NSA's SPECK128: an exhaustive SAT proof, checked with two independent encodings, that the minimum weight of 12-round differential characteristics is 66

Nobuo Otoi
ba-cybersecurity.com, Tokyo, Japan
otoi@ba-cybersecurity.com
ORCID: 0009-0001-1445-3612

Version 1.1, November 2026

Abstract

SPECK is a family of lightweight block ciphers designed by the U.S. National Security Agency (NSA) and standardized in ISO/IEC 29167-22 for RFID. For SPECK128, the minimum weight B_R of a non-trivial R-round XOR-differential characteristic (Lipmaa–Moriai transition probabilities, Markov assumption) is known up to eleven rounds, with $B_{10} = 49$ and $B_{11} = 58$; for twelve rounds a characteristic of weight 66 was known, so $58 \le B_{12} \le 66$. We show by an exhaustive SAT computation that no twelve-round characteristic has weight 65 or less, so $B_{12} = 66$, conditional on the published values $B_1, \ldots, B_{11}$ that enter Matsui's bounding conditions. The SAT model of Sun, Wang and Wang is split by partial weights into 272 subproblems, all shown unsatisfiable by kissat in 66.1 process-days. Nine further SAT problems show that the split misses no assignment; they and the 89 shortest subproblems carry DRAT certificates verified by `drat-trim`. We print a weight-66 characteristic and check it inside the same model. A second, separately written encoding following Wang et al., split into 407 pieces and solved by CaDiCaL, reaches the same answer: every piece is unsatisfiable.

1 Introduction

SPECK [1] is a family of lightweight ARX block ciphers designed by the U.S. National Security Agency (NSA) for constrained devices. It is standardized in ISO/IEC 29167-22 as a crypto suite for RFID air interfaces [2], and SPECK128/128 is one of the parameter sets there. SPECK128 has 64-bit words and 32 to 34 rounds; one round maps (x, y) to

$$x' = \big((x \ggg 8) \boxplus y\big) \oplus k, \qquad y' = (y \lll 3) \oplus x',$$

where $\boxplus$ is addition modulo 2^{64}, $\ggg$ and $\lll$ are rotations, and k is the round key. In differential cryptanalysis [3] one studies characteristics, sequences of XOR differences through the rounds. With independent round keys, XOR differences pass the key addition unchanged, the only non-linear step is the addition, and under the Markov assumption [4] the probability of a characteristic is the product of the round transition probabilities. Its weight is $w = -\log_2 p$. We write B_R for the minimum weight over all non-trivial R-round characteristics (input difference not zero) and set $B_0 = 0$.

For SPECK128, Sun, Wang and Wang determined $B_1, \ldots, B_9 = 0, 1, 3, 6, 10, 15, 21, 30, 39$ with a SAT model of Matsui's bounding conditions [5]. Wang et al. obtained $B_{10} = 49$ [6], and Kurahara, Sakamoto and Isobe obtained $B_{10} = 49$ and $B_{11} = 58$ [7]. For twelve rounds the best characteristic they found has weight 66, listed in the row of their Table 4 that is not certified optimal. So $58 \le B_{12} \le 66$.

Contribution. We close this gap.

Proposition 1. *Assume $B_1, \ldots, B_{11} = 0, 1, 3, 6, 10, 15, 21, 30, 39, 49, 58$. Then every non-trivial twelve-round XOR-differential characteristic of SPECK128 has weight at least 66 under the Lipmaa–Moriai weight, so $B_{12} = 66$.*

The lower bound is a computer-assisted proof by exhaustive SAT solving (§§2–4); the upper bound is the characteristic of Table 2. The lower bound was computed a second time with a separately written encoding and a different solver, with the same result (§5). The methods are standard; what is new is the value. A literature check on 11 November 2026 (IACR ePrint, arXiv, works citing [7] in OpenAlex and Semantic Scholar, the SCIS 2026 programme) found no earlier result on twelve rounds.

2 Model

We use the model of [5]. Rounds are numbered $0, \ldots, 11$; the input difference of round r is (x_r, y_r) and (x_{12}, y_{12}) is the output. Bit 0 is the least significant bit, $\ll 1$ is a shift by one position that drops the top bit and fills in a zero, and $\mathrm{eq}(\alpha, \beta, \gamma)$ marks the positions where the three words agree.

Addition. By Lipmaa and Moriai [8], input differences α, β and output difference γ of $\boxplus$ are compatible if and only if

$$\mathrm{eq}(\alpha \ll 1, \beta \ll 1, \gamma \ll 1) \wedge (\alpha \oplus \beta \oplus \gamma \oplus (\beta \ll 1)) = 0,$$

and then the weight is $\mathrm{HW}\big(\neg\mathrm{eq}(\alpha, \beta, \gamma) \wedge (2^{63} - 1)\big)$. In round r, $\alpha = x_r \ggg 8$, $\beta = y_r$, $\gamma = x_{r+1}$, and $y_{r+1} = (y_r \lll 3) \oplus x_{r+1}$. We encode the compatibility condition bit by bit.

Weight. For each round and each of the 63 lower bit positions there is a weight variable w, with clauses forcing $w = 1$ when the three bits disagree. The converse is not forced: a solver may set $w = 1$ where the bits agree. One sequential counter [9] runs over the 756 weight variables in round order, with variables $c_{i,j} \Leftrightarrow c_{i-1,j} \vee (w_i \wedge c_{i-1,j-1})$ in both directions, so $c_{i,j}$ holds exactly when at least j of the first i weight variables are 1. A unit clause bounds the total by $k = 65$, and a clause on (x_0, y_0) excludes the zero input difference.

Matsui's bounding conditions. For every window of rounds $s..e$ other than 0..11, the weight variables of the window must sum to at most $k - B_s - B_{11-e}$; this is encoded with the same counter as $c_{\mathrm{end}(e),t} \Rightarrow c_{\mathrm{start}(s),t-m}$. These conditions remove no characteristic of weight at most k: the round map is a bijection that sends zero only to zero, so every prefix and suffix of a non-trivial characteristic is non-trivial and weighs at least B_s and B_{11-e}. B_{11} enters the bounds on round 0 alone and on round 11 alone (each at most $65 - 58 = 7$), and B_{10} those on rounds 0–1, round 1, rounds 10–11 and round 10 (each at most 16). This is where the assumption of Proposition 1 matters most: the characteristic of Table 2 is one round of weight 8 followed by eleven rounds of weight 58, and a smaller B_{11} would loosen exactly these bounds.

The base formula has 50 172 variables and 204 759 clauses.

Lemma 1. *If a non-trivial twelve-round characteristic of weight at most 65 exists, the base formula has a satisfying assignment in which every weight variable equals 1 exactly where the three bits disagree.*

Proof. Set the difference variables from the characteristic and each w to its canonical value. The compatibility and XOR clauses hold because the characteristic is valid; the clauses forcing w hold by construction. The counter variables then have their defined values, the counted sum

is the true weight, which is at most 65, and by the argument above every window satisfies its Matsui bound. □

So unsatisfiability of the base formula, or of every piece of a split that covers all its assignments, proves the proposition. The split below fixes counter values, that is sums of weight variables; under the canonical assignment these are true partial weights, and the coverage check is over all assignments, so it covers the canonical one.

3 Split and coverage

Let j be the counter value after the first six rounds (378 weight variables) and a the value after the first three rounds (189). Fixing j takes the two unit clauses $c_{378,j}$ and $\neg c_{378,j+1}$, and likewise for a. The window 0..5 bounds $j \leq 65 - B_6 = 50$, and since any six rounds weigh at least $B_6 = 15$, under the assignment of Lemma 1 a characteristic has $15 \leq j \leq 50$: 36 cases (the coverage check below confirms this range for every assignment of the model). The eight middle cases $32 \leq j \leq 39$ were split again, to spread the work, by $3 \leq a \leq j - 3$, giving $\sum_{j=32}^{39}(j-5) = 244$ sub-cases; the other 28 cases were solved whole. In total there are 272 pieces.

To check that this split misses no assignment, we built from the same code one formula "base formula and $j \notin \{15, \ldots, 50\}$" and, for each $j = 32, \ldots, 39$, "base formula, j fixed, and $a \notin \{3, \ldots, j-3\}$". All nine are unsatisfiable. kissat 4.0.4 [10, 11] wrote a DRAT proof for each, and `drat-trim` [13] verified all nine (the first in 9.7 s, the others in about 0.2 s each).

Every solved file can be regenerated from the code. We regenerated all 289 files (36 cases, 244 sub-cases, 9 coverage formulas) and compared SHA-256 hashes of the DIMACS text with all 325 copies present on the three machines, among them the 282 that were solved; all are identical. The hashes are listed per file in the archive.

4 Computation

Every piece was solved by kissat 4.0.4 in `--unsat` mode, one single-threaded process per piece, between 28 October and 6 November 2026. All 272 pieces are unsatisfiable. Table 1 gives the elapsed time of each process, summed; the machines ran several pieces at once and were in other use, so these are process-days of wall-clock time, not CPU time.

Table 1: Where the 272 pieces were solved (kissat 4.0.4, one process per piece).

machine	pieces	count	time per piece	process-days
Apple M1, 4 at a time	$j = 15..31$	17	61 s – 4.3 d	12.5
AMD Ryzen 9 9900X, up to 20 at a time	$j = 32..39$, all a	244	15 s – 1.5 d	53.1
Apple M1 Max, 5 at a time	$j = 40..50$	11	46 s – 4.7 h	0.5
total		272		66.1

In addition, the case $j = 32$ was solved whole on the Apple M1 (7.6 days, unsatisfiable, ending on 7 November), which agrees with its 27 sub-cases. Three of the Apple M1 times ($j = 30, 31, 32$) are taken from file timestamps because the runner was stopped and restarted.

Certificates. We did not keep DRAT proofs for the long pieces. For the 89 pieces that took at most 30 minutes each, kissat 4.0.4 wrote a DRAT proof afresh and `drat-trim` verified every one. The other $272 - 89$ pieces rest on kissat's answer alone.

5 Checks of the model

The upper bound inside the model. Table 2 gives a twelve-round characteristic of weight 66, obtained by prepending one round of weight 8 to the eleven-round characteristic of weight 58 that kissat found in the model check below. Its validity and weight were recomputed by the checker of the model code and by a second checker that tests the Lipmaa–Moriai conditions bit by bit (`check_lm2.py` in the archive). Fixing all 1 664 difference bits by unit clauses in the model, kissat finds it satisfiable for $k = 66$, so the Matsui conditions with $B_1, \ldots, B_{11}$ do not cut it off, and unsatisfiable for $k = 65$, both within a second.

Table 2: A twelve-round characteristic of SPECK128 of weight 66 (hexadecimal, bit 0 is the least significant bit).

r	x_r	y_r	weight of round r
0	8080000000000a080	84808000000124a0	8
1	0400000000012400	2004000000080104	6
2	2000000000080020	2020000000480801	6
3	0000000000480001	0100000002084008	8
4	000000000e080808	080000001e4a0848	15
5	00000000f2400040	4000000000104200	8
6	0000000000820200	0000000000001202	4
7	0000000000009000	0000000000000010	2
8	0000000000000080	0000000000000000	0
9	8000000000000000	8000000000000000	1
10	8080000000000000	8080000000000004	3
11	8000800000000004	8400800000000020	5
12	8080808000000020	a084808000000124	

Known optima. With the same code and only $B_1, \ldots, B_{R-1}$ in the Matsui conditions, CaDiCaL 1.9.5 through PySAT finds an eight-round characteristic of weight 30 and a ten-round one of weight 49, and shows that eight rounds have none of weight 29; kissat finds a nine-round characteristic of weight 39 and an eleven-round one of weight 58. All four were re-checked by both checkers. The runs that would show none of weight 38 for nine rounds or 48 for ten rounds were stopped after two hours without an answer, so the model check on the unsatisfiable side rests on eight rounds only.

A second encoding. We wrote a second model that shares no code with the first, following Wang et al. [6, Theorems 1–2]: the weight variables are forced in both directions, the bounds on every partial sum are computed beforehand by integer programming, and counter variables exist only inside those bounds. It re-derives the optima of SPECK32 for one to nine rounds (0, 1, 3, 5, 9, 13, 18, 24, 30) and of SPECK128 for one to seven rounds, and with the characteristic of Table 2 fixed it is satisfiable for $k = 66$ and unsatisfiable for $k = 65$. The twelve-round question with $k = 65$ was split in this model into 407 pieces by j and a, and the slowest of those again by the weight of the first nine rounds, each split with its own coverage check; all were solved by CaDiCaL 3.0.1 [12] on the Ryzen machine between 7 and 14 November 2026. Of the 407 pieces, 190 were shown unsatisfiable whole and 221 through their split (four were closed both ways). For each of the 221 the coverage formula, which says that the weight of the first nine rounds lies outside the range allowed by the integer-programming bounds, is unsatisfiable, and all 884 interval pieces are unsatisfiable; two of them were split once more into three intervals of width two, which cover the interval by construction, and closed that way. No run answered satisfiable. Counting every completed run, the second check took about 129 process-days. CaDiCaL and

kissat come from the same group and both models from the same author, so this check is independent in the encoding and the solver program, not in authorship.

6 Discussion

The result concerns characteristics under the Lipmaa–Moriai probabilities and the Markov assumption. Differentials, which sum over characteristics, can be more probable [14], and fixed-key probabilities can differ. The result is conditional on $B_1, \ldots, B_{11}$ as published; B_{10} and B_{11} come from SAT computations of others [6, 7], and we have not re-proved them.

The trusted base of the proof is the model code (checked against the Lipmaa–Moriai formula and the known optima above), the generator of the split (checked by the certified coverage formulas and by regeneration), and, for the $272 - 89$ uncertified pieces, kissat. A probability of 2^{-66} over twelve of 32 to 34 rounds is neither an attack on SPECK128 nor a bound on the security of the full cipher.

Data availability

The archive [15] holds the code; the solver output for 245 of the 272 pieces (for 27 short Ryzen pieces that finished while that machine moved from WSL2 to native Ubuntu only the result line was kept); the table of every solved file with its time, answer and SHA-256 hash; the coverage formulas with their DRAT proofs; the certification log of the short pieces; the model-check logs and characteristics; the characteristic of Table 2 with its fixed-model checks; and the second encoding with its scripts, results and solver outputs.

References

[1] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK families of lightweight block ciphers. Cryptology ePrint Archive, Paper 2013/404, 2013.

[2] ISO/IEC 29167-22:2026. Information technology — Automatic identification and data capture techniques — Part 22: Crypto suite SPECK security services for air interface communications. 2nd edition (1st edition 2018).

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

[4] X. Lai, J. L. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In *EUROCRYPT '91*, LNCS 547, pp. 17–38. Springer, 1991.

[5] L. Sun, W. Wang, and M. Wang. Accelerating the search of differential and linear characteristics with the SAT method. *IACR Transactions on Symmetric Cryptology*, 2021(1):269–315, 2021.

[6] S. Wang, D. Feng, B. Hu, J. Guan, K. Zhang, and T. Shi. New method for combining Matsui's bounding conditions with sequential encoding method. *Designs, Codes and Cryptography*, 91(11):3603–3642, 2023. doi:10.1007/s10623-023-01259-9.

[7] R. Kurahara, K. Sakamoto, and T. Isobe. Finding new differential characteristics on SPECK family. *IEICE Transactions on Fundamentals*, E108-A(5):760–763, 2025. doi:10.1587/transfun.2024EAL2077.

[8] H. Lipmaa and S. Moriai. Efficient algorithms for computing differential properties of addition. In *FSE 2001*, LNCS 2355, pp. 336–350. Springer, 2002.

[9] C. Sinz. Towards an optimal CNF encoding of Boolean cardinality constraints. In *CP 2005*, LNCS 3709, pp. 827–831. Springer, 2005.

[10] A. Biere, K. Fazekas, M. Fleury, and M. Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In *Proc. of SAT Competition 2020*, Department of Computer Science Report Series B, B-2020-1, pp. 51–53. University of Helsinki, 2020.

[11] A. Biere et al. Kissat SAT solver, release 4.0.4. `https://github.com/arminbiere/kissat`.

[12] F. Pollitt, M. Fleury, K. Fazekas, N. Froleyks, A. Schidler, D. Schreiber, and A. Biere. CaDiCaL 3.0. In *SAT 2026*, LIPIcs 377, pp. 40:1–40:14, 2026. doi:10.4230/LIPIcs.SAT.2026.40. Release 3.0.1 used.

[13] N. Wetzler, M. J. H. Heule, and W. A. Hunt Jr. DRAT-trim: efficient checking and trimming using expressive clausal proofs. In *SAT 2014*, LNCS 8561, pp. 422–429. Springer, 2014.

[14] R. Ankele and S. Kölbl. Mind the gap—a closer look at the security of block ciphers against differential cryptanalysis. In *SAC 2018*, LNCS 11349, pp. 163–190. Springer, 2019.

[15] N. Otoi. SPECK128 12-round differential characteristics: code, logs and certificates. Zenodo, 2026. doi:10.5281/zenodo.24136049.