Linear Cryptanalysis of ISO Standard SEED: Exact Trail Optima via Divide-and-Decide Search

Haonan Hong[1] and Kai Zhang[1*]

[1*]Information Engineering University, Zhengzhou, 450001, China.

*Corresponding author(s). E-mail(s): zhkai2010@139.com;

Abstract

SEED is a block cipher developed by the Korea Information Security Agency (KISA) and standardized in ISO/IEC 18033-3. For more than two decades, its best linear cryptanalysis has remained KISA's evaluation: a 5-round distinguisher and no key-recovery attack.

In this paper, a new hierarchical certified search framework, called *divide-and-decide* (D&D), is proposed to determine exact optima of linear trails rather than heuristic upper bounds. D&D divides the trail space at two levels, followed by one decision. At the pattern level, the trail space is partitioned by the positions of trivial rounds, and whole patterns are eliminated before any solver call by the *F-function spacing bound* (FSB) for SEED, an analytic lower bound. At the layer level, each remaining pattern is divided into layers of integer cost, and the lowest layers are enumerated with a satisfiability (SAT) solver to obtain an incumbent trail. At the decision level, the weight beyond the integer cost is treated as an excess, and each remaining layer is closed by a single SAT decision model on this excess instead of being enumerated. For SEED, D&D gives the exact 3- to 7-round optima 18.830, 33.541, 43.472, 55.027, and 65.220, and shows that every 16-round trail has weight at least 112.980. The 6-round hull of weight 53.168 leads to the first linear key-recovery attack on 7-round SEED, and the 7-round hull of weight 63.391 extends linear distinguishers within the codebook from 5 to 7 rounds.

Keywords: Linear cryptanalysis, SEED, divide-and-decide, linear hull, modular addition, key recovery

MSC Classification: 94A60

1 Introduction

SEED is a 128-bit block cipher developed by the Korea Information Security Agency (KISA) and standardized domestically by the Telecommunications Technology Association (TTA). IETF specifications record its use in South Korea for electronic commerce and financial services over both wired and wireless networks [2, 4, 32]. RFC 4269 notes that SEED has been adopted by a wide range of Korean security systems and gives an interoperable public specification of the algorithm [27]. SEED is therefore a deployed cipher rather than a purely academic design.

Its standardization later extended beyond Korea. SEED is one of the 128-bit block ciphers standardized in ISO/IEC 18033-3, together with AES and Camellia [1]. Separate IETF specifications define its use in Cryptographic Message Syntax (CMS), Transport Layer Security (TLS), IPsec ESP, and the Secure Real-time Transport Protocol (SRTP) [2–5], covering secure messaging, transport-layer protection, network-layer confidentiality, and real-time media protection. Because SEED appears in national, international, and Internet standards, its security matters well beyond its country of origin.

A long-lived standard also deserves periodic re-evaluation. Search techniques have advanced considerably since SEED was designed in the late 1990s, and an argument that was convincing with the methods of that time need not reflect the strongest analysis possible today. In differential cryptanalysis, which has received more attention, attacks have reached 9 rounds, although the best 7-round characteristic, obtained in 2014, has not been improved since. In linear cryptanalysis, the results of KISA's security evaluation have remained the best known for the last two decades: the longest published linear distinguisher within the 2^{128} codebook covers 5 rounds, and no linear key-recovery attack has been reported. Whether these results are optimal, or whether stronger linear distinguishers and attacks exist, has remained open.

Linear cryptanalysis is one of the fundamental tools for evaluating the security margin of block ciphers [22, 25], and SEED is a particularly difficult target for it. Each round function interleaves three additions modulo 2^{32} with three G-functions, each consisting of four 8-bit S-boxes followed by a linear diffusion layer. The S-box correlations have noninteger weights, whereas the correlations of modular addition depend on carries and have integer weights. Both kinds of local behavior must in addition satisfy mask-compatibility constraints across the Feistel network. The search problem therefore reduces neither to counting active S-boxes nor to a purely ARX-style model. Manual analysis quickly becomes unmanageable, and an automated search must represent S-box feasibility, modular-addition carries, and multi-round Feistel compatibility simultaneously. This is the main obstacle to a systematic linear analysis of SEED.

1.1 Related Work

KISA's security evaluation [16] already considered the resistance of SEED to linear cryptanalysis. An iterative linear approximation was constructed and evaluated on reduced rounds: the 5-round approximation, with a linear probability of $2^{-100.62}$, is the longest that stays within the codebook, and a 15-round structure was reported

with a linear probability of about $2^{-335.4}$. CRYPTREC later reviewed this evaluation and found no practical linear weakness [6]. Wu, Ma, and Feng subsequently proposed another 15-round linear construction, with a reported linear probability of $2^{-327.84}$ [34]. These remain the principal published linear cryptanalysis results on SEED; both rely on selected iterative structures rather than on a systematic search of the reduced-round mask space. To the best of our knowledge, no later work has optimized unrestricted reduced-round linear trails or linear hulls of SEED, so KISA's construction has served as the reference for the linear security of SEED for the last two decades, although its optimality had not been established even among iterative approximations.

Differential cryptanalysis of SEED has progressed further, but its history also shows how difficult the cipher is to analyze. Yanami and Shimoyama derived 6-round characteristics and attacked 7-round SEED [7]; Sung later constructed a 7-round characteristic and obtained the first attack on 8-round SEED [8]. In 2014, Lu et al. improved the best published 7-round characteristic probability to 2^{-129} and the corresponding differential probability to $2^{-121.07}$, which led to the first attack on 9-round SEED [20]. This characteristic has not been improved in the published literature since.

Automated trail search has meanwhile developed considerably. Matsui's branch-and-bound algorithm initiated the systematic search for differential and linear characteristics [23]. Mouha et al. introduced MILP-based counting of active S-boxes [24], which Sun et al. refined to bit-level S-box models [29], and SAT solvers have since been used to accelerate both the search for trails and the certification of bounds [30]. Most automated models, however, address only one of the two nonlinear components of SEED. For ARX designs, whose only nonlinear operation is modular addition, linear trails are searched with the exact correlation of addition [10, 19, 33]; these models contain no S-boxes. For S-box designs with non-power-of-two weights, Liu et al. proposed a precision-controlled STP model [18]. Taka et al. introduced divide-and-conquer SAT (DC-SAT) [31], which divides the SAT model by the distribution of weight classes and tries the fractional weight combinations one at a time. None of the ciphers treated in [18, 31], which include AES, Camellia, LED, Midori-128, SKINNY-128, and Rijndael-256-256, contains modular addition. SEED requires both at once: noninteger S-box weights must be handled together with exact linear propagation through 32-bit modular additions, and the resulting local approximations must satisfy mask compatibility across the Feistel network.

It has therefore remained open whether KISA's linear estimates for SEED are optimal, and how far the linear cryptanalysis of SEED can reach in distinguishers and in key recovery.

1.2 Contributions

In this paper, the linear cryptanalysis of SEED is carried out systematically, from the structure of its linear trails to exact optima, linear hulls, and a key-recovery attack. Structural theorems are first established that partition the trail space of SEED into a small number of classes with analytic lower bounds, so that whole classes can be excluded before any search. On this basis, a hierarchical certified search framework, divide-and-decide (D&D), is proposed that determines exact trail optima rather than

heuristic upper bounds. The framework is then applied to SEED, which yields the exact optima of 3- to 7-round linear trails, linear distinguishers covering 7 rounds, and the first linear key-recovery attack on SEED. The contributions are detailed as follows.

Table 1 Differential and linear cryptanalysis of SEED; published values are taken from [7, 8, 16, 20].

attack	type	rounds	weight	remark	data	time	memory	reference
DC	distinguisher	6	124	trail	2^{124}	–	–	[7]
		7	122	differential	2^{122}	–	–	[8]
		7	121.07	differential	$2^{121.07}$	–	–	[20]
	key recovery	7	–	–	2^{126} CP	2^{126} F-eval.	–	[7]
		8	–	–	2^{125} CP	2^{122} Enc.	–	[8]
		9	–	–	2^{124} CP	$2^{125.36}$ Enc.	$2^{69.71}$ B	[20]
LC	distinguisher	F	16.771	–	–	–	–	[16]
		F	**9.415**	optimal trail	–	–	–	**Section 4.1**
		5	50.31	–	$2^{100.62}$	–	–	[16]
		5	**43.472**	optimal trail	$\mathbf{2^{86.94}}$	–	–	**Section 4.3**
		6	**55.027**	optimal trail	$\mathbf{2^{110.05}}$	–	–	**Section 4.3**
		6	**53.168**	hull	$\mathbf{2^{106.34}}$	–	–	**Section 5.2**
		7	**65.220**	optimal trail	$\mathbf{2^{130.44}}$	–	–	**Section 4.4**
		7	**63.391**	hull	$\mathbf{2^{126.78}}$	–	–	**Section 5.2**
	key recovery	7	–	–	$\mathbf{2^{115}}$ KP	$\mathbf{2^{115}}$ Enc.	$\mathbf{2^{68}}$ B	**Section 5.4**

Results of this paper are set in bold, and their reference entry gives the section in which they are obtained. All values in the weight column are weights: $w = -\log_2 p$ for a DC characteristic or differential of probability p, and $w = -\log_2 |c|$ for an LC trail or hull of correlation c. The remark states what each weight refers to: a single DC trail (characteristic), a differential summed over trails, an optimal LC trail, that is, the exact optimum over all LC trails, or a linear hull evaluated within a weight window (Section 5.2); a dash marks the iterative approximations of KISA. KISA's 6- and 7-round iterative approximations, both of weight 67.08 (linear probability $2^{-134.16}$), lie beyond the full codebook and are therefore not listed. The KISA weights are converted from linear probabilities by $w = -\frac{1}{2}\log_2 LP$, and row F compares the iterative one-round approximation of [16] with the unrestricted F-function optimum. The distinguisher data complexity is 2^w for DC and 2^{2w} for LC. CP and KP denote chosen and known plaintexts, F-eval. round-function evaluations, Enc. encryptions, and B bytes.

Structural theorems are established that partition SEED trails into a few classes with analytic bounds. The zero masks of any linear trail of SEED are proved to be spaced at least three positions apart (Theorem 1), so that the trail space splits into zero patterns whose number is given in closed form (Theorem 2). From this structure and the exact optima of a single round function and of a reciprocal pair, the F-function spacing bound (FSB) for SEED is derived (Theorem 7); it charges every pattern analytically and has a closed form (Corollary 2). Once these two local optima are certified, FSB requires no further search: it bounds the weight of every 16-round trail by 112.980 in about 3.4 ms and determines which patterns the subsequent search must examine.

A hierarchical divide-and-decide framework is proposed that certifies exact optima. The search is organized in three levels: it divides the trail space at two levels, followed by one decision. At the pattern level, the trail space is partitioned by zero pattern, and FSB discards whole patterns analytically. At the layer level, each remaining pattern is divided into layers of integer cost by charging every active S-box its minimum weight, with the three modular additions of the round function encoded exactly by a carry automaton, and the lowest layers are enumerated with SAT. At the

decision level, the excess, the amount by which the exact weight of a trail exceeds its cost, is used: a layer of cost B can contain a trail of weight below the incumbent w^* only if some trail in it has excess below $w^* - B$, so every remaining layer is settled by one SAT decision model on its total excess (Proposition 2) instead of being enumerated. S-boxes and modular additions are thereby handled exactly within one certified search.

Exact 3- to 7-round optima of SEED are determined, and linear distinguishers within the codebook are extended from 5 to 7 rounds. The minimum weight of an isolated F-function trail is proved to be 9.415 and that of a reciprocal pair 33.541, and the exact optima of 3- to 7-round trails are determined as 18.830, 33.541, 43.472, 55.027, and 65.220. KISA's 4-round evaluation is shown to be optimal, and its iterative one-round approximation and its 15-round construction are shown to be optimal within their structural restrictions, whereas KISA's 5-, 6-, and 7-round estimates are improved by about 6.84, 12.05, and 1.86 bits. The exact 6-round optimum already yields a 6-round distinguisher within the 2^{128} codebook. With linear hulls, evaluated over all trails within a weight window between fixed endpoints, the 6- and 7-round weights are lowered further to 53.168 and 63.391, which brings the 7-round distinguisher within the codebook as well, so that linear distinguishers within the codebook are extended from 5 to 7 rounds.

The first linear key-recovery attack on SEED is proposed. To the best of our knowledge, no linear key-recovery attack on SEED has been published before. By appending one round to the 6-round linear-hull distinguisher, the full 128-bit key of 7-round SEED is recovered with data complexity 2^{115} known plaintexts, time complexity 2^{115} encryptions, and memory complexity 2^{68} bytes, with an estimated success probability of about 81% on average over keys; even when the hull estimate is replaced by the best 6-round trail as a lower bound on the expected squared correlation, about 2^{119} known plaintexts suffice. The same extension of the 7-round hull leaves only one bit of advantage even with the full codebook, so this route ends at seven rounds.

Table 1 places these results beside the published differential and linear cryptanalysis of SEED, separating distinguishers from key-recovery attacks. Summary values are reported to three decimal places; exact expressions or additional digits are retained in proof sections when needed.

2 Preliminaries

This section recalls the standard notions of linear cryptanalysis used throughout the paper, namely correlation and weight, linear trails and hulls [14, 22, 25], and describes the SEED round function [27, 32].

2.1 Basics of Linear Cryptanalysis

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Following [22], a linear approximation is specified by an input mask $\alpha \in \mathbb{F}_2^n$ and an output mask $\beta \in \mathbb{F}_2^m$, and compares the input parity $\alpha \cdot x$ with the output parity $\beta \cdot f(x)$. Its correlation [14] is

$$c_f(\alpha, \beta) = 2 \Pr_x[\alpha \cdot x = \beta \cdot f(x)] - 1.$$

If $\epsilon = \Pr_x[\alpha \cdot x = \beta \cdot f(x)] - \frac{1}{2}$ denotes the usual bias [22], then $c = 2\epsilon$. Thus, $c = 0$ means that the relation holds for exactly half of the inputs, whereas a larger $|c|$ indicates a stronger linear approximation.

Two equivalent measures are used in the SEED literature. The *linear probability* is $LP = c^2$, while the *correlation weight* used throughout this paper is

$$w = -\log_2 |c| = -\frac{1}{2}\log_2 LP. \tag{1}$$

Unless explicitly stated otherwise, "weight" always refers to correlation weight. The three quantities are therefore related by

$$|c| = 2^{-w}, \qquad LP = 2^{-2w}.$$

A smaller weight represents a stronger approximation. For a basic linear distinguisher, the characteristic data scale is proportional to $c^{-2} = 2^{2w}$. Increasing the weight by one bit therefore halves the correlation and quadruples the data requirement. This convention also connects the historical SEED notation to that used below. For example, the KISA value $LP \approx 2^{-335.4}$ corresponds to correlation weight $w \approx 167.7$.

For an n-bit S-box S, the centered linear-approximation table (LAT) [22] is defined by

$$\mathrm{LAT}[a][b] = \#\{x : a \cdot x = b \cdot S(x)\} - 2^{n-1}.$$

For the 8-bit S-boxes of SEED, $c = \mathrm{LAT}[a][b]/128$. For instance, $|\mathrm{LAT}| = 16$ gives $|c| = 2^{-3}$ and $w = 3$, while $|\mathrm{LAT}| = 12$ gives $|c| = 3/32$ and $w = \log_2(32/3) \approx 3.415$. The non-integer weights that appear later are therefore inherent to the S-box correlations rather than approximations introduced by the search model.

Linear trails and hulls.

A linear trail specifies the masks at every intermediate component of the cipher. Under the standard independent-round-key model, the absolute correlation of a trail is the product of its local absolute correlations [22]. Equation (1) converts this product into a sum: if a trail T contains local transitions of weights $w_1, \ldots, w_m$, then

$$|c(T)| = \prod_{i=1}^{m} 2^{-w_i}, \qquad w(T) = \sum_{i=1}^{m} w_i.$$

This additive representation is the principal reason for formulating automatic linear-trail search as a weight-minimization problem.

A trail should be distinguished from a linear hull. A trail fixes one complete sequence of internal masks, whereas a hull fixes only the external input and output masks and contains all trails connecting them. The signed correlations of those trails interfere for a fixed key. Under Nyberg's independent-round-key model [25], however, the expected squared correlation satisfies

$$\mathbb{E}_K[c_{\mathrm{hull}}^2] = \sum_{T} 2^{-2w(T)},$$

and it is convenient to define the corresponding hull weight as

$$w_{\text{hull}} = -\frac{1}{2}\log_2\left(\sum_T 2^{-2w(T)}\right). \tag{2}$$

Since several trails may contribute to the same external masks, w_{hull} can be smaller than the weight of the best individual trail. Following common practice in automated trail-based hull evaluation [15, 19], the hull values reported later evaluate Eq. (2) over the trails found within a weight window between fixed endpoints.

In SEED the round key is XORed at the input of F before any key-independent operation, so $F_K(x) = F_0(x \oplus K)$ and the correlation of a fixed transition is $(-1)^{\alpha\cdot K} c_{F_0}(\alpha, \beta)$. The sign of a trail is therefore an affine function of the round keys, and the cross terms vanish when the round keys are independent and uniform. The round keys of SEED are, however, derived from the master key by a nonlinear key schedule, so independent round keys remain an explicit modeling assumption. All hull values reported below are therefore model-dependent expectations for a randomly drawn key, and no claim is made about the correlation of SEED under any particular key.

2.2 Description of SEED

A *Feistel cipher* splits its state into two halves and applies rounds of the form $(L, R) \mapsto (R,\ L \oplus F(R, K))$, where K is the round key. Throughout this paper, the *round function* is this inner function F, also called the F-function, and not the whole round transformation; it is a permutation if $F(\cdot, K)$ is a bijection on the half-state for every round key K.

SEED is a Feistel cipher with 64-bit halves, 16 rounds, and a 128-bit master key [27, 32]. Let (L_i, R_i) denote the two 64-bit halves entering round i. Apart from the final output convention, the round transformation is

$$(L_i, R_i) \longmapsto \big(R_i,\ L_i \oplus F(R_i; K_{i,0}, K_{i,1})\big),$$

where the 64-bit round subkey is divided into two 32-bit words $(K_{i,0}, K_{i,1})$.

The 64-bit input of F consists of two 32-bit words (C, D). The two round-key words are XORed into C and D, the first word is then XORed into the second, and three applications of G are interleaved with three additions modulo 2^{32}:

$$\begin{aligned}
T_0 &= C \oplus K_{i,0}, & T_1 &= D \oplus K_{i,1},\\
T_1 &\leftarrow T_1 \oplus T_0, & T_1 &\leftarrow G(T_1),\\
T_0 &\leftarrow T_0 \boxplus T_1, & T_0 &\leftarrow G(T_0),\\
T_1 &\leftarrow T_1 \boxplus T_0, & T_1 &\leftarrow G(T_1),\\
T_0 &\leftarrow T_0 \boxplus T_1, & (C', D') &= (T_0, T_1),
\end{aligned}$$

where $\boxplus$ denotes addition modulo 2^{32}. Thus a single F-function contains three G-functions and three modular additions; Figure 1 shows the round function F and the function G.

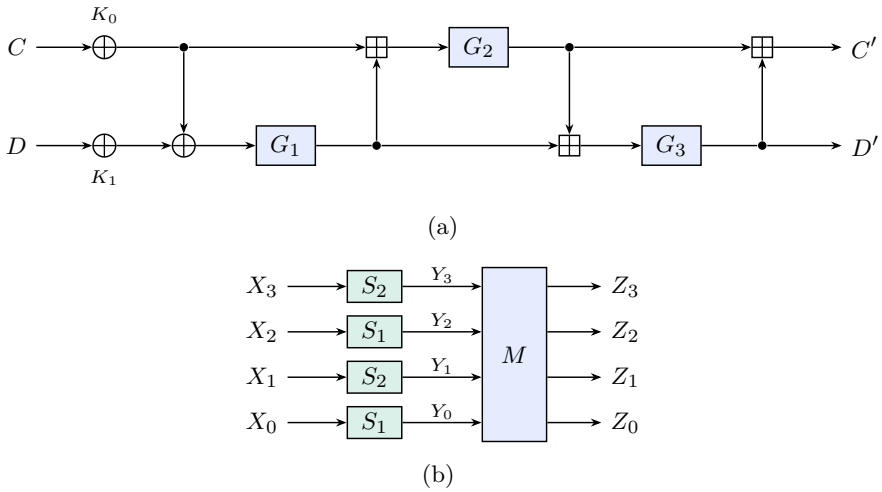


Fig. 1 The SEED round function. (a) The round function F: the 64-bit input (C, D) is combined with the round-key words K_0 and K_1; the upper and lower lanes carry T_0 and T_1, G_1, G_2, G_3 are the three applications of G, and each $\boxplus$ is an addition modulo 2^{32}. (b) The function G, which applies S_1 to the bytes X_0, X_2 and S_2 to X_1, X_3, and whose linear layer M outputs $Z_i = \bigoplus_{j=0}^{3} \left(Y_j \wedge m_{(i+j) \bmod 4}\right)$ with $m_0 = \texttt{fc}$, $m_1 = \texttt{f3}$, $m_2 = \texttt{cf}$, and $m_3 = \texttt{3f}$ in hexadecimal; drawn after the specification [27]

The function G, shown in Figure 1(b), applies S_1 to the bytes X_0 and X_2 and S_2 to the bytes X_1 and X_3 of a 32-bit word $X = X_3 \| X_2 \| X_1 \| X_0$, and then a fixed linear map M on 32 bits, which mixes the four S-box outputs. Since F contains three G-functions, it contains twelve S-box applications.

The components of F have qualitatively different linear behavior, as summarized in Table 2. XOR operations, key XORs, and the linear map M impose deterministic mask relations and do not reduce the correlation magnitude by themselves. S-box transitions are discrete but have noninteger weights such as 3.415 and 3.193. Modular addition is different again: its linear correlation depends on carry-mask compatibility, but every nonzero correlation has a power-of-two magnitude and hence an integer weight [33]. A valid SEED trail must satisfy all three kinds of constraint simultaneously.

Table 2 SEED components relevant to the linear-trail search.

component	occurrences in one F	role in the linear search
8-bit S-box	12	noninteger local weights
modular addition	3	carry-dependent integer weights
linear map M in G	3	masks propagate through $M^{\top}$
XOR of C into D	1	masks propagate deterministically
key XOR	2	masks are unchanged

This combination explains why SEED differs from the two common extremes in automated linear search. It is not an SPN for which an active-S-box bound can capture most of the objective, and it is not a pure ARX design whose local weights are entirely determined by bit-level addition constraints. The search must retain the exact feasibility of both component types while still obtaining a sufficiently strong lower bound to prune the multi-round Feistel space. The next section proposes the search strategy used for this purpose.

3 The Divide-and-Decide Framework

This section proposes divide-and-decide (D&D), a hierarchical search framework used to derive and certify the SEED linear trails of this paper. D&D divides the trail space at two levels, followed by one decision: it divides the trail space by zero pattern, divides each pattern into layers of integer cost, and decides each remaining layer with one SAT decision model. The first division rests on the Feistel structure of SEED: the zero masks of a trail determine its pattern, and a whole pattern can be discarded once its analytic lower bound reaches the weight of the best trail found. The second division and the decision rest on a simple observation. If every active S-box is charged its minimum weight 3, every trail falls into a layer of integer cost B, and its exact weight equals B plus a nonnegative excess. A layer of cost B can contain a trail lighter than the best one found, w^*, only if some trail in it has excess below the *allowed excess* $w^* - B$, so each layer can be settled by a single yes/no decision instead of being enumerated.

Figure 2 shows why this is effective on SEED. From one cost layer to the next, the number of trails grows about tenfold, while the allowed excess shrinks by one bit. Enumeration is inexpensive in the low layers, which are small, and yields the incumbent; decisions are fast in the high layers, which are large but allow little excess.

The three levels of D&D are as follows: the pattern and layer levels perform the two divisions, and the decision level performs the decision. An incumbent trail is kept throughout, and every zero pattern whose lower bound reaches the incumbent weight is discarded without calling the solver.

- *Pattern level.* A trail is described by its sequence of masks $\lambda_{-1}, \lambda_0, \ldots, \lambda_r$ between the rounds, and the positions of the zero masks form its *zero pattern* Z. The trail space is partitioned by zero pattern (Section 3.3), and each pattern receives an analytic lower bound $\mathrm{LB}(Z)$ on the weight of its trails. For SEED, LB is the F-function spacing bound of Section 4.2, which is assembled from the exact optima of an isolated F-function and of a reciprocal pair. These two optima are determined first by the same procedure, which needs no pattern bound for them because each consists of a single structure; the bound is then available for every number of rounds.
- *Layer level.* Within a pattern, trails are grouped into layers of an integer *surrogate cost* $\mathrm{cost}(T)$, which never exceeds the exact weight $w(T)$ (Section 3.1). SAT certifies which layers are empty and enumerates the lowest nonempty ones, which yields the incumbent weight w^*.
- *Decision level.* The difference $\mathrm{exc}(T) = w(T) - \mathrm{cost}(T)$, the *excess* of T, is a sum of per-S-box terms. Every remaining layer $B < w^*$ is decided by one SAT decision

model that asks for a trail of cost B whose excess is below the allowed excess $w^* - B$ (Proposition 2).

Sections 3.1 to 3.4 define these ingredients, and Section 3.6 assembles them into the complete procedure.

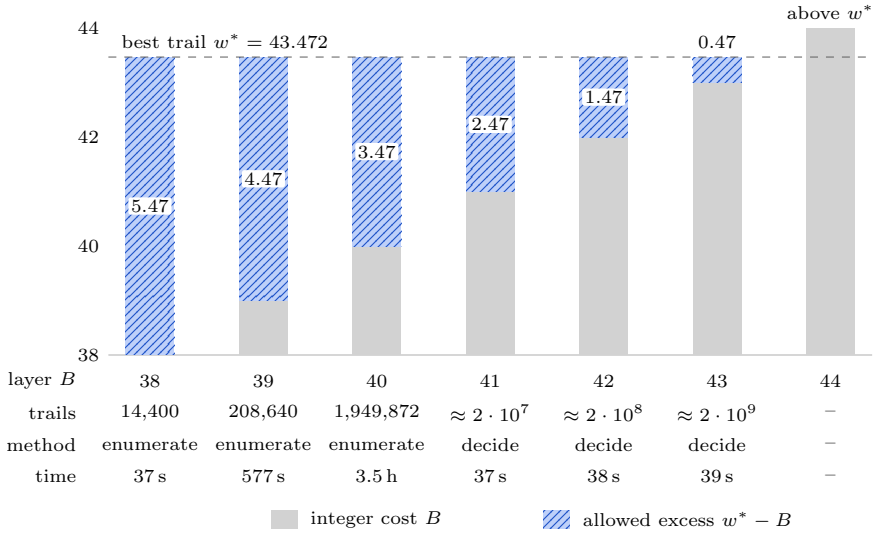


Fig. 2 The second division and the decision of D&D within the 5-round pattern $Z[0, 4]$ of SEED. Each bar is a cost layer B; its lower, plain part is the integer cost and its upper, hatched part the allowed excess $w^* - B$, the largest excess with which a trail of this layer could still beat the best trail, $w^* = 43.472$. From one layer to the next, the number of trails grows about tenfold (the last three counts are estimates), while the allowed excess shrinks by one bit. The small low layers are therefore enumerated, and the large high layers are decided by one SAT decision model each; the times are solver times

3.1 Surrogate Cost, Excess, and Stopping Rule

For a trail T, let $N_S(T)$ be the number of active 8-bit S-boxes and let $w_{\boxplus,j}(T)$ denote the exact correlation weight of the jth modular addition. The exact trail weight is

$$w(T) = \sum_{i \in \mathcal{S}(T)} w_i(T) + \sum_j w_{\boxplus,j}(T),$$

where $\mathcal{S}(T)$ is the set of active S-box transitions. Both SEED S-boxes have the same distribution of absolute LAT values, listed in Table 3. The minimum nonzero S-box weight is 3, attained at $|\mathrm{LAT}| = 16$; only 1275 of the 60945 valid nonzero mask transitions attain it, about 2.1%, so most active S-boxes cost more than three bits.

Table 3 LAT distribution common to both SEED S-boxes, over nonzero input and output masks, with the weight w and the excess $e = w - 3$ of each class. Values are rounded to three decimals.

$\lvert\text{LAT}\rvert$	0	2	4	6	8	10	12	14	16
w	∞	6	5	4.415	4	3.678	3.415	3.193	3
$e = w - 3$	–	3	2	1.415	1	0.678	0.415	0.193	0
count	4080	12240	9180	10200	8670	6120	9180	4080	1275

Since every active SEED S-box has weight at least 3, the following integer quantity is a lower bound on the exact weight:

$$\text{cost}(T) = 3N_S(T) + \sum_j w_{\boxplus,j}(T). \tag{3}$$

Hence

$$\text{cost}(T) \le w(T) \tag{4}$$

for every valid trail. Only the S-box contribution is relaxed; the modular-addition contribution is kept exact.

The inequality in Eq. (4) is the basis of both lower-bound proofs and the stopping condition.

Proposition 1 (Surrogate stopping rule) *Let B be a non-negative integer. If the SAT model with $\text{cost}(T) \le B$ is UNSAT, then every valid trail satisfies $w(T) > B$. Furthermore, suppose that every surrogate level $0, 1, \ldots, B$ has been completely exhausted and that the lightest exact trail found has weight $w^* < B + 1$. Then w^* is globally optimal.*

Proof If a trail with $w(T) \le B$ existed, then $\text{cost}(T) \le w(T) \le B$, contradicting UNSAT. For the second statement, assume that a better trail T' exists with $w(T') < w^* < B + 1$. Since $\text{cost}(T')$ is an integer and $\text{cost}(T') \le w(T') < B+1$, it must satisfy $\text{cost}(T') \le B$. Such a trail would already have appeared in one of the exhausted levels, which is impossible. □

The quantity discarded by Eq. (3) is what the decision level uses. For a valid trail T, define its *excess*

$$\text{exc}(T) = w(T) - \text{cost}(T) = \sum_{i \in \mathcal{S}(T)} e_i(T) \ \ge 0, \qquad e_i(T) = w_i(T) - 3. \tag{5}$$

The modular-addition terms cancel, so the entire gap between the surrogate cost and the exact weight is carried by the active S-boxes, and each term $e_i(T)$ is one of the eight per-class excesses listed in Table 3. Hence the exact weight splits into an integer cost and a nonnegative excess,

$$w(T) = \text{cost}(T) + \text{exc}(T), \tag{6}$$

so the cost of a trail equals its weight exactly when its excess is zero, that is, when all its active S-boxes have the minimum weight 3; the minimum cost equals the optimum

exactly when some trail of minimum cost has zero excess. The surrogate lower bounds never exceed the published optima, the trails found never fall below them, the surrogate is tight exactly when a trail of minimum cost has zero excess, and the excess decision introduced below reaches all three published optima.

Operationally, a completed UNSAT region gives a lower bound on the true weight. Conversely, once the next unexplored integer cost layer is already larger than the best exact trail found so far, no unseen trail can improve that incumbent. The search can then stop with a proof of optimality.

Enumerating a whole layer becomes expensive when it contains millions of assignments, almost all of which carry a large excess. Equation (6) avoids this enumeration. A trail of cost B can be lighter than an incumbent $\bar{w}$ only if its excess is below the allowed excess $\bar{w} - B$, so it suffices to decide whether such a trail exists. To keep the decision within integer arithmetic, each per-S-box excess is expressed in units of $1/\kappa$ bit and rounded down. For a zero pattern P, a cost B, and a threshold $t > B$, let $\mathcal{M}_P(B, t)$ denote the SAT model of pattern P augmented with the constraints

$$\mathrm{cost}(T) \le B, \qquad \sum_{i \in \mathcal{S}(T)} \left\lfloor \kappa\, e_i(T) \right\rfloor \;\le\; \left\lceil \kappa\,(t - B) \right\rceil - 1. \tag{7}$$

The class of each active S-box is identified from its input and output masks, every class whose excess alone exceeds the allowed excess is excluded, and the weighted sum is imposed with a totalizer, an integer weight being represented by repeated literals. This paper uses $\kappa = 40$, so each S-box loses less than 0.025 bit to the rounding. The first constraint is imposed as $\mathrm{cost}(T) \le B$ rather than $\mathrm{cost}(T) = B$, because the cardinality encoding bounds sums from above; this only enlarges the model, so an UNSAT answer still settles the layer of cost B, and a satisfying assignment of smaller cost is evaluated exactly like any other.

Proposition 2 (Excess decision) *Let P be a zero pattern, B an integer, and $t > B$.*

1. *If $\mathcal{M}_P(B, t)$ is UNSAT, then every trail T of pattern P with $\mathrm{cost}(T) = B$ satisfies $w(T) \ge t$.*
2. *Suppose that the layers $\mathrm{cost} \le B_0$ of P have been exhausted with incumbent $\bar{w}$, and that $\mathcal{M}_P(B, \bar{w})$ is UNSAT for every integer B with $B_0 < B < \bar{w}$. Then no trail of pattern P is lighter than $\bar{w}$.*

Proof For the first statement, let T be a trail of pattern P with $\mathrm{cost}(T) = B$ and $w(T) < t$. By Eq. (6), $\mathrm{exc}(T) = w(T) - B < t - B$. Since $\lfloor \kappa e_i \rfloor \le \kappa e_i$ for every term,

$$\sum_{i \in \mathcal{S}(T)} \left\lfloor \kappa\, e_i(T) \right\rfloor \le \kappa\, \mathrm{exc}(T) < \kappa\,(t - B),$$

and an integer strictly below $\kappa(t - B)$ is at most $\lceil \kappa(t - B) \rceil - 1$. Hence T satisfies $\mathcal{M}_P(B, t)$, which contradicts UNSAT. For the second statement, a trail T' of pattern P with $w(T') < \bar{w}$ has $\mathrm{cost}(T') \le w(T') < \bar{w}$ by Eq. (4). If $\mathrm{cost}(T') \le B_0$, it lies in an exhausted layer and cannot be lighter than $\bar{w}$; otherwise $B_0 < \mathrm{cost}(T') < \bar{w}$, and the first statement with $B = \mathrm{cost}(T')$ and $t = \bar{w}$ gives $w(T') \ge \bar{w}$. □

Because the rounding only relaxes the constraint, a satisfying assignment of $\mathcal{M}_P(B,t)$ may still have exact weight at least t. Such an assignment is a *false positive*: its exact weight is evaluated outside the solver, it is blocked, and the decision is repeated. A satisfying assignment with exact weight below t is a lighter trail, which replaces the incumbent and is blocked as well.

In the runs of Section 4 the threshold t is set slightly above the incumbent $\bar{w}$, so the decision may also return trails of weight exactly $\bar{w}$; such a trail is evaluated exactly and blocked like a false positive, so one decision model may be solved several times before it becomes UNSAT. Blocking only assignments whose exact weight is at least the incumbent cannot remove a lighter trail, so if the decision becomes UNSAT after these blockings, no trail of cost B is lighter than $\bar{w}$; since the incumbent never increases, this remains true for every later incumbent.

The separation of integer and fractional weights also underlies DC-SAT [31], which fixes the integer part and then tries the combinations of fractional classes one by one in ascending order. Proposition 2 differs in this second step: all combinations of a layer are settled at once by a single decision model about their total, and only the layers above an incumbent are decided.

Inside the solver, the counter of Eq. (7) raises a conflict as soon as the excess of the S-boxes assigned so far exceeds the allowed excess, so every completion of such an incomplete assignment is discarded, in the spirit of the bounding step of branch-and-bound. Unlike a branch-and-bound search, however, the order of assignments and the reasons for pruning are left to the solver, which learns them as clauses that are reused across the whole layer.

The bound in Eq. (3) is imposed incrementally with a totalizer encoding [11, 21]. The integer cost is raised one level at a time. Whenever SAT returns a feasible assignment, its exact S-box weights and exact modular-addition weights are evaluated and the lightest trail found so far is updated.

3.2 Exact Modeling of Modular Addition

The S-box constraints encode feasibility: an input/output mask pair is allowed exactly when the corresponding LAT entry is nonzero. Their exact weights are evaluated after a SAT solution is obtained. Modular addition is treated differently because its exact weight is already part of the surrogate.

Consider $z = x \boxplus y$ with input masks u, v and output mask s. Wallén's carry-mask representation [33] introduces bits $e_0, \ldots, e_n$ with $e_n = 0$ and $t_i = u_i + v_i + s_i + e_i$. The case $t_i = 2$ is infeasible, while

$$e_{i+1} = 0 \quad \text{for } t_i \in \{0,4\}, \qquad e_{i+1} = 1 \quad \text{for } t_i \in \{1,3\}.$$

For every feasible mask triple, the carry-mask sequence is unique and

$$|c(u,v,s)| = 2^{-\mathrm{HW}(e_1,\ldots,e_{n-1})}.$$

Therefore,

$$w_{\boxplus} = \mathrm{HW}(e_1, \ldots, e_{n-1}), \tag{8}$$

which is an exact integer quantity. The SAT model can thus represent both the feasibility and the precise linear weight of each modular addition.

This representation was cross-checked on 5-bit addition by enumerating all 2^{15} input/output mask triples and comparing the predicted correlations with brute-force evaluation.

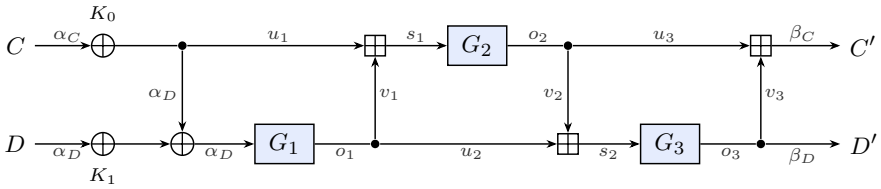


Fig. 3 Linear masks inside the SEED round function F, as used in Sections 3.2 and 4.1; the relations between them are given in the text

For the SEED round function, the same mask equations connect the three additions and the three G-functions (Figure 3). Let the input and output masks of F be $\alpha = (\alpha_C, \alpha_D)$ and $\beta = (\beta_C, \beta_D)$. Writing the three addition transitions as

$$(u_1, v_1) \to s_1, \qquad (u_2, v_2) \to s_2, \qquad (u_3, v_3) \to \beta_C,$$

and denoting the three G-output masks by o_1, o_2, o_3, mask compatibility gives

$$\alpha_C = u_1 \oplus \alpha_D, \qquad o_1 = v_1 \oplus u_2, \qquad o_2 = u_3 \oplus v_2, \qquad o_3 = v_3 \oplus \beta_D.$$

The three G-input masks are α_D, s_1, s_2. Consequently, the exact weight of one F-function is

$$w_F = w_{G_1} + w_{G_2} + w_{G_3} + w_{\boxplus,1} + w_{\boxplus,2} + w_{\boxplus,3},$$

whereas the SAT objective uses

$$\mathrm{cost}_F = 3N_S + w_{\boxplus,1} + w_{\boxplus,2} + w_{\boxplus,3}.$$

Thus the only difference between the surrogate and the exact objective comes from replacing each active S-box weight by its minimum value 3.

3.3 Zero-Pattern Decomposition

The Feistel structure of SEED partitions the trail space before any SAT search begins. Let $(\Gamma L_j, \Gamma R_j)$ denote the two branch masks after j rounds, with $(\Gamma L_0, \Gamma R_0)$ denoting the masks at the cipher input. Thus round j maps $(\Gamma L_{j-1}, \Gamma R_{j-1})$ to $(\Gamma L_j, \Gamma R_j)$. For compact notation, define

$$\lambda_{-1} = \Gamma R_0, \qquad \lambda_j = \Gamma L_j \quad (j \geq 0).$$

Mask matching in round j then gives

$$\Gamma R_j = \lambda_{j-1}, \qquad \alpha_j = \lambda_{j-2} \oplus \lambda_j, \qquad \beta_j = \lambda_{j-1}. \tag{9}$$

The jth F-function is trivial exactly when $\lambda_{j-1} = 0$.

Since every right-half mask repeats the previous left-half mask, an r-round trail is determined by the $r + 2$ masks $\lambda_{-1}, \lambda_0, \ldots, \lambda_r$, and round j involves only λ_{j-2}, λ_{j-1}, and λ_j. The set of positions at which these masks vanish is the *zero pattern* of the trail, and $Z[i_1, \ldots, i_t]$ denotes the pattern whose zeros are exactly $\lambda_{i_1}, \ldots, \lambda_{i_t}$. A zero pattern is *admissible* if any two of its zeros are at least three positions apart; Theorem 1 below shows that the zero pattern of every nonzero trail is admissible. In the round notation of KISA's evaluation, a trivial round is written 0 and a nontrivial one A; at 7 rounds, for example, $Z[0, 3, 6]$ is $0AA0AA0$ and $Z[1, 5]$ is $A0AAA0A$. Zeros at the boundary positions λ_{-1} and λ_r make no round trivial and are therefore not visible in the round notation.

For a valid nonzero-correlation approximation of a round function that is a permutation,

$$\alpha_j = 0 \quad \Longleftrightarrow \quad \beta_j = 0. \tag{10}$$

Indeed, a nonzero input mask cannot have nonzero correlation with the zero output mask, and the zero input mask cannot have nonzero correlation with a nonzero output mask. The SEED F-function is a permutation for every fixed round key: the linear layer M of G has full rank over $\mathbb{F}_2$, so G is a bijection, and every other step of F is an invertible XOR or modular addition. Hence Eq. (10) holds in every round.

Knudsen [17] observed, in a discussion of practical security against differential and linear attacks, that a Feistel cipher whose round function is a permutation needs at least two non-trivial one-round characteristics in every three rounds, and the iterative $0AA$ patterns of KISA's evaluation [16] realize this spacing. The observation is stated there as an informal remark rather than as a formal result. The following theorem formalizes it for linear trails of a Feistel cipher, extends it to the two boundary positions λ_{-1} and λ_r, and shows that the resulting bound is sharp at the level of zero patterns. In this form it yields the exact partition of the trail space used by the search below.

Theorem 1 (Spacing of trivial rounds) *Let E be an r-round Feistel cipher whose round function is a permutation for every round key. In any nonzero linear trail of E, two zero positions among $\lambda_{-1}, \lambda_0, \ldots, \lambda_r$ are separated by at least three indices. Consequently, the trail contains at least $k(r) = r - \lceil r/3 \rceil$ nontrivial round functions, and this count is sharp at the level of zero patterns: for every r some admissible zero pattern has exactly $k(r)$ nontrivial rounds, whereas whether a trail realizes such a pattern depends on the round function.*

Proof Let $\lambda_p = 0$. Since $\beta_{p+1} = \lambda_p = 0$, Eq. (10) gives $\alpha_{p+1} = 0$. By Eq. (9),

$$\lambda_{p-1} = \lambda_{p+1}. \tag{11}$$

Now suppose that another zero occurs one position away, i.e., $\lambda_{p+1} = 0$. Then Eq. (11) gives $\lambda_{p-1} = 0$. Applying the same relation successively to the neighboring zero positions propagates zeros in both directions and collapses the trail to the all-zero trail. At the two

ends of the index range the same recursion applies with the convention that a nonexistent round imposes no constraint, so the propagation terminates one step earlier without affecting the conclusion.

Next suppose that another zero occurs two positions away, i.e., $\lambda_{p+2} = 0$. For round $p+2$,

$$\alpha_{p+2} = \lambda_p \oplus \lambda_{p+2} = 0.$$

Equation (10) then implies $\beta_{p+2} = \lambda_{p+1} = 0$, reducing this case to the adjacent-zero case above. Therefore two zero positions cannot be at distance one or two in a nonzero trail, and their distance is at least three.

A zero λ_{j-1} corresponds to a trivial round j. Among r consecutive round positions, at most $\lceil r/3 \rceil$ such zeros can occur, so at least $r - \lceil r/3 \rceil$ round functions are nontrivial.

For sharpness, place zeros at $\lambda_0, \lambda_3, \lambda_6, \ldots$ and keep every other position nonzero. Any two zeros are then exactly three indices apart, and exactly $\lceil r/3 \rceil$ of the positions $\lambda_0, \ldots, \lambda_{r-1}$ are zero, so exactly $r - \lceil r/3 \rceil$ round functions are nontrivial. □

Since every nonzero trail has exactly one admissible zero pattern, the admissible patterns partition the trail space, and a search may treat them one at a time. The next theorem counts them.

Theorem 2 (Number of admissible zero patterns) *For $r \geq 1$, the number of admissible zero patterns of an r-round trail is*

$$N(r) = \sum_{k \geq 0} \binom{r + 4 - 2k}{k}. \tag{12}$$

Proof Mark each of the $r+2$ positions $\lambda_{-1}, \ldots, \lambda_r$ as zero or nonzero, and consider a pattern with $k \geq 1$ zeros. Let $g_0, \ldots, g_k$ be the numbers of consecutive nonzero positions before the first zero, between consecutive zeros, and after the last zero, so that $g_0 + \cdots + g_k = r + 2 - k$. By Theorem 1, $g_1, \ldots, g_{k-1} \geq 2$, while g_0 and g_k are unrestricted. Subtracting 2 from each of the $k-1$ inner gaps leaves $k+1$ non-negative integers with sum $r + 4 - 3k$, and there are $\binom{(r+4-3k)+k}{k} = \binom{r+4-2k}{k}$ such tuples; the pattern without zeros gives the term $k = 0$. □

To traverse the admissible patterns, the search generates them recursively rather than from Eq. (12). The following corollary gives the recursion and its count.

Corollary 1 (Recursive generation of patterns) *For $r \geq 4$, every admissible r-round zero pattern is obtained either from an admissible $(r-1)$-round pattern by appending a nonzero λ_r, or from an admissible $(r-3)$-round pattern by appending $\lambda_{r-2} \neq 0$, $\lambda_{r-1} \neq 0$, and $\lambda_r = 0$, and the two cases are disjoint. Consequently, $N(r) = N(r-1) + N(r-3)$ for $r \geq 4$, with $N(1) = 4$, $N(2) = 6$, and $N(3) = 9$.*

Proof Consider an admissible r-round pattern. If $\lambda_r \neq 0$, deleting λ_r leaves an admissible pattern on $\lambda_{-1}, \ldots, \lambda_{r-1}$, that is, an $(r-1)$-round pattern, and every such pattern extends in this way. If $\lambda_r = 0$, Theorem 1 forces $\lambda_{r-1} \neq 0$ and $\lambda_{r-2} \neq 0$, so deleting these three positions leaves an admissible pattern on $\lambda_{-1}, \ldots, \lambda_{r-3}$, that is, an $(r-3)$-round pattern; conversely, appending two nonzero positions and a zero to any admissible $(r-3)$-round pattern keeps

every pair of zeros at least three positions apart. The two cases differ in λ_r, which gives the recurrence. The initial values follow from Eq. (12): for $r = 1, 2, 3$ the sum equals $1 + 3 = 4$, $1 + 4 + 1 = 6$, and $1 + 5 + 3 = 9$. □

Hence $N(r)$ grows as ρ^r, where $\rho \approx 1.466$ is the real root of $x^3 = x^2 + 1$. Table 4 lists the values used in this paper; for instance, the 41 patterns at 7 rounds split into $1 + 9 + 21 + 10$ patterns with 0, 1, 2, and 3 zeros. The recursion of Corollary 1 also translates directly into a short generator of all admissible patterns, which makes the analytic lower bound of Section 4.2 simple to implement and fast to evaluate.

Table 4 Number of admissible zero patterns of an r-round trail, broken down by the number k of zero positions, following Eq. (12).

	number of zero positions k							
r	0	1	2	3	4	5	6	$N(r)$
3	1	5	3	–	–	–	–	**9**
4	1	6	6	–	–	–	–	**13**
5	1	7	10	1	–	–	–	**19**
6	1	8	15	4	–	–	–	**28**
7	1	9	21	10	–	–	–	**41**
8	1	10	28	20	1	–	–	**60**
10	1	12	45	56	15	–	–	**129**
12	1	14	66	120	70	6	–	**277**
16	1	18	120	364	495	252	28	**1,278**

The argument uses only the XOR structure of the Feistel network and the bijectivity of the round function; it therefore applies verbatim to differential characteristics, with masks replaced by differences. Whether a zero pattern of minimal size is realized by an actual trail depends on the round function. In the minimal pattern of the proof of Theorem 1, every pair of active rounds enclosed by two consecutive zeros must admit nonzero-correlation transitions $b \to a$ and $a \to b$. This reciprocal-pair structure is what the F-function spacing bound of Section 4.2 exploits. To describe it, the nontrivial rounds of a pattern are grouped into *runs*, that is, maximal blocks of consecutive nontrivial rounds, and a run of two rounds $j, j + 1$ with $\lambda_{j-2} = \lambda_{j+1} = 0$ is called an *enclosed pair*, because its two round functions realize the transitions $X \to Y$ and $Y \to X$ of a reciprocal pair. Figure 4 illustrates these notions, together with the spacing condition of Theorem 1, on 7-round patterns.

In the search, the positions not listed in a pattern are explicitly constrained to be nonzero, so different patterns are disjoint.

This decomposition serves as the outer level of the search. For each pattern, the minimum satisfiable surrogate level is determined first. A pattern whose surrogate floor already exceeds the target bound is excluded from further exact enumeration. The remaining low-floor patterns are then searched independently.

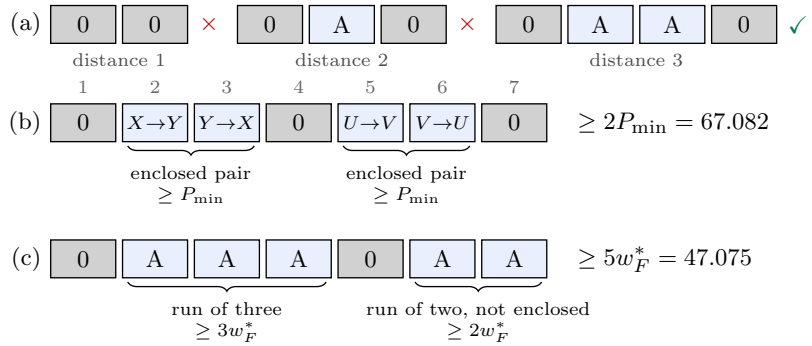


Fig. 4 Zero patterns of 7-round trails in the notation of KISA's evaluation, where 0 marks a trivial round and A a nontrivial one; the boundary positions λ_{-1} and λ_r are not shown. (a) By Theorem 1, two zeros at distance one or two force the all-zero trail. (b) In $Z[0,3,6]$ each pair of active rounds is enclosed by two zeros and realizes the transitions $X \to Y$ and $Y \to X$ of a reciprocal pair, so Theorem 7 charges it $P_{\min}$. (c) In $Z[0,4]$ no run is enclosed and each active round is charged w_F^*; this pattern attains $\mathrm{FSB}(7) = 47.075$

Some patterns admit an additional analytic decomposition. For example, the 7-round pattern $Z[0,3,6]$ contains four active F-functions that form two reciprocal pairs. With $P(X,Y) = w_F(X \to Y) + w_F(Y \to X)$, its weight can be written as

$$w = P(x,y) + P(y,z). \tag{13}$$

Once the exact minimum of P is known, this entire pattern can be bounded analytically; Section 4.2 turns this observation into a bound for every pattern.

3.4 Enumeration Strategy

The search combines the zero patterns of Section 3.3 with the integer surrogate of Section 3.1; the complete procedure is summarized in Section 3.6. For clarity, let $\mathcal{M}_{P,B}$ denote the SAT instance obtained by fixing a zero pattern P and requiring $\mathrm{cost}(T) = B$. A layer (P,B) is called *exhausted* only after repeated solve–block iterations have enumerated all satisfying assignments of $\mathcal{M}_{P,B}$ and the final solver call returns UNSAT.

Pattern-wise search.

Each admissible zero pattern is treated as an independent search space. For a fixed pattern P, the bound $\mathrm{cost}(T) \le b$ is increased until the first satisfiable value is reached. If $\mathrm{cost}(T) \le b-1$ is UNSAT and $\mathrm{cost}(T) \le b$ is SAT, then $b_P = \min_{T \in P} \mathrm{cost}(T) = b$ is the exact surrogate floor of that pattern. The subsequent layer-wise search for P starts at b_P.

Exact evaluation of SAT solutions.

Every satisfying assignment is decoded into a complete linear trail. Its true weight is recomputed from the actual S-box LAT entries and the exact modular-addition

weights. The lightest exact trail obtained over all processed assignments is retained as the current incumbent (T^*, w^*).

Completion for an isolated F-function.

When a single F-function is optimized independently of neighboring rounds, some external input-mask bytes are free after the internal masks have been fixed. For a fixed S-box output mask b, the table $W_{\min}(b) = \min_{a \in \{0,1\}^8} w_S(a \to b)$ is precomputed over the 256 input-byte masks. The four free bytes are then completed independently by table lookup before the exact trail weight is evaluated. Interior multi-round input masks are not completed in this way because they are pinned by the neighboring Feistel rounds; the first and last rounds of a multi-round trail have free input masks and are completed in the same manner.

Because the free bytes are completed outside the solver, the corresponding variables may also be omitted from the blocking clauses, so that one blocking clause removes an entire family of assignments at once. The following observation records why this does not affect Proposition 1.

Proposition 3 (Projection blocking) *Let* Π *be a subset of the model variables such that*

(i) $\mathrm{cost}(T)$ *does not depend on the variables outside* Π*, and*
(ii) for every assignment to Π*, the minimum of* $w(T)$ *over the remaining variables is computed exactly outside the solver.*

Then blocking on Π *instead of on the full assignment preserves both the exhaustion of a surrogate level and the incumbent value* w^* *produced by the search.*

Proof A blocking clause on Π removes exactly the assignments agreeing with the returned solution on Π. By (i) all of them lie in the same surrogate level, so no level boundary is crossed and the level is still declared exhausted only when no assignment of that level remains. By (ii) the lightest trail among the removed assignments has already been evaluated, so the incumbent is not weakened. □

Condition (ii) holds for an isolated F-function because the free input bytes influence only their own S-box transitions and their optimal values are given by $W_{\min}$. It fails for interior rounds of a multi-round trail, where the corresponding masks are pinned by the neighboring F-functions. The completion therefore applies to a single F-function, whereas the interior rounds of multi-round trails are handled by the layer and decision levels.

Local exact refinement of a candidate.

For selected intermediate 8-bit mask variables, the external search driver keeps all other masks fixed and tests all 256 values of the selected byte. Only replacements that preserve local propagation are considered, and a replacement is retained only when the modified trail passes the complete trail check and its recomputed exact weight decreases. The selected coordinates are revisited until no further decrease is obtained.

This is external post-processing of a feasible trail; it is not performed by the SAT solver.

Layer exhaustion and certification.

For each integer cost B, every pattern with $b_P \leq B$ is processed at exact cost B. Each returned assignment is blocked and the same layer is solved again until UNSAT. Patterns with $b_P > B$ have already been proved to contain no assignment at cost at most B by the floor scan. Consequently, after all relevant patterns at level B have been exhausted, the complete search space with surrogate cost at most B has been covered. If $w^* < B + 1$, Proposition 1 certifies w^* as the exact optimum. Otherwise the next layer is either enumerated in the same way or, at the decision level, decided by Proposition 2.

The five strategies above define how individual patterns and cost layers are processed. Their interaction is summarized in Section 3.6.

3.5 Hull Enumeration and Evidence Levels

The trail search above is designed to find and certify light individual trails. Hull enumeration requires a different blocking rule. The external input and output masks are fixed, a zero pattern is fixed, and every complete internal trail assignment is blocked after evaluation. Distinct internal trails must remain distinct because each contributes separately to the squared-correlation sum.

For an enumerated set $\mathcal{E}$,

$$\widehat{w}_{\mathrm{hull}} = -\frac{1}{2}\log_2 \sum_{T \in \mathcal{E}} 2^{-2w(T)}.$$

The set $\mathcal{E}$ used later contains the trails of one zero pattern within a finite exact-weight window; Section 5.1 shows that, for every endpoint pair reported here, no other pattern contributes within that window. The reported hull weights are finite-window values under the independent-round-key model; they refer to fixed endpoints and are not optimized over all endpoints.

The evidence levels are kept separate throughout the paper: an exact optimum is claimed only when the stopping condition of Proposition 1 is met; a lower bound follows from a completed UNSAT region or an analytic argument; a found trail supplies only an upper bound; and an interrupted or resource-limited layer is never used to prove a lower bound.

Every computational proof in this paper relies on the correctness of the SAT solver and of the encoding; no independently checked DRAT or LRAT certificates are provided for the UNSAT answers. Accordingly, "certified" refers to complete SAT/UNSAT-based optimality arguments conditional on this correctness, not to checkable proof certificates. The encoding is cross-checked against reference test vectors, exhaustive small-width correlations, and separate computations of the same quantities within this work (Appendix A). Implementation details and the reported computing environment are collected in the same appendix.

3.6 The Divide-and-Decide Procedure

The complete search assembles the ingredients of this section into the three levels outlined at its beginning. The target $\mathcal{X}$ is a number of rounds r, an isolated F-function, or a reciprocal pair; $\mathcal{P}$ is the set of admissible zero patterns of $\mathcal{X}$, generated by Corollary 1 (a single structure for an isolated F or a pair), each with its analytic lower bound $\mathrm{LB}(P)$, which is 0 for an isolated F or a pair; T_0 is any known trail of $\mathcal{X}$, which supplies the initial incumbent (T^*, w^*); and d is the number of cost layers enumerated above the lowest floor before the decision level takes over. For a pattern P, the *floor* b_P is the least cost at which its SAT model is satisfiable, $\mathcal{M}_P(B)$ is its SAT model restricted to $\mathrm{cost}(T) = B$, and $\mathcal{M}_P(B, t)$ is the decision model of Eq. (7). For a satisfying assignment σ, DecodeTrail(σ) reconstructs the concrete SEED mask trail, whose exact weight $w(T)$ is computed from the actual LAT entries and the exact modular-addition weights, and Improve applies the optional accelerations of Section 3.4, namely the LAT completion of an isolated F-function and the bytewise local refinement. With this notation, the procedure consists of four steps, and Algorithm 1 gives its pseudocode.

Step 1. Partition by zero pattern and prune. All admissible patterns are generated, and every pattern with $\mathrm{LB}(P) \geq w^*$ is discarded without a solver call; for SEED, LB is the F-function spacing bound of Theorem 7. For each remaining pattern, the cost bound $\mathrm{cost}(T) \leq B$ is raised until the first satisfiable value, which is the floor b_P; a pattern that stays UNSAT below w^* is discarded as well.

Step 2. Exhaust the lowest cost layers. Starting from the smallest floor, the layers are processed in ascending order. At layer B, every remaining pattern with $b_P \leq B$ is searched under $\mathrm{cost}(T) = B$; every satisfying assignment is decoded, improved, and evaluated exactly, and is then blocked until the layer is UNSAT. If the incumbent falls below $B + 1$, Proposition 1 certifies it, which settles the isolated F-function, the reciprocal pair, and 3 and 4 rounds.

Step 3. Decide the residual layers. After d layers, every remaining layer (P, B) with $B < w^*$ is closed by the decision model $\mathcal{M}_P(B, w^*)$. An UNSAT answer closes the layer; every satisfying assignment is evaluated exactly and blocked, and a lighter trail also replaces the incumbent and lowers the allowed excess of every later decision.

Step 4. Certify the incumbent. When every layer below w^* of every remaining pattern is closed, the incumbent is optimal.

Theorem 3 (Correctness of D&D) *When Algorithm 1 returns (T^*, w^*), every nonzero trail of $\mathcal{X}$ has weight at least w^*, so w^* is the exact optimum of $\mathcal{X}$.*

Proof Let T be a trail of $\mathcal{X}$ and let P be its zero pattern, which is admissible by Theorem 1. If P was discarded in Step 1, then either $w(T) \geq \mathrm{LB}(P) \geq w^*$, because $\mathrm{LB}(P)$ bounds every trail of P, or P has no assignment of cost below w^*, in which case $w(T) \geq \mathrm{cost}(T) \geq w^*$ by Eq. (4). Otherwise let $B = \mathrm{cost}(T)$. If $B < b_P$, then T cannot exist; if B was exhausted in Step 2, then T was enumerated and $w(T) \geq w^*$; if B was decided in Step 3, then Proposition 2

gives $w(T) \geq w^*$, because a later decrease of the incumbent only weakens the threshold of an earlier decision; and if $B \geq w^*$, then $w(T) \geq B \geq w^*$. □

Algorithm 1 Divide-and-decide search for a SEED linear trail

Input: target $\mathcal{X}$; admissible zero patterns $\mathcal{P}$ with lower bounds LB; a known trail T_0 of $\mathcal{X}$; enumeration depth d

Output: an optimal nonzero trail T^* of $\mathcal{X}$ and its exact weight w^*

$(T^*, w^*) \leftarrow (T_0, w(T_0))$
Phase I, pattern level: partition by zero pattern and prune.
$\mathcal{P}' \leftarrow \{P \in \mathcal{P} : \mathrm{LB}(P) < w^*\}$ ▷ FSB for SEED
for each $P \in \mathcal{P}'$ **do**
    $b_P \leftarrow$ least $B < w^*$ with $\mathcal{M}_P(\mathrm{cost} \leq B)$ SAT, or ∞ ▷ floor of P
end for
$\mathcal{P}' \leftarrow \{P \in \mathcal{P}' : b_P < \infty\}$; **if** $\mathcal{P}' = \emptyset$ **then return** (T^*, w^*)
Phase II, layer level: exhaust the lowest layers.
$B_0 \leftarrow \min_{P \in \mathcal{P}'} b_P$
for $B \leftarrow B_0$ **to** $B_0 + d$ **do**
    for each $P \in \mathcal{P}'$ with $b_P \leq B$ **do**
        while $\mathcal{M}_P(B)$ has a satisfying assignment σ **do**
            $T \leftarrow \mathrm{Improve}(\mathrm{DecodeTrail}(\sigma))$
            if $w(T) < w^*$ **then**
                $(T^*, w^*) \leftarrow (T, w(T))$
            end if
            block σ in $\mathcal{M}_P(B)$
        end while
    end for
    if $w^* < B + 1$ **then**
        return (T^*, w^*) ▷ Proposition 1
    end if
end for
Phase III, decision level: decide the residual layers.
for each $P \in \mathcal{P}'$ with $b_P < w^*$ **do**
    for $B \leftarrow \max(b_P, B_0 + d + 1)$ **while** $B < w^*$ **do**
        while $\mathcal{M}_P(B, w^*)$ has a satisfying assignment σ **do** ▷ Eq. (7)
            $T \leftarrow \mathrm{DecodeTrail}(\sigma)$
            if $w(T) < w^*$ **then**
                $(T^*, w^*) \leftarrow (T, w(T))$ ▷ lighter trail; allowed excess shrinks
            end if
            block σ ▷ never returned again
        end while ▷ layer closed, Proposition 2
    end for
end for
return (T^*, w^*) ▷ Theorem 3

4 Exact Single-Trail Optima of SEED

This section reports the main single-trail results for SEED, from exact local F-function optima to exact reduced-round optima and all-round consequences. Let w_r^* denote the minimum exact weight among all nonzero r-round linear trails.

4.1 Exact Optima of the SEED Round Function

The first step is to determine the smallest possible contribution of a nontrivial SEED F-function. Throughout this subsection, let

$$\alpha = (\alpha_C, \alpha_D), \qquad \beta = (\beta_C, \beta_D)$$

denote the 64-bit linear masks at the external input and output of one F-function, respectively, where each component is a 32-bit word mask. For fixed external masks, let $w_F(\alpha \to \beta)$ denote the minimum exact weight over all valid internal mask assignments connecting α to β.

Theorem 4 *The minimum exact weight of a valid linear trail through one SEED F-function is*

$$\mathrm{w}_F^* = \min_{\alpha \neq 0} w_F(\alpha \to \beta) = 6 + \log_2\left(\frac{32}{3}\right) \approx 9.415,$$

where by Eq. (10) the condition $\alpha \neq 0$ holds if and only if $\beta \neq 0$.

Computational proof The SAT model is UNSAT for cost ≤ 8. At cost 9, all 24,856 internal structures are exhausted in the sense of Proposition 3, the free external input bytes being completed optimally outside the solver. The minimum exact completion contains one active S-box with $|\mathrm{LAT}| = 12$, one with $|\mathrm{LAT}| = 16$, and modular-addition weights $(0, 0, 3)$. Its exact weight is therefore

$$\log_2\left(\frac{32}{3}\right) + 3 + 3 = 6 + \log_2\left(\frac{32}{3}\right).$$

The next unexplored integer cost is 10, which is already larger than the incumbent. Proposition 1 therefore proves global optimality. □

One optimum is

$$\texttt{0000000100000000} \longrightarrow \texttt{0181818120002021}.$$

Its local decomposition is shown in Table 5. In particular, only two S-boxes are active. The noninteger excess above the surrogate is entirely caused by the $|\mathrm{LAT}| = 12$ transition.

Table 5 Weight decomposition of an optimal F-function trail.

component	local event	exact weight
G_1	inactive	0
G_2	one active S-box, $\lvert\mathrm{LAT}\rvert = 12$	$\log_2(32/3) \approx 3.415$
G_3	one active S-box, $\lvert\mathrm{LAT}\rvert = 16$	3
additions	carry weights $(0, 0, 3)$	3
total		$6 + \log_2(32/3) \approx 9.415$

Two restricted local problems are also useful later. The first is the iterative case, in which the input and output masks of F are identical.

Theorem 5 *Among all nonzero iterative F-function trails, that is, trails whose 64-bit input and output masks satisfy $\alpha = \beta$, the minimum exact weight is* 16.771*, attained at $\alpha = \beta =$ 0300030003000300.*

Computational proof All surrogate levels through cost 15 are UNSAT. Cost 16 contains 86 assignments and is completely exhausted; its minimum exact weight is $16.770580\ldots$; since the next integer cost is 17, the stopping rule proves optimality within the iterative restriction. □

The second restricted problem couples an F-trail with its reciprocal. For 64-bit external F-function masks X and Y, define $P(X, Y) = w_F(X \to Y) + w_F(Y \to X)$, where the first term connects input mask X to output mask Y, and the second connects input mask Y to output mask X.

Theorem 6 *The minimum reciprocal-pair weight is*

$$P_{\min} = \min_{X \neq 0} P(X, Y) = 33.541,$$

where, as in Theorem 4, $X \neq 0$ holds if and only if $Y \neq 0$. It is attained by $X = Y =$ 0300030003000300.

Computational proof The crossed two-F model is UNSAT for cost at most 29 and exhausts costs 30, 31, 32, and 33, containing 184, 244, 32,252, and 93,158 assignments, respectively. The best exact weight reaches $33.541160\ldots$ at cost 32 and is not improved at cost 33. The next integer cost is 34, so the incumbent is exact. □

Theorems 5 and 6 will also be used to interpret the historical iterative analysis of SEED.

4.2 The F-Function Spacing Bound

For SEED, Theorem 1 and the two local optima of Section 4.1 already determine a lower bound for any number of rounds, without any multi-round search. The classical argument counts nontrivial round functions; the bound below also recognizes that the two rounds enclosed between consecutive zero positions form a reciprocal pair.

Theorem 7 (F-function spacing bound, FSB) *Let* $w_F^* = 6 + \log_2(32/3) \approx 9.415$ *and* $P_{\min} \approx 33.541$ *be the minimum weights of a nontrivial SEED round-function approximation and of a reciprocal pair (Theorems 4 and 6). Let* T *be a nonzero* r*-round linear trail of SEED, and partition its nontrivial rounds into maximal runs of consecutive rounds. Call a run* enclosed *if it consists of two rounds* $j, j+1$ *with* $\lambda_{j-2} = \lambda_{j+1} = 0$*. Then*

$$w(T) \;\geq\; \sum_R b(R), \qquad b(R) = \begin{cases} P_{\min}, & \text{if } R \text{ is enclosed,} \\ |R|\, w_F^*, & \text{otherwise,} \end{cases}$$

where the sum runs over the runs R *of* T*. For a zero pattern* Z*, the right-hand side depends only on* Z *and is denoted* $\mathrm{FSB}(Z)$*. Consequently* $w_r^* \geq \mathrm{FSB}(r)$*, where* $\mathrm{FSB}(r)$ *denotes the minimum of* $\mathrm{FSB}(Z)$ *over all admissible* r*-round zero patterns.*

Proof The weight of T is the sum of the weights of its nontrivial round functions, and each of them is at least w_F^* because its input and output masks are nonzero. For an enclosed run $\{j, j+1\}$, Eq. (9) with $\lambda_{j-2} = \lambda_{j+1} = 0$ gives $\alpha_j = \lambda_j$, $\beta_j = \lambda_{j-1}$, $\alpha_{j+1} = \lambda_{j-1}$, and $\beta_{j+1} = \lambda_j$. The two round functions therefore realize the transitions $X \to Y$ and $Y \to X$ with $X = \lambda_j$ and $Y = \lambda_{j-1}$, and their total weight is at least $P_{\min}$. The runs are disjoint, so their contributions add. By Theorem 1, every nonzero trail has exactly one admissible zero pattern, which gives the second statement. □

Figure 4(b) and (c) show the two kinds of runs on 7-round patterns. Ignoring the enclosed runs recovers the simpler count

$$\mathrm{w}_r^* \;\geq\; \mathrm{w}_F^* \left(r - \lceil r/3 \rceil\right), \tag{14}$$

which follows from Theorem 1 alone. The refinement matters because enclosed pairs are expensive: for SEED, $P_{\min} = 33.541$ exceeds even three single round functions, $3w_F^* = 28.245$. In that case FSB has a closed form.

Corollary 2 *If* $P_{\min} \geq 3w_F^*$*, then* $\mathrm{FSB}(r) = (r - \lceil r/4 \rceil)\, w_F^*$ *for every* $r \geq 1$*. In particular this holds for SEED, where* $P_{\min} = 33.541 \geq 3w_F^* = 28.245$*.*

Proof The boundary positions λ_{-1} and λ_r make no round trivial, so removing zeros at these positions leaves the trivial rounds unchanged and can only turn an enclosed run into one that is not enclosed. Since $P_{\min} \geq 2w_F^*$, this cannot increase the bound of Theorem 7, and it suffices to consider patterns whose zeros $z_1 < \cdots < z_t$ all lie among $\lambda_0, \ldots, \lambda_{r-1}$. Let p be the number of indices i with $z_{i+1} - z_i = 3$. The enclosed runs are then exactly the p pairs between such zeros, so the bound equals

$$(r - t)\, w_F^* + p\,(P_{\min} - 2w_F^*) \;\geq\; \big(r - (t - p)\big)\, w_F^*,$$

because $P_{\min} - 2w_F^* \geq w_F^*$. For $t \geq 1$, Theorem 1 shows that the $t - 1$ gaps $z_{i+1} - z_i$ consist of p gaps equal to 3 and $t - 1 - p$ gaps of at least 4, and they sum to at most $r - 1$. Hence $3p + 4(t - 1 - p) \leq r - 1$, that is, $4(t - p) \leq r + 3 - p \leq r + 3$, which gives $t - p \leq \lfloor (r+3)/4 \rfloor = \lceil r/4 \rceil$; the case $t = 0$ is immediate. This proves the lower bound. Zeros at $\lambda_0, \lambda_4, \lambda_8, \ldots$ give $t = \lceil r/4 \rceil$ and $p = 0$, which attains it. □

Compared with Eq. (14), the corollary replaces $\lceil r/3 \rceil$ by $\lceil r/4 \rceil$: once a reciprocal pair costs at least three single round functions, the minimizing patterns avoid enclosed pairs, which are exactly the iterative $0AA$ patterns, and place their zeros four positions apart. With $w_F^* = 9.415$ (Theorem 4) and $P_{\min} = 33.541$ (Theorem 6), selected SEED values are listed in Table 6.

Table 6 Lower bounds on the weight of an r-round SEED linear trail obtained without multi-round search, namely the active-round count of Eq. (14) and $\mathrm{FSB}(r) = (r - \lceil r/4 \rceil)\, w_F^*$ of Corollary 2, compared with the exact optima determined by D&D.

r	Eq. (14)	FSB(r)	exact optimum
3	18.830	**18.830**	18.830
4	18.830	**28.245**	33.541
5	28.245	**28.245**	43.472
6	37.660	**37.660**	55.027
7	37.660	**47.075**	65.220
8	47.075	**56.490**	–
10	56.490	**65.905**	–
12	75.320	**84.735**	–
14	84.735	**94.150**	–
16	94.150	**112.980**	–

Computing FSB.

Computing FSB(r) requires no solver. The admissible zero patterns are generated by the recursion of Corollary 1, which turns the spacing condition of Theorem 1 into a short recursive routine, and each is charged by the rule of Theorem 7. By Eq. (12) there are 41 admissible patterns at 7 rounds and 1,278 at 16 rounds (Table 4). In single-threaded CPython 3.12, generating and charging the 1,278 patterns at 16 rounds takes about 3.4 ms, and all patterns from 1 to 16 rounds take about 9.2 ms. For SEED even this enumeration is unnecessary, since Corollary 2 gives the value directly; the enumeration is the general procedure when $P_{\min} < 3w_F^*$. The same rule gives each individual pattern its own analytic bound, which serves as a filter for the pattern-wise search of Section 3: a pattern whose bound exceeds the current incumbent contains no improving trail and can be discarded without any SAT call, as happens for $Z[0,3,6]$ at 7 rounds. The proof of Theorem 7 uses SEED only through w_F^*, $P_{\min}$, and Theorem 1. The same charging therefore applies to any Feistel cipher whose round function is a permutation once its two local optima are known, although this paper uses it only for SEED.

Three consequences are used below. First, $\mathrm{FSB}(3) = 18.830$ is attained, so the 3-round optimum follows without a 3-round search (Theorem 8). Second, FSB is the analytic part of every pattern-wise argument in this paper: the 4-round pattern $Z[0,3]$, the 7-round pattern $Z[0,3,6]$, the 6-round free-prefix pattern $Z[-1,2,5]$, and the historical 15-round pattern $(0AA)^5$ consist of enclosed pairs, so their bounds $P_{\min}$, $2P_{\min}$, $2P_{\min}$, and $5P_{\min}$ are instances of Theorem 7. Third, $\mathrm{FSB}(r) > 64$ for every $r \geq 10$,

so from 10 rounds onwards no single trail yields a basic linear distinguisher below the 2^{128} codebook, and the full 16-round cipher satisfies $\mathrm{w}_{16}^{*} \geq 112.980$.

FSB is coarse at 5, 6, and 7 rounds, where it gives 28.245, 37.660, and 47.075 against the exact optima 43.472, 55.027, and 65.220. Runs of three or more rounds are charged only w_F^{*} per round, although the Feistel mask relations generally prevent all of their round functions from attaining the isolated minimum simultaneously. The layer and decision levels of D&D close this gap.

4.3 Exact Optima for Three to Six Rounds

The 3-round case can be solved analytically once the exact one-F bound is known.

Theorem 8 *The minimum weight of a 3-round SEED linear trail is*

$$\mathrm{w}_3^{*} = 2\mathrm{w}_F^{*} = 12 + 2\log_2\left(\frac{32}{3}\right) \approx 18.830.$$

Proof Theorem 7 gives $\mathrm{w}_3^{*} \geq \mathrm{FSB}(3) = 2\mathrm{w}_F^{*}$. The pattern $Z[1]$, with a trivial middle round, attains this bound: its two active F-functions have free input masks and a common output mask $\lambda_0 = \lambda_2$, so both can use the optimal isolated-F transition of Theorem 4. Direct SAT search and a scan of all nine 3-round zero patterns independently return the same value. □

The 3-round optimum does not extend to four rounds. Its pattern $Z[1]$ cannot be preceded or followed by a trivial round without violating Theorem 1, and the only 4-round pattern with two nontrivial rounds is $Z[0,3]$, whose two active rounds form an enclosed run.

Corollary 3 *The minimum weight of a 4-round SEED linear trail is* $\mathrm{w}_4^{*} = P_{\min} \approx 33.541$. *It is attained by the iterative* $0AA0$ *trail with* $\alpha = \beta =$ *0300030003000300 in both active rounds.*

Proof By Theorem 7, every trail in $Z[0,3]$ weighs at least $P_{\min}$, and the stated trail attains this value by Theorem 6. Every other admissible 4-round pattern has at least three nontrivial rounds. A complete 4-round floor probe shows that all of them are UNSAT for cost ≤ 35, so their trails weigh at least 36. □

The value coincides with the 4-round evaluation of KISA's report, so that evaluation is optimal; the comparison is taken up in Section 4.6.

For five and six rounds, the layer level alone is not enough: their lowest layers yield an incumbent, but the layers between the incumbent and the next integer contain millions of assignments. The decision level closes them. Table 7 lists every part of the resulting certificates, and the thresholds 43.4725, 55.0275, and 65.2205 of the decisions lie slightly above the incumbents, so that rounding of the reported weights cannot hide a lighter trail.

Table 7 Certificates of the exact 5-, 6-, and 7-round optima. Floor rows are proved by UNSAT of the cost bound; for the enumeration rows and the decision rows of Proposition 2, the last two columns give the number of exhausted assignments and the total solver time.

rounds	patterns	cost	level	result	assignments	time
5	16 patterns	≤ 43	floor	UNSAT	–	–
	$Z[0,3]$, $Z[1,4]$	41–43	enumeration	none below 43.472	–	–
	$Z[0,4]$	38–40	enumeration	lightest 43.472	2,172,912	3.7 h
	$Z[0,4]$	41–43	decision	UNSAT	–	114 s
6	10 patterns	–	FSB	≥ 56.490	–	–
	$Z[0,4]$, $Z[1,5]$	49–50	enumeration	none below 55.027	501,120	1.9 h
	18 remaining	51–55	decision	UNSAT[a]	–	4.1 h
7	36 patterns	≤ 65	floor	UNSAT	–	–
	$Z[0,3,6]$	–	FSB	67.082	–	–
	$Z[1,5]$	59–60	enumeration	none below 65.220	837,888	3.3 h
	4 remaining	61–65	decision	UNSAT[a]	–	2.5 h

The 501,120 assignments of the 6-round enumeration split evenly between the two patterns, 250,560 each, which took 3,460 s and 3,521 s. [a]UNSAT after blocking, as in Section 3.1, the trails whose weight equals the incumbent: 40 in each of the 6-round patterns $Z[0,4]$ and $Z[1,5]$ at cost 51, and 40 in the 7-round pattern $Z[1,5]$ at cost 61. No false positive occurred.

Theorem 9 *The minimum 5-round trail weight is* $\mathrm{w}_5^* = 43.472$.

Proof Among the 19 admissible 5-round patterns, 16 have no assignment of cost at most 43, so their trails weigh at least 44 by Eq. (4). The patterns $Z[0,3]$ and $Z[1,4]$ have floor 41, and their layers of cost 41 to 43 are exhausted without a trail lighter than 43.472. For $Z[0,4]$, the layers of cost 38, 39, and 40 are exhausted with 14,400, 208,640, and 1,949,872 assignments; the lightest has weight 43.472 and cost 38. The decisions $\mathcal{M}_{Z[0,4]}(B, 43.4725)$ for $B = 41, 42, 43$ are UNSAT, and trails of cost at least 44 weigh at least 44. By Theorem 3, no 5-round trail is lighter than 43.472. □

Theorem 10 *The minimum 6-round trail weight is* $\mathrm{w}_6^* = 55.027$.

Proof Of the 28 admissible 6-round patterns, 10 have $\mathrm{FSB}(Z) \geq 56.490$ and are discarded by Theorem 7. Among the remaining 18, only the mirror patterns $Z[0,4]$ and $Z[1,5]$ have floor at most 50. For each of them, the layers of cost 49 and 50 are exhausted with 14,480 and 236,080 assignments, and none is lighter than 55.027; the incumbent itself, a trail of weight 55.027 and cost 51, is reached from the cost-50 layer by local refinement. For all 18 patterns, the decisions $\mathcal{M}_P(B, 55.0275)$ with $51 \leq B \leq 55$ are UNSAT, after the 40 trails of weight exactly 55.027 found at cost 51 in each of $Z[0,4]$ and $Z[1,5]$ are blocked. Trails of cost at least 56 weigh at least 56. By Theorem 3, no 6-round trail is lighter than 55.027. □

4.4 Exact Optimum for Seven Rounds

The 7-round search contains 41 admissible zero patterns, using the $Z[i_1, \ldots, i_t]$ notation defined above. A complete floor probe through cost 65 finds only five patterns

with any satisfying surrogate level in this range: $Z[1,5]$, $Z[0,3,6]$, $Z[0,4]$, $Z[0,6]$, and $Z[2,6]$. Their first satisfiable costs are listed in Table 8. The remaining 36 patterns have no SAT assignment with cost at most 65.

Table 8 Seven-round patterns with surrogate floor at most 65.

zero pattern	trivial rounds	surrogate floor
$Z[1,5]$	2,6	59
$Z[0,3,6]$	1,4,7	60
$Z[0,4]$	1,5	61
$Z[0,6]$	1,7	61
$Z[2,6]$	3,7	61

The reciprocal-pair result excludes one of the low-floor patterns.

Corollary 4 *Every 7-round trail in pattern $Z[0,3,6]$ has weight at least $2P_{\min} = 67.082$.*

Proof The zeros of $Z[0,3,6]$ enclose two pairs of active rounds, so Theorem 7 gives weight at least $2P_{\min}$. □

Theorem 11 *The minimum 7-round trail weight is $\mathrm{w}_7^* = 65.220$.*

Proof The 36 patterns outside Table 8 have no assignment of cost at most 65, so their trails weigh at least 66, and Corollary 4 excludes $Z[0,3,6]$. For $Z[1,5]$, cost 59 contains 57,376 assignments with minimum exact weight 69.109, and cost 60 contains 780,512 assignments, none lighter than 65.220; the incumbent, a trail of weight 65.220 and cost 61, is reached from the cost-60 layer by local refinement. For the four patterns $Z[1,5]$, $Z[0,4]$, $Z[0,6]$, and $Z[2,6]$, the decisions $\mathcal{M}_P(B, 65.2205)$ with $61 \le B \le 65$ are UNSAT, after the 40 trails of weight exactly 65.220 found at cost 61 in $Z[1,5]$ are blocked, and the patterns $Z[0,4]$, $Z[0,6]$, and $Z[2,6]$ have no assignment below their floor 61. Trails of cost at least 66 weigh at least 66. By Theorem 3, no 7-round trail is lighter than 65.220. □

One optimal trail has endpoints

$$(\Gamma L_{\text{in}}, \Gamma R_{\text{in}}) = (\texttt{000200020833383b}, \texttt{0300000000000000}),$$
$$(\Gamma L_{\text{out}}, \Gamma R_{\text{out}}) = (\texttt{0300000000000000}, \texttt{000200020833383a}).$$

Without the decision level, closing the cost-61 layer of $Z[1,5]$ would require enumerating it completely: a continued enumeration processed 1,648,172 of its assignments, and the growth by a factor of about 13.6 between costs 59 and 60 suggests a layer of order 10^7 assignments, beyond what the blocking-clause approach handles on the reported machine. The decisions close each of the 20 residual layers instead, the hardest in about 36 minutes. The mirror patterns $Z[0,4]$ and $Z[2,6]$, whose trails correspond under reversal of the round order, give the same outcome in every layer.

4.5 Summary of the Single-Trail Results

Figure 5 summarizes the single-trail results established above: panel (a) compares them, together with the linear hulls of Section 5.2, with KISA's evaluation, and panel (b) shows the analytic lower bounds for every number of rounds. All single-trail values from 3 to 7 rounds in panel (a) are exact optima, whereas the curves of panel (b) are lower bounds only.

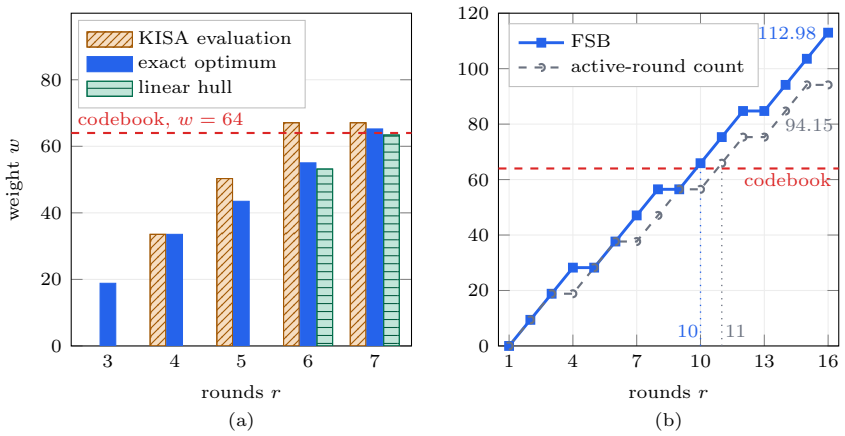


Fig. 5 Single-trail weights and linear hulls of SEED. (a) Exact single-trail optima of this section and linear hulls of Section 5.2 beside the iterative approximations of KISA's evaluation. (b) The F-function spacing bound $(r - \lceil r/4 \rceil)\, w_F^*$ of Corollary 2 and the active-round count $(r - \lceil r/3 \rceil)\, w_F^*$ of Eq. (14); dotted lines mark the first round at which each exceeds the codebook; the values of KISA's evaluation are taken from [16]

The excess of Eq. (5) explains both why the layer level alone leaves gaps at 5, 6, and 7 rounds and why the decision level closes them quickly. Only 1275 of the 60945 valid nonzero S-box transitions attain the minimum weight 3 (Table 3), so a trail whose active S-boxes all attain it is exceptional. The optimal 5-, 6-, and 7-round trails carry excesses of 5.472, 4.027, and 4.220 bits above their costs 38, 51, and 61. A lighter trail in a higher layer would need a smaller excess, and the allowed excesses of the decisions, at most 2.473, 4.028, and 4.221 bits, leave only a few combinations of S-box classes, which the solver refutes in at most about 36 minutes per layer.

Finally, FSB is sharp on the historical 15-round zero pattern $(0AA)^5$. Its zeros at $\lambda_0, \lambda_3, \ldots, \lambda_{15}$ enclose five pairs, so by Theorem 7 every conforming trail has weight at least $5P_{\min} = 167.706$. Assigning the optimal iterative mask 0300030003000300 to the active F-functions attains this value. Thus 167.706 is exact within the $(0AA)^5$ pattern; it is not a claim about the unknown unrestricted 15-round optimum.

4.6 Revisiting Earlier Linear Cryptanalysis of SEED

KISA's evaluation.

The 1999 SEED security evaluation [16] reports linear probabilities, which are converted here using $\mathrm{w} = -\frac{1}{2}\log_2 LP$. Its quoted local conventions are reproduced, including $\max|\mathrm{LAT}| = 16$, the reported $03 \to 01$ S-box transition with $|\mathrm{LAT}| = 14$, the G approximation $\mathtt{03000300} \to \mathtt{01000100}$, and the two modular-addition examples $(\mathtt{02000200}, \mathtt{02000200}) \to \mathtt{03000300}$ and $(1, 1) \to 1$. These checks rule out a simple LP or byte-order mismatch in the subsequent comparison.

Fixing the report's iterative mask $\alpha = \beta = \mathtt{0300030003000300}$ gives weight 16.771, and Theorem 5 proves that this is the best iterative F-function approximation. The unrestricted isolated-F optimum of Theorem 4 is nevertheless 9.415, about 7.36 bits lighter. Likewise, Theorem 6 proves that the report's 15-round $(0AA)^5$ construction has exact weight 167.706 within that zero pattern. This is a restricted-pattern optimum. Without the pattern restriction, Theorem 7 gives $\mathrm{w}^*_{15} \geq \mathrm{FSB}(15) = 103.565$, so the unrestricted 15-round optimum lies in [103.565, 167.706].

Table 1 places the reported values beside those obtained here. At 4 rounds the report's $0AA0$ approximation is optimal: Corollary 3 gives $\mathrm{w}^*_4 = P_{\min} = 33.541$, which is the reported value. At 5, 6, and 7 rounds the exact optima of Theorems 9, 10, and 11 improve the reported evaluations by about 6.84, 12.05, and 1.86 bits.

Wu, Ma, and Feng (2003).

Wu et al. [34] construct a 15-round approximation from three ingredients: the one-bit S-box relation $X[0] \approx S_1(X)[0]$, for which $|\mathrm{LAT}| = 12$; the observation that the 32-bit mask $\mu = \mathtt{00010001}$ is a fixed point of the G linear layer, so that G activates only its two S_1 positions and has linear probability $9^2/2^{20}$; and the linearity of the least significant bit of modular addition. Propagating μ through F yields three F-function approximations, each with linear probability $9^4/2^{40}$ (weight 13.660). These are combined into a 5-round iterative approximation with four active F-functions, and three iterations give the reported 15-round weight 163.92.

The S-box relation, the fixed-point property of μ, and the G correlation are reproduced exactly in the present framework. For the same F-function endpoints, the exact carry-mask model of Section 3.2 gives a minimum weight of 15.660, with cost ≤ 13 proved UNSAT. The value 13.660 could therefore not be reproduced under the exact model; the difference lies in the treatment of the modular additions: μ has active bits at positions 0 and 16, and the exact model assigns the addition mask $\mathtt{00010001}$ weight 16.

5 Linear Hulls and a Seven-Round Key-Recovery Attack

This section extends the single-trail results to linear hulls, uses the 6-round hull in a 7-round key-recovery attack, and explains why this route ends at seven rounds.

5.1 Hull Enumeration

For an r-round instance, let

$$\Lambda_{\text{in}} = (\Gamma L_0, \Gamma R_0), \qquad \Lambda_{\text{out}} = (\Gamma L_r, \Gamma R_r)$$

denote the fixed 128-bit input and output linear masks, expressed as the two 64-bit Feistel-branch masks. Let P be a fixed structural zero pattern and let $w_{\min}$ be the lightest trail found for these endpoints in that pattern. For a true-weight window Δ, define

$$\mathcal{E}(P, \Lambda_{\text{in}}, \Lambda_{\text{out}}, \Delta) = \left\{ T : \begin{array}{l} T \text{ follows } P, \\ T \text{ connects } \Lambda_{\text{in}} \text{ to } \Lambda_{\text{out}}, \\ w(T) \le w_{\min} + \Delta \end{array} \right\}.$$

Only complete internal trail assignments are blocked during this enumeration. The isolated-F completion and local refinement used to improve single-trail incumbents are disabled, because distinct internal trails are distinct summands of the hull.

Following Eq. (2), the enumerated set gives

$$\widehat{C}^2 = \sum_{T \in \mathcal{E}} 2^{-2w(T)}, \qquad \widehat{w}_{\text{hull}} = -\frac{1}{2} \log_2 \widehat{C}^2. \tag{15}$$

By construction, $\mathcal{E}$ contains only one zero pattern and only trails inside a finite true-weight window; the complete hull of the same endpoints would also contain the trails above the window, which are not included here.

Completeness with respect to the pattern.

The restriction to a single pattern can be checked directly for fixed endpoints. By Eq. (9), the endpoints fix $\lambda_{-1} = \Gamma R_0$, $\lambda_0 = \Gamma L_0$, $\lambda_{r-1} = \Gamma R_r$, and $\lambda_r = \Gamma L_r$, so a pattern is compatible with them only if its zero set contains exactly those of these four positions that are zero. For each compatible pattern other than P, the SAT model with both endpoints fixed and with the disjointness constraints of Section 3.3 is probed at every surrogate cost up to $\lfloor w_{\min} + \Delta \rfloor$. Since $\text{cost}(T) \le w(T)$, an UNSAT answer throughout this range excludes every trail of that pattern from the window. For all three endpoint pairs reported below the outcome is the same: at 6 rounds the two other compatible patterns, $Z[0]$ and $Z[0,3]$, are UNSAT through cost 61, and at 7 rounds all eight other compatible patterns are UNSAT through cost 71. The first satisfiable cost of the enumerated pattern itself (50 at 6 rounds and 60 at 7 rounds) agrees with the first nonempty level of the corresponding hull enumeration. Consequently, within the enumeration window, every trail between the chosen endpoints belongs to P, and the single-pattern restriction removes nothing. The remaining qualifications are the fixed choice of endpoints and the finite window Δ; the reported values make no claim that the selected endpoints are globally optimal.

Two counts are reported for each hull: the number of complete SAT assignments processed at the visited surrogate levels, and the number of trails whose recomputed exact weight lies inside the window.

5.2 Six- and Seven-Round Linear Hulls

Table 9 summarizes the hulls described below.

Six rounds.

The lightest 6-round trail found in Section 4.3 has weight $w_{\min}^{(6)} = 55.027$ and belongs to pattern $Z[0, 4]$, corresponding to trivial rounds 1 and 5. Its fixed external masks are

$$\Lambda_{\text{in}}^{(6)} = (\texttt{0000000000000000}, \texttt{0002000287909415}),$$
$$\Lambda_{\text{out}}^{(6)} = (\texttt{0300000000000000}, \texttt{0002000287909414}).$$

Using the six-bit window $\Delta = 6$, i.e., $w(T) \leq 61.027$, 477,968 SAT assignments are processed and 1,783 trails satisfy the exact weight cutoff. Their accumulated squared correlation gives the hull weight $\widehat{w}_{\text{hull}}^{(6)} = 53.168$, improving the best single trail by $g_6 = 1.859$ bits.

Seven rounds.

The primary 7-round endpoint pair is induced by the lightest trail found in pattern $Z[1, 5]$, whose exact trail weight is $w_{\min}^{(7)} = 65.220$. The fixed masks are

$$\Lambda_{\text{in}}^{(7)} = (\texttt{000200020833383b}, \texttt{0300000000000000}),$$
$$\Lambda_{\text{out}}^{(7)} = (\texttt{0300000000000000}, \texttt{000200020833383a}).$$

With a three-bit true-weight window, 218,272 assignments are processed and 114 trails are accepted, giving $\widehat{w}_{\text{hull}}^{(7)}(\Delta = 3) = 63.573$. Extending the window to six bits processes 1,366,320 assignments and accepts 1,941 trails, giving $\widehat{w}_{\text{hull}}^{(7)}(\Delta = 6) = 63.391$. The corresponding gain over the best single trail is $g_7 = 1.829$ bits. The change from the three-bit to the six-bit window is 0.182 bits. This is an empirical convergence check for the selected endpoint pair and pattern, not a theoretical bound on the contribution of the still-unenumerated tail.

A second endpoint pair, induced by another weight-65.220 trail of the same $Z[1, 5]$ cluster, with input masks (`00030002c5e3e427`, `0300000000000000`) and output masks (`0300000000000000`, `00030002c5e3e426`), is evaluated with the same window $\Delta = 6$. It processes 1,366,320 assignments and accepts 1,320 trails, giving $\widehat{w}_{\text{hull}} = 63.476$ and a gain of 1.744 bits; the accepted count stops increasing from cost 69 onwards. The two 7-round pairs therefore agree to within 0.085 bits, and the 6-round hull gives 1.859 bits. These figures describe only the endpoint pairs examined here; they are not an upper bound on the hull gain of another endpoint pair.

Table 9 Linear hulls computed with window $\Delta = 6$.

rounds	pattern	endpoints	assignments	trails in window	hull weight	gain
6	$Z[0,4]$	primary	477,968	1,783	53.168	1.859
7	$Z[1,5]$	primary	1,366,320	1,941	63.391	1.829
7	$Z[1,5]$	second	1,366,320	1,320	63.476	1.744

The gain is the weight of the best single trail between the same endpoints minus the hull weight. For each row, every other zero pattern compatible with the endpoints (2 patterns at 6 rounds and 8 at 7 rounds) is UNSAT at all surrogate costs inside the window (Section 5.1), so all trails in the window follow the listed pattern.

For the primary 6-round endpoint, $c_{\mathrm{rms}} = 2^{-53.168}$ and $c_{\mathrm{rms}}^{-2} = 2^{106.34}$, far below the codebook, which motivates the one-round extension below; for the primary 7-round endpoint, $c_{\mathrm{rms}}^{-2} = 2^{126.78}$ is already close to the codebook (Section 5.5).

5.3 Seven-Round Key-Recovery Attack

The primary 6-round hull of Section 5.2 is placed in rounds 1–6 and one final SEED round is appended. Let (L_7, R_7) denote the ciphertext state and let K_7 denote the 64-bit round subkey used by the last F-function. The Feistel relation is $R_6 = L_7$ and $L_6 = R_7 \oplus F(L_7, K_7)$. For the 6-round output masks $(\Gamma L_6, \Gamma R_6) = \Lambda_{\mathrm{out}}^{(6)}$, the corresponding output parity becomes

$$\begin{aligned} \Gamma L_6 \cdot L_6 \oplus \Gamma R_6 \cdot R_6 = \Gamma L_6 \cdot R_7 \oplus \Gamma R_6 \cdot L_7 \\ \oplus \Gamma L_6 \cdot F(L_7, K_7). \end{aligned} \tag{16}$$

The first two terms are known from the ciphertext; only the last term depends on the guessed last-round subkey.

The attack would proceed in four steps; it has not been executed, and the figures reported below are analytic estimates.

Step 1. Collect data and form the observable parity. For each of N known plaintext-ciphertext pairs, the plaintext side of the 6-round approximation is evaluated directly. On the ciphertext side, $\Gamma L_6 \cdot R_7$ and $\Gamma R_6 \cdot L_7$ are also known directly. The only missing contribution is $\Gamma L_6 \cdot F(L_7, K_7)$, which depends on the last-round subkey hypothesis.

Step 2. Evaluate the last-round contribution under subkey hypotheses. Since the round key is XORed at the input of F, the missing term depends only on $L_7 \oplus K_7$. The data are therefore accumulated in a table of 2^{64} signed counters indexed by L_7, and the statistics of all 2^{64} hypotheses for K_7 follow from a fast Walsh–Hadamard transform [13] instead of evaluating every key separately for every text.

Step 3. Sort the subkey hypotheses. The 2^{64} candidates are sorted by the magnitude of their linear statistic. With an advantage of a_{adv} bits, the correct subkey lies among the best $2^{64-a_{\mathrm{adv}}}$ candidates.

Step 4. Recover the master key. In the SEED key schedule, the subkey of round 7 is $K_7 = (G(A + C - KC),\ G(B - D + KC))$, where KC is the round constant and (A, B, C, D) are the four 32-bit words of the master key after the rotations of the first six rounds. Since G is a permutation, each candidate for K_7 fixes $A + C$ and $B - D$ modulo 2^{32}; the remaining 64 bits, for instance A and B, are found by exhaustive search with trial encryptions. The retained candidates therefore cost $2^{64-a_{\mathrm{adv}}} \cdot 2^{64} = 2^{128-a_{\mathrm{adv}}}$ trial encryptions in total.

For the output mask $\Gamma L_6 = \texttt{0300000000000000}$, a one-sided Boolean dependency test is applied to the map $(x, K) \mapsto \Gamma L_6 \cdot F(x, K)$. For each of the 64 bits of K, a witness (x, K) is found for which flipping that bit changes the parity. Thus every last-round subkey bit is relevant to the Boolean function used in the attack, so the direct last-round tabulation adopted here enumerates all $g = 64$ subkey bits. This witness test establishes the dependency of the chosen attack statistic; it is not a universal lower bound on the key dimension of every possible attack formulation.

5.4 Complexity of the Key-Recovery Attack

The primary 6-round hull gives $c_{\mathrm{rms}} = 2^{-53.168}$ and hence the natural statistical data scale $c_{\mathrm{rms}}^{-2} = 2^{106.34}$.

For the success estimate, Selçuk's Gaussian approximation [28] gives, for an a_{adv}-bit advantage,

$$P_S \approx \Phi\Big(\sqrt{N}\,|c| - \Phi^{-1}\big(1 - 2^{-a_{\mathrm{adv}}-1}\big)\Big). \tag{17}$$

With the convention $c = 2\epsilon$ of Section 2.1, the factor $\sqrt{N}\,|c|$ equals the more familiar $2\sqrt{N}\,\epsilon$. Here an advantage of a_{adv} bits means that the correct 64-bit last-round subkey is placed among the best $2^{64-a_{\mathrm{adv}}}$ hypotheses.

Under the independent-round-key model, the fixed-key correlation of a hull that combines many trails of comparable magnitude is approximately normal with mean zero and variance c_{rms}^2 [14]; in the 6-round hull the lightest trail contributes only about 8% of c_{rms}^2. The magnitude $|c(K)|$ therefore falls below c_{rms} for most keys, with mean about $0.80\,c_{\mathrm{rms}}$, and this spread dominates the estimate: at $|c| = c_{\mathrm{rms}}$ every operating point of Table 10 succeeds with probability close to one, whereas averaging Eq. (17) over keys with $a_{\mathrm{adv}} = 20$ gives 73.1%, 80.8%, and 86.3% for $N = 2^{114}$, 2^{115}, and 2^{116}. Since N stays far below the codebook, the refinements for wrong-key randomization and for sampling without replacement [9, 12] are negligible here: at $N = 2^{115}$ the wrong-key correlation variance 2^{-128} is only 2^{-13} of the sampling variance $1/N$.

Table 10 Operating points of the 7-round key-recovery attack for an advantage of $a_{\mathrm{adv}} = 20$ bits, with the success probability at the root-mean-square correlation and averaged over keys.

data N	P_S at $\lvert c\rvert = c_{\mathrm{rms}}$	P_S, key average	master-key search	total time
2^{114}	$> 99.9\%$	73.1%	2^{108}	$2^{114.0}$
2^{115}	$> 99.9\%$	80.8%	2^{108}	$2^{115.0}$
2^{116}	$> 99.9\%$	86.3%	2^{108}	$2^{116.0}$

The time is dominated by processing the N texts. The transform costs $3 \cdot 2^{64} \cdot 64 \approx 2^{71.6}$ table operations, which in the memory-access convention of Lu et al. [20], where eight table accesses count as one encryption, corresponds to about $2^{68.6}$ encryptions, and the master-key search costs $2^{128-a_{\mathrm{adv}}} = 2^{108}$ trial encryptions. Counting each processed text as one unit, the attack at $N = 2^{115}$ therefore has data complexity about 2^{115} known plaintexts, time complexity about 2^{115} encryptions, and memory complexity about 2^{68} bytes for 2^{64} sixteen-byte counters, which rule out any overflow in the transform, with a success probability of about 81% on average over keys. Eight-byte counters would halve this to 2^{67} bytes, but under the model above the correct key then overflows with probability about 2.5%.

As a conservative check that does not rely on the hull enumeration, the expected squared correlation is at least that of the best 6-round trail alone, $2^{-110.05}$. With this lower bound in place of c_{rms}^2, the same key-averaged estimate for $a_{\mathrm{adv}} = 20$ reaches 82.6% at $N = 2^{119}$, so the attack remains far below exhaustive search even without any hull gain.

For SEED, differential cryptanalysis currently reaches further: the 9-round differential attack of Lu et al. [20] requires about 2^{124} chosen plaintexts, $2^{125.36}$ encryptions, and $2^{69.71}$ bytes of memory.

5.5 Limits of Straightforward Extensions

The same direct-tabulation model explains why the attack does not readily extend to eight rounds.

The 7-round hull with one appended round.

Appending one round to the primary 7-round hull, with $c_{\mathrm{rms}} = 2^{-63.391}$, gives an 8-round statistic of the same form as Eq. (16). Even with the full codebook, Eq. (17) averaged over keys places the correct last-round subkey in the better half of the hypotheses only with probability about 64%, so a one-bit advantage is the limit, and the subsequent master-key search keeps the total cost at the level of exhaustive search. Among the endpoint pairs examined, none changes this: the 1,416 trails of weight below 66.22 in the four 7-round patterns with the lowest floors form 616 endpoint pairs, and within this window the strongest pair exceeds the primary pair by only 0.04 bits.

Two-sided extension of the 6-round hull.

Appending one guessed round at each end of the 6-round hull exposes 128 outer-round key bits in the direct table, giving a 2^{128}-entry counter space. This route is already infeasible at the level of direct tabulation.

Free-prefix route.

A different possibility is to impose $\Gamma R_0 = 0$, so that a prefix round can be absorbed without guessing its round key. The natural 6-round pattern satisfying this constraint is $Z[-1, 2, 5]$, and its four active F-functions decompose into two reciprocal pairs exactly as in Eq. (13).

Corollary 5 *Every 6-round trail in pattern $Z[-1, 2, 5]$ has weight at least $2P_{\min} = 67.082$, and this value is attained by assigning the optimal iterative mask of Theorem 5 to all four active F-functions. The minimum weight of that pattern is therefore exactly* 67.082.

Proof The zeros at λ_{-1}, λ_2, and λ_5 enclose two pairs of active rounds, so Theorem 7 gives the lower bound; the upper bound is the stated construction. □

The value 67.082 lies about 12 bits above the lightest unrestricted 6-round trail found in Section 4.3. A complete 6-round floor probe confirms that $Z[-1, 2, 5]$ has the smallest surrogate floor among the free-prefix patterns, namely 60; the next two patterns $Z[-1, 3]$ and $Z[-1, 5]$ have floor 61, and the remaining resolved floors are 66, 69, 70, 72, 72, and 78. The free-prefix condition therefore raises the cost of the strongest currently identified 6-round structures by substantially more than the one round of key guessing it removes, although the exact optimum of every free-prefix pattern is not known.

Free-prefix route with seven rounds.

For the analogous route with a 7-round free-prefix distinguisher, the completed 7-round probe used in Theorem 11 proves every pattern with $\Gamma R_0 = 0$ UNSAT through cost 65. A deeper exploratory probe finds first SAT floors beginning at 72 for several resolved patterns, but four high-cost patterns do not reach a first SAT level within the computational budget. Consequently, no universal 72-bit lower bound or impossibility statement for 9-round linear cryptanalysis is claimed.

The evidence supports only a route-specific conclusion: the 7-round statistical scale is already close to the full codebook, the directly exposed outer key has 64 bits for the selected endpoint, and the hull gains are only 1.744–1.859 bits. Multiple linear cryptanalysis, multidimensional techniques, and alternative outer-round decompositions remain outside this analysis.

6 Conclusion

This paper has proposed divide-and-decide (D&D), a new hierarchical certified search framework, and applied it to the linear cryptanalysis of SEED. D&D divides the trail space at two levels, followed by one decision: it divides the trail space by zero pattern and prunes the patterns with the F-function spacing bound, divides each remaining pattern into layers of integer cost and enumerates the lowest layers with SAT, and decides each remaining layer by one SAT decision model on its excess. Its pattern level rests on the spacing theorem for trivial rounds and on the count of admissible zero patterns; its decision level rests on the decomposition of the exact weight into an integer surrogate cost and a nonnegative excess.

Applied to SEED, the framework determines the exact minimum weight of a linear trail through a single F-function, 9.415, and the exact optima of 3- to 7-round linear trails, 18.830, 33.541, 43.472, 55.027, and 65.220. KISA's 4-round evaluation is optimal, whereas KISA's 5-, 6-, and 7-round estimates are improved by about 6.84, 12.05, and 1.86 bits. FSB gives a lower bound on the trail weight for every number of rounds

without multi-round search, 112.980 at 16 rounds; these single-trail bounds do not bound linear hulls.

The exact 6-round optimum already yields a 6-round distinguisher within the codebook, and linear hulls for selected endpoints lower the 6- and 7-round weights further to 53.168 and 63.391, which brings the 7-round distinguisher within the codebook as well. On the 6-round hull, the first linear key-recovery attack on SEED recovers the full key of 7-round SEED with data complexity 2^{115} known plaintexts, time complexity 2^{115} encryptions, and memory complexity 2^{68} bytes, with an estimated success probability of about 81% on average over keys. These figures are analytic estimates, and the attack has not been executed. Appending one round to the 7-round hull leaves only one bit of advantage even with the full codebook. Although far from practical, the attack shows how far single-approximation linear key recovery reaches on SEED.

The search is tailored to SEED, but its two principal ingredients are not. The spacing argument behind FSB applies to any Feistel cipher whose round function is a permutation, once its single-round and reciprocal-pair optima are known, and the decision level applies to trail searches whose exact weight splits into an integer cost and a nonnegative excess that the solver can encode by component classes. Applying D&D to other ciphers that combine S-boxes with modular addition is a natural direction for future work.

Statements and Declarations

- **Funding.** This work is partially supported by the National Cyber Security–National Science and Technology Major Project under Grant No. 2026ZD1500800, the Natural Science Foundation of Henan under Grant No. 242300421173, the National Cryptologic Science Fund of China under Grant No. 2025NCSF02044, and the National Natural Science Foundation of China under Grant No. 61802437.

- **Competing interests.** The authors have no competing interests to declare that are relevant to the content of this article.
- **Data availability.** The numerical results supporting the findings of this study are reported in the article; the search code and the computation logs are available from the corresponding author on reasonable request.
- **Code availability.** The search code and computation logs will be made publicly available upon publication.
- **Author contributions.** Haonan Hong conceived and implemented the search framework, carried out the analysis and the experiments, and wrote the manuscript. Kai Zhang supervised the research and revised the manuscript. Both authors read and approved the final manuscript.

Appendix A Reproducibility Notes

The SAT implementation uses PySAT [26] with CaDiCaL 1.9.5. It was checked against the RFC 4269 SEED test vectors, independently generated S-box LATs, exhaustive 5-bit modular-addition correlations, and an independent 3-round trail-pairing calculation. The pattern-completeness check of Section 5.1 is a separate probe with both endpoints fixed, run with the endpoints and windows of Table 9. For every reported

layer, the logs record the proof-critical counts and whether the layer ends by UNSAT or by a resource bound. The reported environment is an Intel Core i9-14900HX with 31.7 GB RAM under Windows 11, Python 3.12.10, PySAT 1.9.dev15, and CaDiCaL 1.9.5. Wall-clock time is treated only as an implementation characteristic; all proof claims depend on completed search regions or analytic arguments. Solver timeouts are never promoted to negative results; in particular, the exact 7-round optimum of Theorem 11 depends only on the completely processed region cost ≤ 60 and on decisions that ended in UNSAT.

The decisions of Proposition 2 use the same SAT models with the excess constraints of Eq. (7), $\kappa = 40$, and the thresholds 43.4725, 55.0275, and 65.2205. Before use, the decision code was checked against complete enumeration. The cost-38 layer of the 5-round pattern $Z[0, 4]$ was enumerated with the exact weights of all 14,400 assignments, and decisions with thresholds 43.4725, 44, 44.5, 45, and 46, run until UNSAT with every found trail blocked, recovered exactly the 8, 88, 224, 584, and 1,440 trails below each threshold; the only other assignments returned were 16 false positives at threshold 44.5, all rejected by exact evaluation. On the reported machine, the 3 decisions at 5 rounds took 114 s in total, the 90 decisions at 6 rounds 4.1 h, and the 20 decisions at 7 rounds 2.5 h, the hardest single decision about 36 min. The enumeration depth d was not tuned: the 5-round layers of cost 39 and 40 were enumerated before the decision level was introduced, and deciding them instead would leave the certificates unchanged. The FSB timings of Section 4.2 were measured in single-threaded CPython 3.12 on an x86-64 machine.

Table 11 lists the sizes of representative models. The base model contains the S-box feasibility constraints, the carry-automaton encoding of the modular additions, and the Feistel mask relations; the cost counter is the totalizer over the cost literals, built up to the largest cost used; and a decision adds, for every S-box, the one-hot recognition of its input and output masks, one indicator per admitted LAT class, and a weighted totalizer bounded by the allowed excess. The excess encoding has 17 to 29 times as many clauses as the base model, which is why it is used only for the residual layers.

True S-box weights are derived from exact integer LAT entries and modular addition uses exact carry weights. Logarithmic totals are stored as double-precision values for candidate comparison, but the lower-bound proofs do not depend on those floating-point totals. The double-precision error of a weight, a sum of at most 60 logarithms, is below 10^{-10}, far below the margins between the exact incumbents 43.471973, 55.027188, and 65.219834 and their thresholds 43.4725, 55.0275, and 65.2205, which are $5.3 \cdot 10^{-4}$, $3.1 \cdot 10^{-4}$, and $6.7 \cdot 10^{-4}$. No independently checked DRAT/LRAT certificates are claimed for the present runs.

Table 11 Sizes of representative SAT models: the base model, the cost counter, and the excess encoding added by a decision.

model	part	S-boxes	variables	clauses
reciprocal pair	base	24	1,240	103,589
	cost counter, cost ≤ 36		1,560	10,623
5 rounds, $Z[0,4]$	base	36	2,436	157,052
	cost counter, cost ≤ 43		2,436	18,816
	excess encoding, layer 41		80,008	2,751,882
6 rounds, $Z[0,4]$	base	48	3,120	209,103
	cost counter, cost ≤ 55		3,443	31,202
	excess encoding, layer 51		162,067	5,927,709
7 rounds, $Z[1,5]$	base	60	3,804	261,154
	cost counter, cost ≤ 65		4,481	45,418
	excess encoding, layer 61		203,758	7,558,303

References

[1] ISO/IEC. *ISO/IEC 18033-3:2010, Information technology—Security techniques—Encryption algorithms—Part 3: Block ciphers*. International Organization for Standardization, 2010. https://www.iso.org/standard/54531.html

[2] J. Park, S. Lee, J. Kim, J. Lee. Use of the SEED Encryption Algorithm in Cryptographic Message Syntax (CMS). RFC 4010, IETF, 2005. https://doi.org/10.17487/RFC4010

[3] H. J. Lee, J. H. Yoon, J. I. Lee. Addition of SEED Cipher Suites to Transport Layer Security (TLS). RFC 4162, IETF, 2005. https://doi.org/10.17487/RFC4162

[4] H. J. Lee, J. H. Yoon, S. L. Lee, J. I. Lee. The SEED Cipher Algorithm and Its Use with IPsec. RFC 4196, IETF, 2005. https://doi.org/10.17487/RFC4196

[5] S. Yoon, J. Kim, H. Park, H. Jeong, Y. Won. The SEED Cipher Algorithm and Its Use with the Secure Real-Time Transport Protocol (SRTP). RFC 5669, IETF, 2010. https://doi.org/10.17487/RFC5669

[6] CRYPTREC. *Analysis of SEED*. CRYPTREC-EX-1041-2001, 2003.

[7] H. Yanami, T. Shimoyama. Differential Cryptanalysis of a Reduced-Round SEED. *Security in Communication Networks*, 186–198, 2002. https://doi.org/10.1007/3-540-36413-7_14

[8] J. Sung. Differential Cryptanalysis of Eight-Round SEED. *Information Processing Letters* 111(10):474–478, 2011. https://doi.org/10.1016/j.ipl.2011.02.004

[9] T. Ashur, T. Beyne, V. Rijmen. Revisiting the Wrong-Key-Randomization Hypothesis. *Journal of Cryptology* 33(2):567–594, 2020. https://doi.org/10.1007/s00145-020-09343-2

[10] A. Biryukov, V. Velichkov, Y. Le Corre. Automatic Search for the Best Trails in ARX: Application to Block Cipher Speck. In *Fast Software Encryption (FSE 2016)*, LNCS 9783, pp. 289–310, Springer, 2016. https://doi.org/10.1007/978-3-662-52993-5_15

[11] O. Bailleux, Y. Boufkhad. Efficient CNF Encoding of Boolean Cardinality Constraints. *CP 2003*, LNCS 2833, 108–122. https://doi.org/10.1007/978-3-540-45193-8_8

[12] A. Bogdanov, E. Tischhauser. On the Wrong Key Randomisation and Key Equivalence Hypotheses in Matsui's Algorithm 2. In *FSE 2013*, LNCS 8424, pp. 19–38. Springer, 2014. https://doi.org/10.1007/978-3-662-43933-3_2

[13] B. Collard, F.-X. Standaert, J.-J. Quisquater. Improving the Time Complexity of Matsui's Linear Cryptanalysis. *ICISC 2007*, 77–88. https://doi.org/10.1007/978-3-540-76788-6_7

[14] J. Daemen, V. Rijmen. Probability Distributions of Correlation and Differentials in Block Ciphers. *Journal of Mathematical Cryptology* 1(3):221–242, 2007. Preprint: https://eprint.iacr.org/2005/212

[15] M. Huang, L. Wang. Automatic Search for the Linear (Hull) Characteristics of ARX Ciphers: Applied to SPECK, SPARX, Chaskey and CHAM-64. Cryptology ePrint Archive, Paper 2019/1319, 2019. https://eprint.iacr.org/2019/1319

[16] Korea Information Security Agency. *SEED 128 Algorithm Self Evaluation*. KISA, 1999.

[17] L. R. Knudsen. Practically Secure Feistel Ciphers. In *Fast Software Encryption (FSE 1993)*, LNCS 809, pp. 211–221, Springer, 1994. https://doi.org/10.1007/3-540-58108-1_26

[18] Y. Liu, H. Liang, M. Li, L. Huang, K. Hu, C. Yang, M. Wang. STP Models of Optimal Differential and Linear Trail for S-box Based Ciphers. *Science China Information Sciences* 64(5), 2021. https://doi.org/10.1007/s11432-018-9772-0; also IACR ePrint 2019/025.

[19] Y. Liu, Q. Wang, V. Rijmen. Automatic Search of Linear Trails in ARX with Applications to SPECK and Chaskey. In *Applied Cryptography and Network Security (ACNS 2016)*, LNCS 9696, pp. 485–499, Springer, 2016. https://doi.org/10.1007/978-3-319-39555-5_26

[20] J. Lu, W.-S. Yap, M. Henricksen, S.-H. Heng. Differential attack on nine rounds of the SEED block cipher. *Information Processing Letters* 114(3):116–123, 2014. https://doi.org/10.1016/j.ipl.2013.11.004

[21] R. Martins, S. Joshi, V. Manquinho, I. Lynce. Incremental Cardinality Constraints for MaxSAT. *CP 2014*, LNCS 8656, 531–548. https://doi.org/10.1007/978-3-319-10428-7_39

[22] M. Matsui. Linear Cryptanalysis Method for DES Cipher. *EUROCRYPT 1993*, LNCS 765, 386–397. https://doi.org/10.1007/3-540-48285-7_33

[23] M. Matsui. On Correlation Between the Order of S-boxes and the Strength of DES. *EUROCRYPT 1994*, LNCS 950, 366–375. https://doi.org/10.1007/BFb0053451

[24] N. Mouha, Q. Wang, D. Gu, B. Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. *Inscrypt 2011*, LNCS 7537, 57–76. https://doi.org/10.1007/978-3-642-34704-7_5

[25] K. Nyberg. Linear Approximation of Block Ciphers. *EUROCRYPT 1994*, LNCS 950, 439–444. https://doi.org/10.1007/BFb0053460

[26] A. Ignatiev, A. Morgado, J. Marques-Silva. PySAT: A Python Toolkit for Prototyping with SAT Oracles. *SAT 2018*, LNCS 10929, 428–437. https://doi.org/10.1007/978-3-319-94144-8_26

[27] H. J. Lee, S. J. Lee, J. H. Yoon, D. H. Cheon, J. I. Lee. The SEED Encryption Algorithm. RFC 4269, IETF, 2005. https://doi.org/10.17487/RFC4269

[28] A. A. Selçuk. On Probability of Success in Linear and Differential Cryptanalysis. *Journal of Cryptology* 21(1):131–147, 2008. https://doi.org/10.1007/s00145-007-9013-7

[29] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, L. Song. Automatic Security Evaluation and (Related-key) Differential Characteristic Search. *ASIACRYPT 2014*, LNCS 8873, 158–178. https://doi.org/10.1007/978-3-662-45611-8_9

[30] L. Sun, W. Wang, M. Wang. Accelerating the Search of Differential and Linear Characteristics with the SAT Method. *IACR Trans. Symmetric Cryptol.* 2021(1):269–315. https://doi.org/10.46586/tosc.v2021.i1.269-315

[31] K. Taka, K. Sakamoto, R. Ito, R. Shiba, S. Utsumi, T. Isobe. Divide-and-Conquer SAT for Exploring Optimal Differential and Linear Characteristics and Its Applications. *IACR Trans. Symmetric Cryptol.* 2025(3):516–576. https://doi.org/10.46586/TOSC.V2025.I3.516-576

[32] Telecommunications Technology Association. *128-bit Symmetric Block Cipher (SEED)*. TTAS.KO-12.0004, 1999.

[33] J. Wallén. Linear Approximations of Addition Modulo 2^n. *FSE 2003*, LNCS 2887, 261–273.

[34] W. Wu, H. Ma, D. Feng. Security on Korean Encryption Standard. *Acta Electronica Sinica* 31(4):585–588, 2003. https://www.ejournal.org.cn/en/article/114696782/