-trail test inside CLAASP's linear model, so the link to CLAASP's CNF rests on the written argument and on the shared rotation table of the differential case.

Window bounds. The linear window bounds depend on the start index from six rounds on. We computed them for every start index s and $2 \le r \le 7$ rounds, level by level (for one round the bound 1 is immediate): at each level a trail of weight w is found, and the instance with bound $w - 1$, whose windows use the bounds of the levels below, is unsatisfiable, for each of the two symmetry classes from four rounds on and as one instance for two and three rounds. These 40 instances carry DRAT proofs checked by `drat-trim`; six further proofs from intermediate descent steps are not used. Unlike the differential window bounds of Section 4.2, these instances contain our counter and window clauses and, from four rounds on, the symmetry split. For eight rounds and every start index

we then ran a cube campaign as below, at the weight of the best trail found minus one. Table 3 gives the values.

Split and coverage. A cube fixes the weights w_0 of round 0 and w_{01} of rounds 0–1 and adds the clause of one symmetry class. For nine rounds at start index 0 and bound 70, a trail has $w_0 \le 70 - B_1(8) = 10$ and $w_{01} \le 70 - B_2(7) = 20$; with $1 \le w_0$ and $w_0 + 1 \le w_{01}$ this gives 145 pairs and 290 cubes. The eight-round campaigns are built in the same way, with 252, 232 and 264 cubes for $s = 1, 2, 3$. The campaign for $s = 0$ was generated first, with seven-round window values derived by superadditivity from the six-round bounds (42, 43, 43, 42 for $s = 0, \ldots, 3$) in place of the later certified ones; these are weaker but valid, and give $w_0 \le 61 - 43 = 18$, $w_{01} \le 61 - 42 = 19$ and 342 cubes. Derived values also stand in for $B_1(7)$ (used by a window) and $B_3(7)$ (not used) in the campaign for $s = 1$, for $B_1(7)$ (not used) in that for $s = 2$, and for $B_2(8)$ and $B_3(8)$ (not used) in the nine-round campaign. In each campaign a separate formula, unsatisfiable with a checked DRAT proof, states for each class that the pairs omit no assignment of the instance; the two range bounds are window clauses of the instance itself.

Result. All 1,380 cubes, the ten coverage formulas and the 40 level instances are unsatisfiable, each with a DRAT proof checked by `drat-trim`; no cube needed a further split. The eight-round result at $s = 0$ uses the 342 cubes of its campaign; the nine-round result uses that campaign, the 252 cubes for $s = 1$ and the 290 nine-round cubes. Two checker runs at level seven were killed by the kernel when the machine ran out of memory; the instances were solved again, and the new proofs, of the same size and SHA-256 hash as the first, were accepted. The cube campaigns took 100 CPU-hours (62 solving and proof generation, 38 checking) and the level instances 8, within 23 hours of wall time on the 12-core machine of Section 4.3; the nine-round campaign took 9 h 21 min of wall time. The proofs total 262 GB; the largest is 3.2 GB (seven rounds, $s = 0$). As a guard against an over-constrained encoding, one cube of the nine-round instance generated in the same way for bound 71 ($w_0 = 5$, $w_{01} = 10$, class 0) is satisfiable within 138 seconds and returns a trail of weight 71; this tests one cube only. Together with the trails, and under the dependences listed below, the values of Table 3 are optimal. In particular, the eight-round entry 62 of Table 4 is optimal, and so is the nine-round entry 71, which that table does not mark as optimal. The nine-round trail has 71 active S-boxes, all of weight 1; Baudrin et al. [BHNS26] give, without a proof, 67 as the minimum number of linearly active S-boxes for nine rounds.

Ten rounds. A separate descent on CLAASP's complete CNF for ten rounds, without a split, without symmetry and without clauses of ours, found a trail of weight 74, an entry Table 4 leaves open, with per-round weights $3, 1, 3, 9, 22, 20, 9, 3, 1, 3$ and active S-boxes $3, 1, 3, 9, 21, 18, 9, 3, 1, 3$. A single `CaDiCaL` 3.0.1 run on the same CNF reports bound 73 unsatisfiable after 26.2 h (`kissat` did not reach that bound), with no proof, measured while other campaigns shared the machine. By superadditivity the certified values give the lower bound $B_0(9) + B_1(1) = 72$. So the ten-round optimum lies between 72 and 74, and 74 is the best trail found, with solver evidence that nothing lighter exists.

Dependences. Beyond the checked proofs, the linear bounds depend on our counter, window and cube generators and the level driver; on CLAASP's encoding of the linear weight described above; on the three-line subclass; on CLAASP's default linear model being the model of Table 4 for the comparison with it; and, from four rounds on, on the rotational-symmetry argument for masks and its link to CLAASP's CNF.

5 Ascon

Ascon [DEMS21] is the family standardised by NIST in SP 800-232 [NIS25]. Its 320-bit permutation applies a 5-bit S-box to each of 64 columns and a linear layer $\Sigma_i(x_i) = x_i \oplus (x_i \ggg r_{i,1}) \oplus (x_i \ggg r_{i,2})$ to each of five words. The number of differentially active S-boxes over R rounds is a standard yardstick. For $R \leq 3$ the minimum is $1, 4, 15$ [DEMS21]. For $R = 4$ the designers reported a characteristic with 44 [DEMS21], improved to 43 [MR22], and the proven lower bound is 36 [EME22]. For $R = 5$ the best published counts are 78 [DEMS21] and 72 [MR22]. Bounds on trail *weight*, a different quantity, were improved in [HMMD22].

Result. Table 5 gives a 5-round characteristic with 68 active S-boxes (per round $15, 11, 9, 7, 26$) whose first four rounds form a 4-round characteristic with 42. To our knowledge, as of November 2026, these are below the best published counts of 43 and 72 (Table 4). The 4-round interval for the minimum becomes $[36, 42]$.

What the result is not. It is not a proof of the minimum: a bounded search for 41 or fewer active S-boxes over four rounds did not finish within our budget. It is not an improvement in probability: the 4-round characteristic has weight 138, whereas the best known 4-round characteristic has weight 107 [HMMD22]; the 5-round one has weight 226. Fewer active S-boxes do not imply higher probability (the 43-S-box characteristic of [MR22] likewise had lower probability than the 44-S-box one). The 5-round characteristic is not the output of a 5-round search; it extends the 4-round one by one round. The result says nothing about the security of Ascon, whose permutation has 12 rounds in initialisation and finalisation.

Search. A bit-level SAT model holds the S-box-layer input and output differences. Each column's DDT relation is encoded by blocking clauses (169 per S-box after greedy shortening), the linear layer by XOR constraints, and an activity variable per column feeds a sequential counter [Sin05] with window bounds [Mat95, SWW21] from the known minima $1, 4, 15$ for one to three rounds. Rotational symmetry is removed by forcing column 0 of the first difference to be active. The model has 16,603 variables and 107,432 clauses. Descending from a loose bound, CaDiCaL 1.9.5 (via PySAT, one core) passed through $59, 58, 52, 49, 47, 46, 45$ and then 42 active S-boxes. For the fifth round, the last S-box layer of the 4-round characteristic has 7 active columns; enumerating all 8^7 output differences and applying Σ gives a minimum of 26 active S-boxes in round five. Extending backward instead costs 57.

Checks. Independently of the SAT model, each round of the characteristic was compared with the reference implementation `pyascon` [E+24] on 20 random state pairs per round, and the S-box table was recovered from that implementation and matched to the specification. In addition, the permutation was transcribed from the specification into Lean 4 (core library only), and the following statements were proved by kernel evaluation (`decide +kernel`, without `native_decide`): the S-box derived from the round function equals the specification table; the 12-round permutation maps a fixed input to the output computed by `pyascon`; both characteristics are valid, have a non-zero first difference, and have 42 and 68 active S-boxes with the stated per-round counts, each round being realised by an explicit pair of 320-bit states. The development checks in about 12 seconds. What remains trusted is the transcription of the specification into Lean, which the fixed-input test addresses only partially.

6 Reproducibility

The artefacts are provided as anonymised supplementary material with this submission. A separate file gives, for the published values we compare with, the table, section and page of the source where each is printed, and marks those we did not check against the source.

ARADI. The driver for CLAASP's model, the generator of the hybrid CNF, the cube generator and runner, the fixed-trail checker, the trails, and per-cube tables with solving time, checker verdict and the SHA-256 hash of each proof. The DRAT proofs themselves (36.3 GB for the S-box campaign) were deleted after checking and are not shipped; they can be regenerated in about 12 CPU-hours. CLAASP is pinned at commit `f2239d6` (v3.2.0), `kissat` at 4.0.4, and `drat-trim` at commit `2e3b2dc`. For the twenty window-bound instances the material contains the generator and runner, the result table, the `drat-trim` logs, the solver output for the satisfiable side, and the table of CNF hashes behind the subclass checks. For eight rounds: the instance and auxiliary generators, the cube and sub-cube generators, the scheduler, the per-instance table with solving and checking time, verdict and SHA-256 of each proof, the coverage formulas, the written symmetry argument with the script and output of its numerical checks, and the weight-122 trails. The 586 GB of proofs were deleted after checking. For linear trails: the descent driver, the per-step logs and the trails with their recomputed weights; for the certified eight- and nine-round bounds, the instance generator, the level driver, the cube and coverage generators, the scheduler, per-instance tables with time, verdict and SHA-256 of each proof, the window-bound files, the written argument and the symmetry check. The 262 GB of proofs were deleted after checking.

Ascon. The SAT model, the two characteristics in machine-readable form, the `pyascon` comparison script, and the Lean 4 development.

7 Use of AI tools

An AI coding assistant (Claude Code, Anthropic) was used in this work as follows. Appendix A gives the models, the prompts and the editing.

1. It wrote the code that generates the SAT models for ARADI and Ascon.
2. It wrote the code for the weight counter, the window clauses and the split into cubes.
3. It ran the solvers and `drat-trim` and compiled the result tables.
4. It wrote an S-box model of ARADI that is independent of CLAASP, used for cross-checks.
5. It wrote the checkers for trails: recomputation of the weight from the round function, and checks with the differences fixed in the model.
6. It wrote the three-line subclass of CLAASP's cipher class used for the window bounds, and the checks of that subclass.
7. It wrote the rotational-symmetry argument for ARADI and the script that tests it numerically.
8. It wrote the Lean 4 development.
9. It searched for prior work and checked the relevant passages in the sources.
10. It drafted all sections of this paper, including the tables and the figure.
11. A second instance acted as an adversarial reviewer of the drafts, and the assistant translated and anonymised the supplementary material.

The evidence is not uniform. The written arguments outside the DRAT proofs (the rotational symmetry, the exactness of the counter, the ranges and coverage of the splits) were drafted by the assistant and examined by a second instance acting as reviewer and by numerical tests; no proof assistant has checked them. The ARADI lower bound and the window bounds it uses are backed by DRAT proofs checked by `drat-trim`, with the remaining dependences listed in Table 1; so are the eight-round bound and the linear bounds of Table 3, which in addition rest on the symmetry argument. The ten-round linear lower bound rests on a solver's answer. The Ascon characteristics are checked by the Lean kernel.

8 Conclusion

The six-round single-key differential weight of ARADI printed as optimal in [BFG+24] is not optimal: a trail of weight 82 exists in the same model, with 41 active S-boxes of weight 2, so the lower bound implied by the active-S-box count of [BHNS26] is tight. With the dependences stated in Section 4.2, a DRAT-checked argument confirms that 82 is the optimum. The same procedure, with a rotational symmetry, confirms the eight-round entry 122. For linear trails it confirms that 62 is optimal for eight rounds and shows that 71 is optimal for nine; for ten rounds we find a trail of weight 74, against the lower bound 72 that follows from the nine-round optimum, with an unproved solver answer that nothing lighter than 74 exists. For Ascon, two characteristics lower the best published active-S-box counts for four and five rounds from 43 to 42 and from 72 to 68, without improving the best known probabilities. For the differential and the eight- and nine-round linear results on ARADI and for Ascon, the evidence can be re-checked without trusting the search that produced it, and Table 1 says what must still be trusted. Further work would strengthen the ARADI results: splitting CLAASP's complete CNF directly, so that no encoding of ours is involved; checking mechanically that the symmetry maps CLAASP's clause set to itself; producing a proof for the ten-round linear bound; and re-checking all proofs with a formally verified checker.

References

[AK18] Ralph Ankele and Stefan Kölbl. Mind the gap — a closer look at the security of block ciphers against differential cryptanalysis. In *SAC 2018*, volume 11349 of *LNCS*, pages 163–190, 2018.

[B+25] Armin Biere et al. Kissat SAT solver, release 4.0.4, 2025. URL: https://github.com/arminbiere/kissat.

[BFFH20] Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximilian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT competition 2020. In *Proceedings of SAT Competition 2020: Solver and Benchmark Descriptions*, pages 51–53, 2020. Version 1.9.5 used.

[BFG+24] Emanuele Bellini, Mattia Formenti, David Gérault, Juan Grados, Anna Hambitzer, Yun Ju Huang, Paul Huynh, Mohamed Rachidi, Raghvendra Rohit, and Sharwan K. Tiwari. CLAASPing ARADI: Automated analysis of the ARADI block cipher. In *INDOCRYPT 2024*, pages 90–113, 2024. Cryptology ePrint Archive, Paper 2024/1324, revised 29 August 2024. URL: https://eprint.iacr.org/2024/1324, doi:10.1007/978-3-031-80311-6_5.

[BGG+24] Emanuele Bellini, David Gerault, Juan Grados, Yun Ju Huang, Rusydi Makarim, Mohamed Rachidi, and Sharwan Tiwari. CLAASP: A crypto-

graphic library for the automated analysis of symmetric primitives. In *SAC 2023*, volume 14201 of *LNCS*, 2024. Software: https://github.com/Crypto-TII/claasp.

[BHNS26] Jules Baudrin, Rachelle Heim Boissier, Mahnoor Naseer, and François-Xavier Standaert. From simple mathematical upper bounds to tight tool-based bounds for the minimum number of active S-boxes in LS-designs. *IACR Communications in Cryptology*, 3(3), 2026. doi:10.62056/a3ksa66bm.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3):33, 2021.

[E+24] Maria Eichlseder et al. pyascon: Python implementation of Ascon, 2024. URL: https://github.com/meichlseder/pyascon.

[EME22] Johannes Erlacher, Florian Mendel, and Maria Eichlseder. Bounds for the security of Ascon against differential and linear cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2022(1), 2022.

[GMNP25] Shibam Ghosh, Bastien Michel, and María Naya-Plasencia. Differential-MITM attack on 14-round ARADI. Cryptology ePrint Archive, Paper 2025/1918, 2025. URL: https://eprint.iacr.org/2025/1918.

[GMW24] Patrick Greene, Mark Motley, and Bryan Weeks. ARADI and LLAMA: Low-latency cryptography for memory encryption. Cryptology ePrint Archive, Paper 2024/1240, 2024. URL: https://eprint.iacr.org/2024/1240.

[HKM16] Marijn J. H. Heule, Oliver Kullmann, and Victor W. Marek. Solving and verifying the Boolean Pythagorean triples problem via cube-and-conquer. In *SAT 2016*, volume 9710 of *LNCS*, pages 228–245, 2016.

[HKWB12] Marijn J. H. Heule, Oliver Kullmann, Siert Wieringa, and Armin Biere. Cube and conquer: Guiding CDCL SAT solvers by lookaheads. In *HVC 2011*, volume 7261 of *LNCS*, pages 50–65, 2012.

[HMMD22] Soundes El Hirch, Silvia Mella, Alireza Mehrdad, and Joan Daemen. Improved differential and linear trail bounds for ASCON. *IACR Transactions on Symmetric Cryptology*, 2022(4), 2022.

[IMMS18] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT 2018*, volume 10929 of *LNCS*, pages 428–437, 2018.

[LM01] Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In *FSE 2001*, volume 2355 of *LNCS*, pages 336–350, 2001.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *EUROCRYPT '91*, volume 547 of *LNCS*, pages 17–38, 1991.

[Mat95] Mitsuru Matsui. On correlation between the order of S-boxes and the strength of DES. In *EUROCRYPT '94*, volume 950 of *LNCS*, pages 366–375, 1995.

[MMRS25] Surajit Mandal, Sandip Kumar Mondal, Raghvendra Rohit, and Santanu Sarkar. Improved differential cryptanalysis of ARADI. Cryptology ePrint Archive, Paper 2025/1976, 2025. URL: https://eprint.iacr.org/2025/1976.

[MR22] Rusydi H. Makarim and Raghvendra Rohit. Towards tight differential bounds of Ascon: A hybrid usage of SMT and MILP. *IACR Transactions on Symmetric Cryptology*, 2022(3):303–340, 2022.

[NIS25] NIST. Ascon-based lightweight cryptography standards for constrained devices. Technical Report SP 800-232, National Institute of Standards and Technology, 2025.

[Oto26] Nobuo Otoi. The six-round differential optimum of ARADI is 82. Zenodo, 2026. doi:10.5281/zenodo.24025026.

[Sin05] Carsten Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *CP 2005*, volume 3709 of *LNCS*, pages 827–831, 2005.

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Accelerating the search of differential and linear characteristics with the SAT method. *IACR Transactions on Symmetric Cryptology*, 2021(1):269–315, 2021.

[THM21] Yong Kiam Tan, Marijn J. H. Heule, and Magnus O. Myreen. cake_lpr: Verified propagation redundancy checking in CakeML. In *TACAS 2021*, volume 12652 of *LNCS*, pages 223–241, 2021.

[WHJ14] Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt Jr. DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In *SAT 2014*, volume 8561 of *LNCS*, pages 422–429, 2014.

A Use of generative AI: details

Sections affected. All sections, tables and appendices of this paper; the code in the supplementary material; the Lean 4 development.

Tools and versions. Claude Code (Anthropic) with the models Claude Opus 5.5, Claude Opus 4.8, Claude Fable 5.1 and Claude Fable 5.2, October to November 2026.

Prompts. There was no single prompt. The tool was directed interactively, in Japanese, by short instructions such as “make ARADI the main result”, “check the submission rules from primary sources” and “mention the NSA once, early on”.

Post-generation editing. None by hand. Revisions were made by further instructions to the tool.

Table 1: Claims, their evidence, and what is trusted beyond the independent check.

Claim	Evidence	Independent check	Trusted in addition
ARADI, 6 rounds: a trail of weight 82 exists (84 is not optimal)	trail of Table 2	trail fixed in CLAASP's S-box and bit-sliced models; weight recomputed from the round function	that CLAASP's models at the commit we use are those of [BFG+24]
ARADI, 6 rounds, S-box model: no trail of weight ≤ 81	945 unsatisfiable cubes	DRAT, `drat-trim`: all 945 cubes and all 20 window bounds	our counter, window and cube generator; a three-line subclass of CLAASP's cipher class used for the window bounds; our driver script and a SageMath shim
ARADI, 8 rounds, S-box model: no trail of weight ≤ 121	1,099 cubes and 7,671 sub-cubes, all unsatisfiable	DRAT, `drat-trim`: all 8,770, and 26 auxiliary instances (activity bounds, range bounds, coverage)	as for six rounds, plus our activity counter and CLAASP's encoding of activity; the rotational-symmetry argument (not machine-checked); the six-round bound
ARADI, linear, up to 8 rounds (every start index) and 9 rounds: optimal weights of Table 3	trails; 40 unsatisfiable level instances and 1,380 unsatisfiable cubes	trail weights recomputed from the LAT; DRAT, `drat-trim`: all level instances, cubes and 10 coverage formulas	our counter, window, cube generator and level driver (also inside the window bounds); our driver script and the SageMath shim; CLAASP's linear model and its weight encoding; the three-line subclass; for $r \geq 4$ the rotational-symmetry argument for masks (not machine-checked); for the comparison, that CLAASP's default linear model is that of [BFG+24]
ARADI, linear, 10 rounds: a trail of weight 74; none lighter	trail; one UNSAT run at 73	trail weight recomputed from the LAT; none for the UNSAT side	`CaDiCaL`'s answer at 73; CLAASP's linear model; our driver script and the SageMath shim
Ascon: characteristics with 42 (4 rounds) and 68 (5 rounds) active S-boxes	characteristic of Table 5	Lean 4 kernel (validity and active-S-box counts); comparison with `pyascon`	our transcription of the specification into Lean; the weights 138 and 226, recomputed by our script only

Table 2: A six-round single-key differential trail of ARADI of weight 82. State differences (w, x, y, z) at the input of the S-box layer of each round, and after the last round; 32-bit words in hexadecimal. Rounds are numbered from 1 here and from 0 in the text.

round	w	x	y	z	weight
1	`00000000`	`2854044a`	`00000000`	`00000000`	18
2	`00000000`	`08040040`	`00000000`	`00000000`	6
3	`00000000`	`18260851`	`00000000`	`00000000`	18
4	`16062820`	`16022820`	`00000000`	`00000000`	16
5	`00040424`	`00000000`	`00000000`	`00000000`	8
6	`210d2005`	`00000000`	`00000000`	`00000000`	16
out	`3c892ec7`	`00000000`	`1c813ec7`	`00000000`	
				total	82

Table 3: Linear weights $B_s(r)$ of ARADI: the minimum weight of an r-round linear trail whose linear layers start at index s (absolute-correlation scale, CLAASP's default linear model). Each value is the weight of a trail found and checked; for $r \geq 2$ the bound one below is refuted with DRAT proofs checked by `drat-trim`, and for $r = 1$ it is immediate. The refutations use our encoding and, for $r \geq 4$, the symmetry argument (Table 1). For $r \leq 5$ the values do not depend on s. The row $s = 0$ equals the linear column of Table 4 of [BFG+24].

r	1	2	3	4	5	6	7	8	9
$s = 0$	1	4	7	16	25	41	53	62	71
$s = 1$	1	4	7	16	25	40	53	60	–
$s = 2$	1	4	7	16	25	42	50	61	–
$s = 3$	1	4	7	16	25	40	52	62	–

Table 4: Differentially active S-boxes of the Ascon permutation. The 5-round lower bound shown is the trivial one implied by the 4-round bound.

Rounds	proven lower bound	fewest active S-boxes in a published characteristic
3	15 [DEMS21]	15 [DEMS21]
4	36 [EME22]	44 [DEMS21], 43 [MR22], **42** (this work)
5	37 (trivial)	78 [DEMS21], 72 [MR22], **68** (this work)

Table 5: The 5-round Ascon characteristic. Differences at the input (ΔX_r) and output (ΔY_r) of the S-box layer of round r, words $x_0, \ldots, x_4$ in hexadecimal, bit 0 least significant. #: active S-boxes; w: weight of the round. Rounds 0–3 form the 4-round characteristic.

r	ΔX_r	ΔY_r	#	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