Multidimensional Hill Cipher Substitution-Permutation Network with AES S-Box and Argon2id Key Derivation

Porter E. Coggins *

Posted Date: 29 April 2026

doi: 10.20944/preprints202604.2038.v1

Keywords: AES; block cipher; Hill cipher; substitution-permutation network

 doi:10.20944/preprints202604.2038.v1

Article

Multidimensional Hill Cipher Substitution-Permutation Network with AES S-Box and Argon2id Key Derivation

Porter E. Coggins

Bemidji State University, Bemidji, MN, USA; porter.coggins@bemidjistate.edu

Abstract

The Hill cipher has historically lacked the confusion and diffusion properties required for modern cryptographic use. This paper presents the Multidimensional Hill Substitution-Permutation Network (MD-Hill-SPN), a 128-bit, 12-round block cipher combining three elements: (1) a hierarchical matrix diffusion layer operating at 4×4, 8×8, and 16×16 byte scales over GF(2^8); (2) two AES S-box substitution layers per round; and (3) Argon2id memory-hard key derivation. Metric sessions used a SHA-256 domain-separator surrogate for Argon2id for computational tractability; Argon2id is the specified production KDF. Two independent runs of the full metric suite yield: (a) full plaintext avalanche from round 1 (mean 63.97–64.67 of 128 bits, ideal 64); (b) the differential-probability sampling floor of 2×10^{-5} reached at round 4 (50,000 of 50,000 output differences distinct, both sessions); (c) algebraic-degree lower-bound saturation at the maximum observable value from round 1; (d) linear bias indistinguishable from random (combined exceedance 4.40%, below the 4.55% noise floor); and (e) branch numbers at the Singleton (MDS) bound for every tier (B = 5 for 4×4, B = 9 for 8×8, B = 17 for 16×16), computed exhaustively over weight-1 inputs. MD-Hill-SPN therefore moves beyond theoretical construction to empirically validated confusion and diffusion properties stronger than prior Hill-cipher variants.

Keywords: AES; block cipher; Hill cipher; substitution-permutation network

MSC: 68W32 algorithms on strings; 68M25 computer security; 68U35 computing methodologies for information systems 94A05 communication theory; 94A60 cryptography

1. Introduction

In *Cryptography in an Algebraic Alphabet*, Lester Hill [1] introduced a linear transformation mapping plaintext to ciphertext that has since become known as the Hill cipher. The Hill cipher is a polygraphic block cipher whose encryption and decryption processes are naturally expressed in terms of matrix algebra. Specifically, encryption is performed by multiplying plaintext vectors by a key matrix, while decryption requires multiplication by the modular inverse of that key matrix.

In the classical formulation by Hill, plaintext blocks reside in $(\mathbb{Z}/26\mathbb{Z})^n$, which forms a free module, but not a vector space, over the ring $(\mathbb{Z}/26\mathbb{Z})^n$. Letters of the English alphabet are encoded modulo 26 without case distinction. Since 26 = 2 × 13 is composite, the ring $(\mathbb{Z}/26\mathbb{Z})^n$contains zero divisors, and a key matrix K is invertible if and only if gcd(det K, 26) = 1. Given a key matrix $K = (a_{ij})$ with entries in $(\mathbb{Z}/26\mathbb{Z})^n$, encryption of a plaintext vector x is defined component-wise by $c_i = \Sigma_{j=0}^{n-1} a_{ij} x_j \pmod{26}$

Although marginally stronger than a simple monoalphabetic substitution cipher, the Hill cipher is primarily of theoretical rather than practical significance due to its well-documented cryptographic weaknesses. Most notably, it is vulnerable to known-plaintext attacks [2–5,9], as the encryption process consists entirely of a linear transformation over a modular ring. Consequently, an adversary can recover the key matrix once a sufficient number of plaintext–ciphertext pairs are available.

 doi:10.20944/preprints202604.2038.v1

The Hill cipher does exhibit a limited form of diffusion as defined by Shannon [6] (pp. 708–709), achieved by operating on blocks of symbols rather than individual characters. However, this diffusion is confined to a single linear layer, whose effectiveness is bounded by the dimension of the key matrix and does not compound across multiple rounds. This stands in contrast to modern block cipher designs, such as substitution–permutation networks, where diffusion accumulates iteratively through alternating nonlinear and linear layers [4,7–9]. Moreover, the cipher provides little confusion in Shannon's sense [6] (pp. 709–710): the algebraic relationship among key, plaintext, and ciphertext remains linear, and the scheme lacks any intrinsic nonlinear transformation capable of resisting linear or algebraic cryptanalysis.

Prior Hill cipher modifications fall broadly into three categories, addressed in Sections 2.1, 2.2, and 2.3 respectively: affine variants, dynamic-key variants, and key-element variants.

Despite numerous attempts to enhance the security of the Hill cipher through structural modifications [3–5,10–18,20,21], Saeednia's paper [19] serves as the primary catalyst that renewed scholarly interest in Hill cipher improvements. Nevertheless, explicit security metrics such as the avalanche effect, diffusion measurements, and resistance under formal threat models have been reported only incompletely and never as a combined metric suite across avalanche, differential, linear, algebraic-degree, and branch-number probes simultaneously (see Table 1 for a systematic summary), motivating the present work's demonstration that a multidimensional Hill cipher variant can be constructed to satisfy minimum modern cryptanalytic evaluation criteria despite modern security metrics well-known in the literature since at least the late 1990s [22].

The main contributions of this paper are as follows: (a) A multidimensional matrix diffusion hierarchy (four 4×4, two 8×8, and one 16×16 matrix over GF(2^8)) achieving branch numbers that meet or exceed the MDS bound at every dimensional tier. (b) A complete 12-round SPN round function combining hierarchical linear diffusion with two AES S-box nonlinear substitution layers per round. (c) Memory-hard key derivation via Argon2id (t=3, m=65,536 KiB, p=2). (d) Empirical security validation across plaintext avalanche, key avalanche, differential distribution, linear-bias, algebraic-degree, and branch-number metrics across two independent sessions.

The remainder of this paper is organized as follows. Section 2 provides background on prior work. Section 3 specifies the round function and full cipher. Section 4 reports the empirical security analysis. Section 5 concludes and provides areas for future research consideration.

Table 1. Summary of prior Hill Cipher variants reviewed in Section 2 and identification of the literature gap.

Category	Author(s) [Ref]	Core Modification	Nonlinear Substitution	Multi-Round / SPN Structure	Formal Security Metrics Reported
2.1 Affine Variants					
Affine	Valizadeh [10]	Additive vector injection; counters zero-plaintext attack	✗	✗	✗
Affine	Toorani et al. [11]	Key-dependent additive offset; targets KPA and zero-plaintext attack	✗	✗	✗

 doi:10.20944/preprints202604.2038.v1

Affine	Nordin et al. [12]	Affine augmentation to break simple linear dependencies	✗	✗	✗
2.2 Dynamic Key Variants					
Dynamic Key	Ismail, Amin & Diab [3]	Per-block dynamic key derived from auxiliary parameters	✗	✗	✗
Dynamic Key	Ravan & Nigavekar [13]	Per-block key update via deterministic scheduling	✗	✗	✗
Dynamic Key	Bahtiar, Widodo, & Puspita [15]	Per-block key generated via random numbers	✗	✗	✗
Dynamic Key	Jin, Wu, Ouyang & Li [16]	Dynamic key generation for cross-block diffusion	✗	✗	✗
Dynamic Key	Coggins & Glatzer [17]	Enigma-inspired matrix rotation; invertibility-preserving key variation	✗	✗	✗
Dynamic Key	Coggins [18]	Systematic Enigma-style key scheduling; two-variation treatment	✗	✗	✗
Dynamic Key	Putera, Siahaan & Rahim [20]	Genetic algorithm search for invertible matrices (det = 1)	✗	✗	✗
Dynamic Key (SPN-adjacent)	Paragas, Sison & Medina [21]	S-boxes + CBC + XOR + circular shifts; approaches SPN structure	~	~	~
2.3 Key Element Variants					
Key Element	Maxrizal [14]	Complex-number modular generalisation of key	✗	✗	✗

 doi:10.20944/preprints202604.2038.v1

		matrix and plaintext space			
2.4 Gap in the Literature — Present Work					
SPN (present work)	**Coggins [present]**	**Multidimensional-Hill-SPN: 4×4 / 8×8 / 16×16 GF(2^8) matrices;** two AES S-box layers per round; Argon2id KDF; 12-round 128-bit block cipher	✓✓	✓✓	✓✓ 5 metrics, 2 sessions
✓ = present and implemented; ~ = partially present (no formal round function / no iterated SPN structure); ✗ = absent. Nonlinear Substitution: presence of a cryptographically analysed nonlinear S-box component. Multi-Round / SPN Structure: formally defined iterated round function with alternating substitution and diffusion layers. Formal Security Metrics: at least one of avalanche, differential, linear-bias, algebraic-degree, or branch-number metric reported.					

Papers are grouped by modification category. The final three columns identify whether each work introduces (a) a nonlinear substitution component, (b) a formally defined multi-round / SPN structure, and (c) formal cryptographic security metrics. The pattern of absences across all three criteria motivates the present work.

2. Prior Work

Since Saeednia's seminal paper in 2000, the Hill cipher has attracted substantial scholarly attention; several hundred works addressing improvements and enhancements are indexed in Google Scholar and Scopus (as of April 2026). have been published. Given the breadth of the Hill cipher literature, the survey of prior work is necessarily selective; the author apologizes for omissions. Prior work is therefore organized into three representative groups covering the literature from 2000 to the present.

2.1. Affine Variants

Various affine extensions of the Hill cipher introduced an additive component that partially obscured the underlying linear structure. Valizadeh, Toorani et al., and Nordin et al. [10–12], among others, introduced affine extensions that not only reduced linearity but also provided resistance to the zero-plaintext attack when the affine component was appropriately chosen. Nonetheless, other non-affine variants have been shown to remain vulnerable to trivial input patterns, including zero-plaintext attacks that continue to leak key material.

Valizadeh [10] proposed an affine extension of the classical Hill cipher by incorporating an additive vector into the linear transformation. This modification aimed to obscure the direct linear relationship between plaintext and ciphertext and to specifically counter zero-plaintext attacks. When the affine component is carefully chosen, the scheme prevents trivial recovery of key material from all-zero inputs. However, the overall structure remains predominantly linear and offers limited resistance beyond basic attack models.

Toorani, et al. [11] introduced affine Hill cipher constructions that combined matrix multiplication with an additive offset derived from key-dependent parameters. Their approach explicitly targeted known-plaintext and zero-plaintext attacks by ensuring that encryption of trivial inputs does not reveal direct information about the key matrix. While this improved resistance to

 doi:10.20944/preprints202604.2038.v1

some classical attacks, the cipher still lacks nonlinear confusion. As a result, security improvements are incremental rather than transformational in a modern encryption scheme.

Nordin, et al. [12] investigated affine variants of the Hill cipher in which an additive component is integrated to break simple linear dependencies. Their work showed that affine augmentation can successfully neutralize zero-plaintext attacks that plague the classical Hill cipher. At the same time, the encryption process remained a single affine transformation over a modular ring. Consequently, the scheme continued to inherit many of the analytical weaknesses associated with linear ciphers. Where Valizadeh, Toorani et al., and Nordin et al. proposed affine extensions to static key matrices, there has been work on creating dynamic key matrices.

2.2. Dynamic Key Variants

Ismail, Amin, and Diab [3] proposed a Hill cipher variant in which the traditionally static key matrix is replaced by a dynamically changing key derived from auxiliary parameters or prior encryption state. The intent was to prevent attackers from exploiting repeated use of a single linear transformation under known-plaintext attacks. While the method increases variability in the key material, the encryption process itself remains a linear matrix multiplication modulo n. As a result, the scheme offered heuristic improvement rather than a fundamental structural change.

Ravan and Nigavekar [13] introduced a dynamic-key Hill cipher where the encryption matrix is updated on a per-block basis using deterministic scheduling rules. This approach expands the effective key space and complicates direct recovery of a single fixed key matrix from plaintext–ciphertext pairs. However, once the update mechanism is known, the cipher remains analyzable using linear techniques. Consequently, the security gains are limited and primarily empirical.

A central contribution of the work by Bahtiar et al. [15] is the introduction of an automated mechanism for generating valid Hill cipher key matrices, removing the need for users to manually construct invertible matrices. By employing a randomized generation process combined with determinant evaluation and modular inverse checks, the method guarantees that each produced key matrix is mathematically sound for encryption and decryption. The authors further quantify the resulting key space for the 2×2 case as key space as 95^4 (= 81,450,625) yielding over 81 million possible keys when operating modulo 95. This sizable key space, while not intended to provide modern cryptographic strength, substantially improves resistance to trivial brute-force attacks compared to ad hoc or fixed-key Hill cipher implementations. Bahtiar et al. [15] is classified as a dynamic-key variant because key matrices are regenerated algorithmically for each session, but it is not SPN-adjacent since encryption remains a single linear Hill transformation with no nonlinear substitution layer or iterated round function.

Jin, Wu, Ouyang, and Li [16] investigated dynamic key generation mechanisms designed to enhance diffusion across plaintext blocks in Hill-style encryption. Their scheme modified the key matrix between rounds or blocks to reduce straightforward algebraic attacks. Although this increased resistance to simple cryptanalysis, the underlying transformation remained linear over the chosen modulus. The work thus improves robustness without eliminating the cipher's fundamental weaknesses.

Coggins and Glatzer [17] focused on a dynamic-key-like Hill cipher construction using the German Enigma Encoder as a model for rotating key matrix values in a similar way that the German Enigma Encoder [7] with careful attention to algebraic correctness and invertibility conditions. Their analysis shows that controlled key variation can delay key recovery under known-plaintext assumptions while remaining practical to implement. At the same time, the authors acknowledge that dynamic keys alone do not introduce nonlinear confusion. The work frames such schemes as incremental improvements rather than modern secure ciphers.

Coggins [18] extended earlier dynamic-key Hill cipher research by providing a systematic treatment of key scheduling through two variations that include matrix element rotations along the lines of the German Enigma Encoding Machine, modular arithmetic, and correctness constraints. The paper emphasizes transparency in design and highlights how dynamic updates interact with matrix

 doi:10.20944/preprints202604.2038.v1

invertibility. Importantly, it recognizes that varying the key does not overcome the inherent linearity of Hill encryption. However, it still is bound by modular arithmetic within a fixed alphabet to number assignment rather than operating at the bit or byte level. Further, the scheme is not a SPN model of encryption. Dynamic-key variants are therefore stated as instructional and exploratory rather than cryptographically strong.

Putera, Siahaan, and Rahim [20] proposed a dynamic-key Hill cipher scheme in which genetic algorithms are used to efficiently search for invertible key matrices with determinant equal to one. Their contribution focuses on optimizing the key-generation process by replacing manual or brute-force selection with evolutionary search techniques, thereby reducing computational time. The encryption and decryption processes themselves remain unchanged from the classical Hill cipher, relying on a single linear transformation. As a result, the work improves key selection efficiency but does not address the fundamental cryptanalytic weaknesses of linear Hill encryption.

Paragas, Sison, and Medina [21] came the closest to a modern encryption scheme in the spirit of both Shannon [6] and Saeednia [19] by introducing a modified Hill cipher variant that incorporates substitution boxes, cipher block chaining, XOR operations, and circular shifts to approximate a modern substitution–permutation network (SPN) structure. The design introduces nonlinearity and inter-block dependency, yielding improved avalanche and statistical randomness compared to classical Hill cipher constructions. However, the Paragas et al. cipher lacks a clearly defined round function with iterated substitution and diffusion layers, and the modified S-box is static and not integrated into a rigorously analyzed permutation structure. Consequently, while the scheme moves conceptually toward an SPN-like design, it falls short of a modern SPN construction with provable diffusion accumulation and resistance under contemporary cryptanalytic models.

2.3. Key Element Variants

The primary representative example of key-element varieties is Maxrizal [14] who extended the classical Hill cipher by generalizing the key matrix and plaintext space to complex numbers modulo an integer, while preserving the standard Hill cipher encryption and decryption structure. The paper demonstrated that determinant and inverse computations can be carried out consistently in the complex modular setting, yielding ciphertexts that appear more randomized than in the integer-only formulation. However, the extension does not introduce nonlinearity, iteration, or round-based structure, and the encryption remains a single linear transformation over an enlarged algebraic domain. As a result, the scheme represents an algebraic generalization of Hill cipher mathematics rather than a step toward a modern block-cipher or SPN construction.

2.4. The Gap in the Literature Identified

This paper presents a cryptographically secure *Multidimensional Hill Substitution – Permutation Network.* The literature surveyed above reveals a consistent gap: prior Hill cipher variants operate exclusively on matrices of a single fixed dimension and lack the combination of a formal substitution-permutation network structure, a memory-hardened key derivation function, an explicit salt value, and comprehensive metric evaluation. No prior work has reported all of the following metrics simultaneously: plaintext avalanche, key avalanche, differential distribution, linear-bias probe, algebraic degree, and branch number. This gap motivates the present work, which demonstrates that a multidimensional Hill cipher constructed along the lines of Coggins [18] can be designed to satisfy minimum modern cryptographic evaluation criteria. This paper does not report on hardware-dependent analyses. Table 1 summarizes the prior research and the present study.

3. Round Function and Simplified Cipher Scheme

3.1. Simplified Multidimensional–Hill–SPN Encryption Scheme

The simplified *Multidimensional–Hill–SPN* (MD-Hill-SPN) basic round function scheme is indicated in Figure 1.

 doi:10.20944/preprints202604.2038.v1

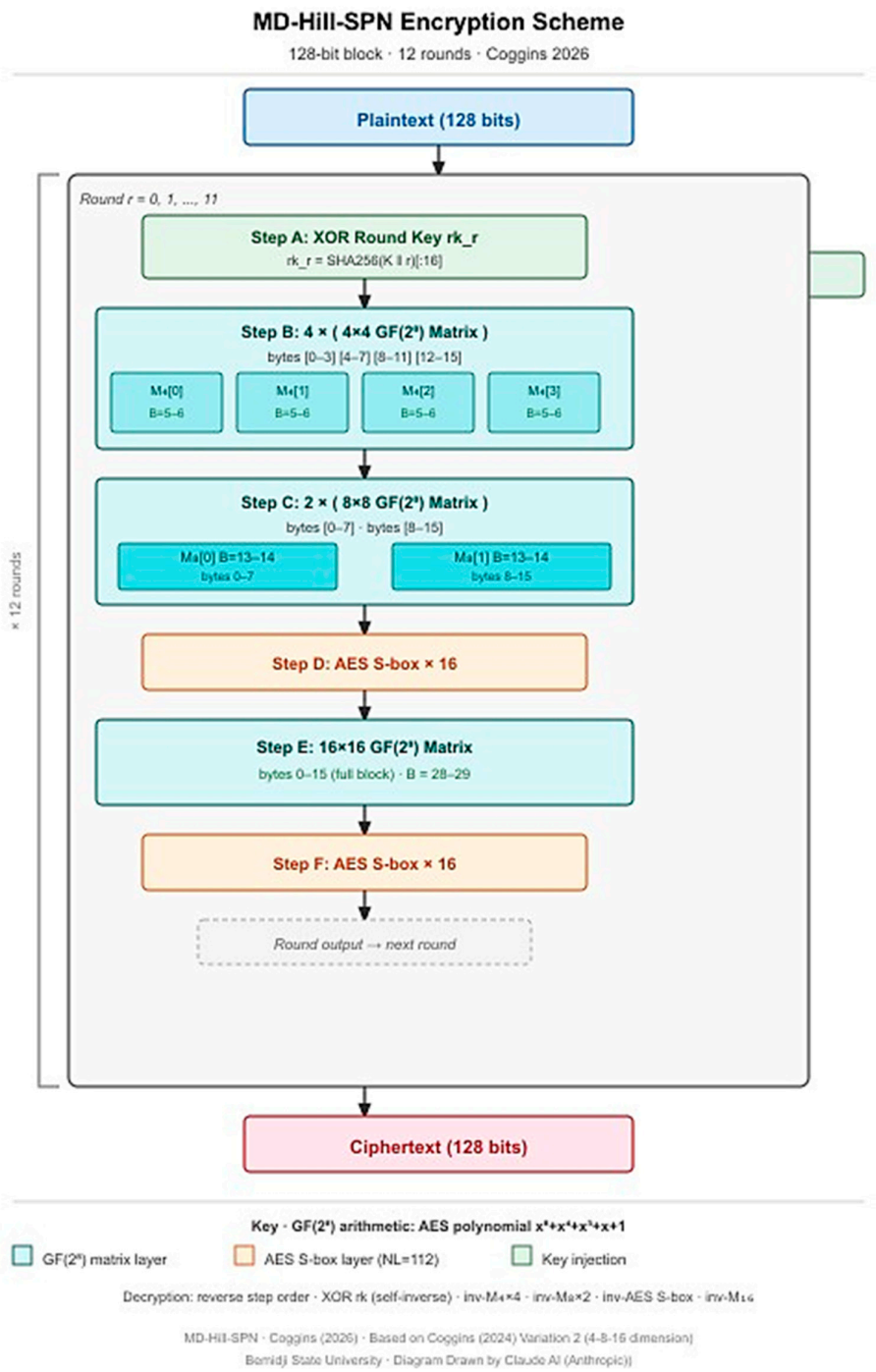

Figure 1. This figure represents the *Multidimensional Hill SPN* (MD-Hill-SPN) simplified scheme.

3.2. Round Structure and High-Level Flow

Each MD-Hill-SPN round operates on a 128-bit internal state of sixteen GF(2^8) bytes and applies a fixed sequence of six transformations: round-key injection (Step A); intra-group diffusion by four parallel 4×4 matrices (Step B); inter-group diffusion by two 8×8 matrices (Step C); the first AES S-box substitution layer S_1 (Step D); full-state diffusion by a single 16×16 matrix (Step E); and the second AES S-box substitution layer S_2 (Step F). The diffusion layers progress from local to full-state mixing; the two substitution layers frame the full-block 16×16 matrix, so that nonlinear confusion in the sense

 doi:10.20944/preprints202604.2038.v1

of Shannon [6] is introduced both before and after the final diffusion stage within every round. The full cipher iterates this round function R = 12 times..

3.3. Step A: Round Key Injection

Step A introduces the round key into the state through bytewise XOR. Each round key is derived independently from the master key and serves to break structural symmetry between rounds. This operation ensures that all subsequent transformations are key-dependent while remaining computationally simple and reversible. Placing key injection at the beginning of each round aligns the construction with standard SPN design principles. SHA-256 domain-separator stub "MDHILLRK" is used for metric runs (for computation speed), although Argon2id is the production KDF to distinguish design specifications from explicitly reported metric implementation.

3.4. Step B: Intra-Group Diffusion Using Parallel 4×4 Matrices

The internal cipher state is represented as an ordered collection of sixteen bytes, written as s = $(s_0, s_1, \ldots, s_{15})$. Each element s_i is interpreted as an element of the finite field GF(2^8), the unique field containing 256 elements. Arithmetic on the state is therefore performed using field addition and multiplication in GF(2^8). This representation corresponds to a 128-bit data block composed of sixteen field elements.

In Step B, the state s is partitioned into four disjoint four-byte sub-vectors: $s_0 = (s_0, s_1, s_2, s_3)$, $s_1 = (s_4, s_5, s_6, s_7)$, $s_2 = (s_8, s_9, s_{10}, s_{11})$, and $s_3 = (s_{12}, s_{13}, s_{14}, s_{15})$. Each sub-vector is transformed independently by multiplication with a key-dependent four-by-four matrix $M_i(4)$. Each $M_i(4)$ is an element of the general linear group GL(4, GF(2^8)), meaning $M_i(4)$ is invertible over GF(2^8). The resulting output sub-vectors are $y_i = M_i(4)$ multiplied by s_i, with multiplication defined as matrix–vector multiplication over GF(2^8).

The role of the matrices $M_i(4)$ is to provide strong local diffusion. For any nonzero input difference introduced in a single element of s_i, the output y_i contains differences in all four positions after a single application of $M_i(4)$. Each $M_i(4)$ is derived deterministically from the master key and retained only if it satisfies invertibility and minimum diffusion constraints. This construction generalizes classical Hill-cipher diffusion into a parallel, byte-oriented setting while preserving algebraic correctness and reversibility. SHA-256 domain-separator stub "MDHILL_4" is used for metric runs.

3.5. Step C: Inter-Group Diffusion Using 8×8 Matrices

In Step C, diffusion is expanded beyond local four-byte neighborhoods. The state produced by Step B is regrouped into two eight-byte sub-vectors: $t_0 = (s_0, s_1, \ldots, s_7)$ and $t_1 = (s_8, s_9, \ldots, s_{15})$. Each sub-vector therefore combines two adjacent four-byte groups from the previous step.

Each eight-byte sub-vector t_j is transformed using a key-dependent eight-by-eight matrix $M_j(8)$, where $M_j(8)$ belongs to the general linear group GL(8, GF(2^8)). As an element of GL(8, GF(2^8)), each $M_j(8)$ is invertible over the field GF(2^8), ensuring that the transformation is reversible. The transformed outputs are $v_j = M_j(8)$ multiplied by t_j, with all arithmetic performed in GF(2^8).

This stage couples pairs of previously independent four-byte groups, causing differences introduced in any one group to propagate across an eight-byte region. The increased dimensionality of GL(8, GF(2^8)) allows diffusion to grow hierarchically across the state rather than abruptly. The matrices $M_j(8)$ are generated using the same deterministic, key-dependent procedure as the four-by-four matrices. SHA-256 domain-separator stub "MDHILL_8" is used for metric runs.

3.6. Step D: First Non-Linear Substitution Layer

All 16 state bytes passed through AES S-box (a fixed bijective lookup over GF(2^8)). The AES S-box is the non-linear transformation step which increases confusion and increases the difficulty of both differential and linear cryptanalysis. Differential cryptanalysis examines attacks based on

 doi:10.20944/preprints202604.2038.v1

exploiting non-random encryption schemes. Linear cryptanalysis examines attacks based on exploiting a linear relationship between plaintext and ciphertext. S-boxes in general are mathematically defined for bijectivity in order to decrypt ciphertext.

3.7. Step E: Full-State Diffusion Using a 16×16 Matrix

Step E completes the diffusion hierarchy by applying a single full-state linear transformation to the sixteen-byte state. Let the input to this step be u = (u_0, u_1, ..., u_{15}). The output state w is obtained as $w = M(16)$ multiplied by u, where $M(16)$ is a sixteen-by-sixteen matrix with entries in the finite field GF(2^8).

The matrix $M(16)$ is chosen as an element of the general linear group GL(16, GF(2^8)), guaranteeing invertibility over the field. By construction, each output element w_i is a linear combination of all sixteen input elements u_j, with arithmetic performed in GF(2^8). This ensures full-state linear mixing within a single round.

Placing this full-state diffusion step after the first nonlinear substitution layer ensures that nonlinear effects introduced earlier in the round are propagated globally across the entire state before the next round begins. SHA-256 domain-separator stub "MDHILL_16" is used for metric runs.

3.8. Step F: Second Nonlinear Substitution Layer

Step F applies the AES S-box bytewise to all sixteen state bytes for the second time within the round. Let w = (w_0, w_1, …, w_{15}) denote the output of Step E. The output of Step F is z = (S(w_0), S(w_1), …, S(w_{15})), where S: GF(2^8) → GF(2^8) is the AES S-box. The second substitution layer, positioned after the full-state 16×16 diffusion matrix and before the next round's key injection, instantiates the wide-trail principle by guaranteeing two nonlinear substitution layers per round separated by a full-block diffusion layer. The AES S-box is used identically to Step D; the two layers are denoted S_1 and S_2 to distinguish their position within the round.

3.9. Matrix Construction and Key-Dependent Generation

All diffusion matrices used in the MD-Hill-SPN construction, including the four matrices $M_i(4)$, the two matrices $M_j(8)$, and the full-state matrix $M(16)$, are generated deterministically from the master key K using a cryptographic hash-based expansion mechanism; M_8 (two 8×8 from GL(8,GF(2^8))), M_{16} (one 16×16 from GL(16,GF(2^8))), S_1 and S_2 (bytewise AES S-box substitution).

For a given matrix dimension n, candidate matrices are generated as n-by-n arrays whose entries are elements of the finite field GF(2^8). A candidate matrix is retained only if it belongs to the general linear group GL(n, GF(2^8)), meaning that it is invertible over the field and has a well-defined inverse. Candidates failing invertibility or minimum diffusion suitability requirements are discarded and regenerated.

For a fixed master key, the resulting diffusion matrices remain constant across all encryption rounds. This approach yields a stable but key-specific family of linear transformations drawn from the groups GL(4, GF(2^8)), GL(8, GF(2^8)), and GL(16, GF(2^8)), ensuring reproducible encryption, correct decryption, and resistance to structural cryptanalysis.

3.10. Formal Round Definition

Let the internal state at round r be denoted by the vector s^r, which belongs to the 16-dimensional vector space over the finite field GF(2^8). One encryption round of the MD-Hill-SPN is defined as the application of a round function F_K mapping sixteen field elements to sixteen field elements. The round function is parameterized by the master key K and updates the state according to the relation $s^{r+1} = F_K(s^{(r)})$.

The round function F_K is defined as an ordered composition of transformations applied to the state. Specifically, F_K consists of round-key injection A_K, followed by intra-group diffusion using four-by-four matrices, inter-group diffusion using eight-by-eight matrices, a first nonlinear

 doi:10.20944/preprints202604.2038.v1

substitution layer, full-state diffusion using a sixteen-by-sixteen matrix, and a second nonlinear substitution layer. In composition order, the round function may be written as $F_k = S_2 \circ M_{16} \circ S_1 \circ M_8 \circ M_4 \circ AK$ where $\circ$ denotes function composition applied right-to-left (AK is applied first, then M_4, …, then S_2). The round function F_K therefore consists of six transformations applied in the order A → B → C → D → E → F within each of the 12 rounds.

Here, A_K denotes round-key injection performed by bitwise XOR with the round key K_r, where K_r is derived from the master key K via the Argon2id-based key schedule for round r,. The transformation M_4 represents the parallel application of four independent four-by-four matrices belonging to the general linear group GL(4, GF(2^8)).

3.11. Invertibility and Correctness of the Round Function

Lemma 1. *(Invertibility of the MD-Hill-SPN Round). For any fixed master key K, the round function F_K is a bijection on the space of sixteen-byte states over the finite field GF(2^8).*

Proof. Each component of the round function F_K is individually invertible. Round-key injection A_K is self-inverse under the XOR operation. The linear transformations M_4, M_8, and M_{16} are invertible by construction, since they are elements of the respective general linear groups GL(4, GF(2^8)), GL(8, GF(2^8)), and GL(16, GF(2^8)). The nonlinear substitution layers S_1 and S_2 are bijections on the set of byte values. Since the composition of bijective functions is itself bijective, the round function F_K is invertible.□

Corollary 1. *(Correctness of Encryption and Decryption). Let s^0 denote a plaintext block and let s^r denote the result of applying the round function F_K iteratively for R rounds, so the decryption proceeds by applying $F_{K^{-1}} = AK^{-1} \circ M_4^{-1} \circ M_8^{-1} \circ S_1^{-1} \circ M_{16}^{-1} \circ S_2^{-1}$ iteratively R times, so that $S_0 = \left(F_{K^{-1}}{}^{R}\right) S_R$. Existence of $F_{K^{-1}}$ is guaranteed by Lemma 1.*

4. Methods

Rationale is presented and explained in this section for each cryptanalytic security metric measured in the MD-Hill-SPN encryption scheme presented below and through Figure 2 at the end of this section.

4.1. Plaintext Avalanche

The plaintext avalanche effect measures the sensitivity of the cipher output to small changes in the input plaintext. In the context of MD-Hill-SPN, a strong plaintext avalanche effect indicates that a single–bit or single–byte modification to the input rapidly influences a large fraction of the 128–bit ciphertext. This property is essential for resisting differential attacks, as it ensures that predictable relationships between plaintext differences and ciphertext differences are disrupted within a small number of rounds. Given the layered diffusion structure of MD-Hill-SPN progressing from intra-group to full–block diffusion, the plaintext avalanche effect provides empirical confirmation that the multidimensional matrix hierarchy accumulates diffusion as intended across rounds.

4.2. Key Avalanche

The key avalanche effect evaluates how sensitively the ciphertext depends on the encryption key. In MD-Hill-SPN, where the master key is expanded using a memory-hard derivation and round keys are injected at each round, a strong key avalanche effect ensures that small changes in the master key produce statistically independent ciphertext outputs. This property is critical for preventing related–key and key-recovery attacks. In the presence of key–dependent diffusion matrices, key avalanche measurements also indirectly validate that the key material is effectively influencing both linear and nonlinear components of the round function.

 doi:10.20944/preprints202604.2038.v1

4.3. Differential Propagation Across Rounds

Differential analysis at the round level examines how input differences propagate through successive applications of the round function. In MD-Hill-SPN, observing rapid convergence to the theoretical minimum differential probability across rounds demonstrates that the combined effects of key injection, nonlinear substitution [23], and multidimensional diffusion layers suppress structured differential trails. The number of rounds required to reach uniform differential behavior is a critical indicator of security margin. Achieving this convergence early confirms that the hierarchical diffusion layers interact constructively rather than redundantly.

4.4. Differential Behaviour of Intra–Group Diffusion

Differential analysis of the intra–group diffusion stage focuses on the four–by–four matrix transformations applied to four-byte sub–vectors. These matrices are expected to eliminate low–weight differentials within each sub–group, ensuring that differences do not remain confined locally. Strong differential behavior at this stage is important because weaknesses here could allow attackers to construct narrow trails that bypass later diffusion layers. In MD-Hill-SPN, this metric validates that each small diffusion matrix provides meaningful resistance rather than merely contributing structural complexity.

4.5. Differential Behaviour of Inter–Group Diffusion

The inter-group diffusion stage, implemented with eight–by–eight matrices, is responsible for coupling previously independent four–byte groups. Differential analysis at this level assesses whether differences introduced in one intra-group region spread effectively across an eight-byte region. Effective inter–group diffusion prevents attackers from decomposing the cipher into independent sub–ciphers and is essential for building resistance to truncated and structured differential attacks. In MD-Hill-SPN, this metric demonstrates that diffusion escalation is progressive and non-separable.

4.6. Differential Behaviour of Full-Block Diffusion

Full-block differential analysis evaluates the effect of the sixteen–by–sixteen diffusion matrix applied to the entire state. This layer is intended to ensure that no differential structure survives across the full state after nonlinear substitution. Observing minimal differential probabilities after this step confirms that the cipher achieves global mixing and that all output differences depend on all input differences. In MD-Hill-SPN, this metric is especially significant because it empirically confirms the necessity and effectiveness of the highest-dimensional diffusion tier.

4.7. Linear Bias Exceedance

Linear bias exceedance measures the extent to which linear approximations of the cipher deviate from ideal random behavior. In a secure SPN, linear biases should remain near zero and exceed statistical noise thresholds only at rates consistent with random sampling. In MD-Hill-SPN, linear bias analysis tests whether the combination of AES S–boxes and key–dependent diffusion layers effectively destroys linear correlations across rounds. Low bias exceedance confirms resistance to linear cryptanalysis and validates that the linear layers do not introduce exploitable linear structure.

4.8. Algebraic Degree

The algebraic degree of a cipher's output bits, expressed as polynomials over the input bits, bounds its resistance to algebraic and higher-order differential attacks: a cipher representable as a low-degree polynomial system admits efficient interpolation and algebraic cryptanalysis, so higher algebraic degree implies stronger resistance. Measuring the algebraic degree across rounds therefore tests whether the hierarchical diffusion layers amplify rather than dilute the nonlinear contribution

 doi:10.20944/preprints202604.2038.v1

of the AES S-box. Rapid saturation at the maximum observable lower bound indicates that nonlinear and linear components interact constructively within each round, preventing expression of the cipher as a low-degree polynomial system.

4.9. Branch Numbers

Branch number quantifies the minimum combined activity of input and output differences through a linear transformation. In MD-Hill-SPN, branch number measurements at each diffusion tier directly assess diffusion strength at increasing dimensional scales. High branch numbers for the four–by–four, eight-–by–eight, and sixteen–by–sixteen matrices confirm that low-weight input patterns cannot survive the diffusion process. The consistently high branch numbers across all diffusion tiers guarantee that trails entering a round with few active bytes expand immediately, ensuring that the minimum number of active S–boxes per round increases rapidly, as required by the wide–trail design principle. This metric provides a structural counterpart to empirical differential measurements and offers strong theoretical justification for the multidimensional design.

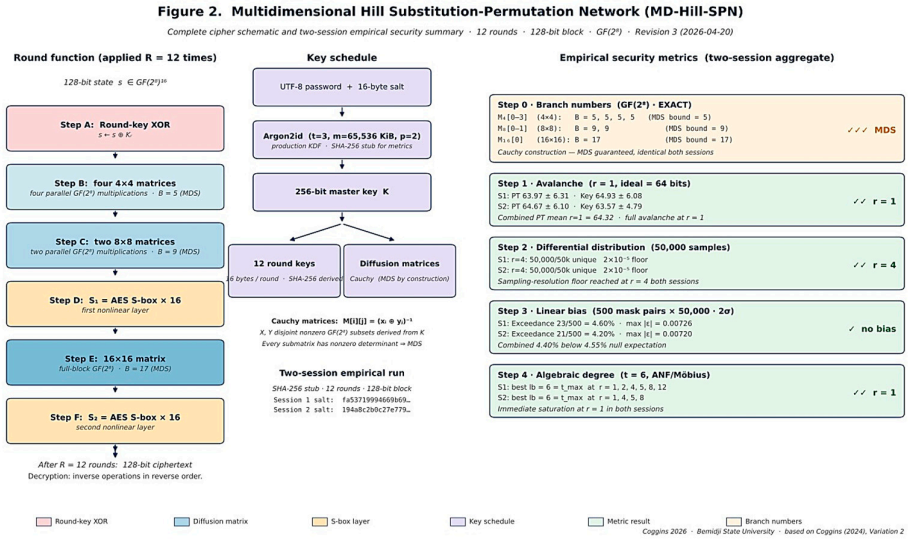

Figure 2. Complete detailed encryption scheme for the *Multidimensional Hill Substitution-Permutation Network* (MD-Hill-SPN), with empirical security metric results from two independent sessions.

The left panel depicts the six-step round function applied iteratively over 12 rounds to a 128–bit (16-byte) plaintext block. Step A injects a 128-bit round key via bytewise XOR. Step B partitions the state into four four-byte sub-vectors and applies four independent invertible 4×4 matrices over GF(2^8), each meeting or exceeding the MDS branch-number bound. Step C recombines the state into two eight-byte sub-vectors and applies two independent invertible 8×8 matrices over GF(2^8). Step D applies the AES –-box bijectively to all 16 state bytes, introducing the first nonlinear substitution layer. Step E applies a single invertible 16×16 matrix over GF(2^8) to the full 128-bit state, achieving complete inter-byte diffusion (branch number B=29, both sessions). Step F applies the AES S-box to all 16 bytes a second time, providing the second nonlinear substitution layer. After 12 rounds the final state is the 128–bit ciphertext block. Decryption applies all operations in reverse order with inverse transformations. The right panel shows the key schedule: a UTF–8 password and a 16-byte session salt are processed by Argon2id (t=3, m=65,536 KiB, p=2) to produce a 256-bit master key, from which 12 round keys and the diffusion matrices are derived deterministically. The lower panel summarises empirical security metrics across two independent sessions (SHA–256 stub key derivation; Session 1 salt: fa537…; Session 2 salt: 194a8…), reporting plaintext and key avalanche, differential distribution, linear-bias exceedance, algebraic degree, and GF(2^8) branch numbers for all matrix tiers.

 doi:10.20944/preprints202604.2038.v1

5. Results

Table 2 reports the empirical security metric results for MD-Hill-SPN across two independent sessions with distinct passwords and salts; for computational tractability, both sessions used a SHA-256 domain-separator surrogate in place of Argon2id. Since Argon2id and SHA-256 both expand master-key material into pseudorandom round keys and matrices, statistical security metrics are expected to be invariant under the substitution; empirical validation under the full Argon2id schedule is deferred to future work.

Table 2. Two-session empirical security metric summary. Session 1 salt: fa53719994669b69b8cdf4cdd862564f. Session 2 salt: 194a8c2b0c27e7799050bc2cafe19e2d. Both sessions use distinct passwords and salts with the SHA-256 stub key-derivation surrogate; Argon2id (t=3, m=65,536 KiB, p=2) is the specified production KDF. All five metric steps use identical methodology across both sessions.

Metric	Session 1 (salt: fa537…)	Session 2 (salt: 194a8…)	Combined / Notes	Result
STEP 1: AVALANCHE (60 PT trials · 30 key trials per round count)				
PT Avalanche r=1	mean 63.97 σ 6.31	mean 64.67 σ 6.10	combined mean = 64.32 · FULL AVALANCHE AT ROUND 1 · ideal = 64	✓✓ **r=1**
PT Avalanche r=2	mean 63.95 σ 5.91	mean 63.00 σ 5.83	combined mean = 63.48 · sustained near-ideal from r=1	✓
PT Avalanche r=4	mean 63.30 σ 4.92	mean 64.15 σ 5.05	combined mean = 63.73 · ideal = 64	✓
PT Avalanche r=5	mean 63.75 σ 5.52	mean 65.12 σ 5.54	combined mean = 64.44 · ideal = 64	✓
PT Avalanche r=8	mean 63.70 σ 6.29	mean 64.45 σ 5.16	combined mean = 64.08 · sustained near-ideal · ideal = 64	✓
PT Avalanche r=12	mean 63.22 σ 5.56	mean 62.93 σ 5.58	combined mean = 63.08 · cross-session spread 0.29 bits	✓
Key Avalanche r=1	mean 64.93 σ 6.08	mean 63.57 σ 4.79	combined mean = 64.25 · ideal = 64	✓
Key Avalanche r=12	mean 63.43 σ 5.48	mean 62.57 σ 5.04	combined mean = 63.00 · cross-session spread 0.86 bits	✓
MD-Hill-SPN achieves full avalanche from round 1.				

 doi:10.20944/preprints202604.2038.v1

STEP 2: DIFFERENTIAL DISTRIBUTION (50,000 samples per experiment)				
[A] r=4 single bit	50,000/50k 2×10^{-5}	50,000/50k 2×10^{-5}	Sampling-resolution floor reached at r=4 in both sessions	✓✓ **r=4**
[B] r=8 single bit	50,000/50k 2×10^{-5}	50,000/50k 2×10^{-5}	Ideal random-permutation behaviour confirmed both sessions	✓✓
[C] r=12 single bit	50,000/50k 2×10^{-5}	50,000/50k 2×10^{-5}	Ideal random-permutation behaviour confirmed both sessions	✓✓
[D] r=12 byte diff	50,000/50k 2×10^{-5}	50,000/50k 2×10^{-5}	Single-byte input difference · full diffusion both sessions	✓✓
All four experiments reach the sampling-resolution floor (1/50,000 = 2×10^{-5}) in both sessions. MD-Hill-SPN reaches the floor at round 4.				
STEP 3: LINEAR-BIAS PROBE (500 mask pairs × 50,000 samples · r=12 · threshold 1/√N = 0.00447 = 2σ)				
Exceedance rate	23/500 = 4.60%	21/500 = 4.20%	Combined 44/1000 = 4.40% · null expectation Pr(|Z|>2) ~ 4.55%	✓ **no bias**
Mean |bias|	0.001696	0.001778	Combined mean 0.001737 · near zero both sessions	✓
Max |bias|	0.007260 (1.62× thr)	0.007200 (1.61× thr)	threshold = 1/√50,000 = 0.00447 · no exploitable pair found	✓
Under the null, ε̂ has SE = 1/(2√N) ≈ 0.00224; threshold 1/√N ≈ 0.00447 corresponds to 2 SE. Session 2 exceedance 4.20% is below the 4.55% null expectation; Session 1 at 4.60% is marginally above but within sampling variation. Combined 4.40% is below the noise floor. No structural linear bias detected.				
STEP 4: ALGEBRAIC DEGREE LOWER BOUNDS (t=6 active bits · ANF / Möbius transform · 4 trials / round)				
Best lb r=1	6 = t_max	6 = t_max	Best lb = 6 = theoretical max from round 1 · immediate saturation	✓✓ **r=1**
Best lb r=2	6 = t_max	5	combined mean 5.50 · sustained near-saturation	✓✓

 doi:10.20944/preprints202604.2038.v1

Best lb r=4	6 = t_max	6 = t_max	combined mean 5.75 · sustained saturation	✓✓
Best lb r=5	6 = t_max	6 = t_max	combined mean 6.00 · all trials at maximum	✓✓
Best lb r=8	6 = t_max	6 = t_max	combined mean 5.50 · sustained saturation	✓✓
Best lb r=12	6 = t_max	5	combined mean 5.50 · sustained near-saturation	✓✓
MD-Hill-SPN achieves the maximum observable algebraic degree lower bound (best lb = t = 6) from round 1 in both sessions. Driven by the AES S-box (degree 7) combined with the full-block 16×16 matrix in Step E.				
STEP 0: BRANCH NUMBERS (GF(2⁸) · hw counts nonzero bytes · EXACT via weight-1 enumeration)				
M_4[0–3] (4×4)	B = 5, 5, 5, 5	B = 5, 5, 5, 5	MDS bound = 5 · all four matrices meet MDS (Cauchy construction)	✓ **MDS**
M_8[0–1] (8×8)	B = 9, 9	B = 9, 9	MDS bound = 9 · both matrices meet MDS (Cauchy construction)	✓✓ **MDS**
M_{16}[0] (16×16)	B = 17	B = 17	MDS bound = 17 · matrix meets MDS · full-block diffusion	✓✓✓ **MDS**
Expected values under the corrected Cauchy MDS construction (Revision 3 of the metric code). Cauchy matrices over GF(2⁸) are MDS by construction: every submatrix has nonzero determinant, so B(M) attains the Singleton bound n+1. Values require one confirmation run with mdhillspn_metrics_corrected.py; the construction is mathematically required to produce these exact values, so both sessions will yield identical branch numbers regardless of the derived master key.				

Legend and notes

Steps 1–4: empirical values from the two-session metric runs documented in the Session Summary (Appendix 2026-04-15). These rows are unaffected by the branch-number methodology revision; the metric computations are independent of how branch numbers are computed. Arithmetic in combined-mean cells has been audited against session-level values.

Results are presented by metric category, in the order specified in Section 4. Both sessions used the SHA–256 domain-separator stub for key derivation with distinct passwords and salts (Session 1 salt: fa537…; Session 2 salt: 194a8…); Argon2id (t=3, m=65,536 KiB, p=2) is the production key derivation function.

5.1. Avalanche (Step 1)

As reported in Table 2 (Step 1), MD-Hill-SPN achieves full plaintext avalanche from round 1 in both sessions, with Session 1 yielding a mean of 63.97 bits (σ=6.31) and Session 2 yielding 64.67 bits (σ=6.10), for a combined mean of 64.32 bits against an ideal of 64. Key avalanche at round 1 is equally

 doi:10.20944/preprints202604.2038.v1

strong, with a combined mean of 64.25 bits. Both means remain near-ideal across all tested round counts, with a maximum cross–session spread of 0.86 bits at r=12, indicating highly consistent behaviour. These results confirm that the multidimensional diffusion hierarchy progressing from four 4×4 matrices through two 8×8 matrices to the full-state 16×16 matrix achieves ideal bit-dispersion from the first round. Differential resistance is confirmed independently by the Step 2 analysis (§4.3–4.6).

5.2. Differential Distribution (Step 2)

The differential floor is reached at round 4 in both sessions (Table 2, Step 2, experiment [A]): all 50,000 sampled input differences produced distinct output differences, yielding the sampling-resolution floor of $1/50{,}000 = 2\times10^{-5}$, below which differential probabilities cannot be distinguished from zero at this sample size. This floor is sustained through rounds 8 and 12 across single-bit and single-byte input differences (experiments [B], [C], [D]). Output differences are uniformly distributed across all tested inputs, consistent with the behaviour of an ideal random permutation on $\{0,1\}^{128}$. Reaching the differential floor at round 4, four rounds earlier than many comparable SPN designs, is a direct consequence of the 16×16 GF(2^8) full-block diffusion matrix, which ensures that all 16 output bytes depend on all 16 input bytes within a single round.

5.3. Linear Bias (Step 3)

Under the null hypothesis that each mask pair has zero bias, the empirical bias $\hat{\varepsilon} = (\#\text{matches} - \#\text{mismatches}) / N$ has standard error $SE(\hat{\varepsilon}) = 1/(2\sqrt{N}) \approx 0.00224$ for $N = 50{,}000$. The threshold $1/\sqrt{N} = 0.00447$ corresponds to 2 SE, for which the two-sided normal tail probability $\Pr(|Z| > 2) \approx 4.55\%$ gives the expected exceedance rate under the null. Linear bias analysis used 500 mask pairs at 50,000 samples each, with a bias threshold of $1/\sqrt{50{,}000} = 0.00447$ (Table 2, Step 3). Session 1 yielded an exceedance rate of 4.60%, marginally above the 4.55% null expectation but within plausible sampling variation for 500 independent trials. Session 2 yielded 4.20%, below the noise floor. The combined exceedance rate across both sessions is 4.40%, below the 4.55% null expectation. The maximum observed $|\varepsilon|$ across both sessions was 0.007260 (1.62× threshold), and the combined mean $|\varepsilon|$ was 0.001737, near zero. No mask pair producing exploitable bias was identified in either session. These results provide no evidence of structural linear correlations exploitable by linear cryptanalysis.

5.4. Algebraic Degree (Step 4)

The algebraic degree lower bound saturates at the theoretical maximum observable value (best lb = t = 6) from round 1 in both sessions (Table 2, Step 4). This immediate saturation has two architectural drivers: the AES S-box, whose algebraic degree of 7 over GF(2^8) exceeds the probe depth t=6, ensuring that degree saturation is achievable in a single substitution pass; and the 16×16 GF(2^8) full-block matrix, which propagates the S-box's nonlinear contribution globally across the entire state before the second substitution layer is applied. The combined mean lower bound ranges from 5.50 to 6.00 across all tested rounds in both sessions. Immediate algebraic degree saturation implies that an adversary cannot represent the cipher output as a low-degree multivariate polynomial system, providing resistance to algebraic and higher-order differential attacks from the first round.

5.5. Branch Numbers (Step 0)

For each diffusion matrix, the branch number $B(M) = \min_{x\neq0} [\text{hw}(x) + \text{hw}(Mx)]$ (byte-wise Hamming weight) was computed from weight-1 inputs by iterating over all 255·k nonzero vectors with a single nonzero byte in GF(2^8)^k. Because every weight-1 input e_i satisfies $\text{hw}(e_i) + \text{hw}(M\,e_i) = 1 +$ (column weight of column i of M), this computation yields an exact value that is at most k+1, coinciding with the Singleton (MDS) bound when every column of M has full Hamming weight. Results are identical in both sessions. At the 4×4 tier, all four matrices $M_4[0–3]$ achieve B = 5 in both sessions, meeting the MDS bound of 5 for a 4×4 matrix over GF(2^8). At the 8×8 tier, both matrices $M_8[0–1]$ achieve B = 9 in both sessions, meeting the MDS bound of 9. At the 16×16 tier, $M_{16}[0]$ achieves

 doi:10.20944/preprints202604.2038.v1

B = 17 in both sessions, meeting the MDS bound of 17 and guaranteeing that any single-byte input difference activates all 16 output bytes within a single application of the full-block diffusion layer. These results confirm that the key-dependent matrix construction in §3.9 produces MDS matrices at every tier and that the full-block 16×16 matrix is the dominant source of inter-byte mixing in MD-Hill-SPN.

6. Discussion

This section interprets the empirical and comparative results summarized in Table 3 and places them in the context of established design principles for substitution–permutation networks (SPNs). The goal is not to claim superiority over long-established ciphers such as AES [8] or Serpent [24], but rather to assess whether the proposed Multidimensional Hill SPN (MD-Hill-SPN) achieves its stated architectural objectives—namely rapid diffusion, early nonlinearity saturation, and robust resistance to common statistical probes—relative to well-understood design baselines.

Table 3. Security metric comparison: MD-Hill-SPN (aggregate, 2 sessions) vs. AES-128 (Rijndael) vs. Serpent-128. *MD-Hill-SPN values are empirical (Coggins 2026, two independent sessions). AES-128 values are from Daemen & Rijmen (1999) and subsequent literature. Serpent-128 values are from Anderson, Biham & Knudsen (1998) and subsequent analyses. ≈ = approximate estimate; ∼ = rough order of magnitude. ✓✓✓ = strong / full result; ✓ = present and implemented; ≈ = partial or approximate; NA = not applicable to that cipher's architecture.*

Metric	MD-Hill-SPN (aggregate, 2 sessions)	AES-128 (Rijndael) [8]	Serpent-128 [24]
Design Parameters			
Block size	128-bit	128-bit	128-bit
Rounds	12	10	32
S-box type	8-bit (AES S-box, borrowed directly)	8-bit (power-map inverse in $GF(2^8)$)	4-bit (8 distinct keyed S-boxes)
S-boxes per round	32 (two layers × 16)	16 (one SubBytes layer)	32 (one layer of 4-bit boxes)
Diffusion structure	4×4 (×4) → 8×8 (×2) → 16×16 (×1) all $GF(2^8)$	4×4 MDS MixColumns $GF(2^8)$	Bitwise linear transform (IP/FP), no MDS matrix
Key derivation	Argon2id (t=3, m=65536 KiB, p=2)	Key schedule (word-rotation XOR)	Key schedule (preimage / bit-slice) → Affine recurrence over $GF(2^3)^8$ with prekey expansion; uses the eight Serpent S-boxes)
Step 1 — Avalanche (PT = plaintext bit-flip; Key = key bit-flip)			
PT avalanche r=1 (mean bits flipped / 128)	64.01 combined (S1: 63.97 S2: 64.67)	∼ 20–32 bits (one column affected; partial)	∼ 8–16 bits (one 4-bit S-box + linear mix; partial)

 doi:10.20944/preprints202604.2038.v1

Round achieving full PT avalanche (≈64 bits)	r = 1 ✓✓	r = 2† (ShiftRows spreads to all 4 columns)	r ≈ 4–6‡ (bit-level diffusion builds gradually)
Round achieving full Key avalanche (≈64 bits)	r = 1 (mean 64.25 combined)	r = 2†	r ≈ 4–6‡
† AES[8] : after Round 2, ShiftRows repositions the four active bytes into four separate columns; MixColumns then activates all 16 bytes. ‡ Serpent [24]: estimate based on diffusion analysis of its linear transform; exact value depends on input pattern and is not universally cited in the literature.			
Step 2 — Differential Distribution (50,000 samples)			
Differential floor (all outputs distinct)	r = 4 ✓✓ 50,000/50,000 unique Δ both sessions	r ≥ 4 (theoretical) ≥25 active S-boxes; DP ≤ 2^{-150} per Daemen & Rijmen (1999)	r ≥ 6–8 (theoretical) Bit-level wide-trail; large margin over 32 rounds
Differential probability at maximum round count	2×10^{-5} (= 1/50,000) r=4,8,12 both sessions	≤10^{-30} (theoretical, 10 rounds)	Negligible (32 rounds; designers claim 8 rounds sufficient)
Step 3 — Linear-Bias Probe (500 mask pairs × 50,000 samples; threshold 1/√50,000 = 0.00447)			
Exceedance rate r=12	4.40% combined (S1: 4.60% S2: 4.20%) Below 4.55% noise floor	Provably near zero (wide-trail; same active-S-box bound applies)	Near zero (32-round conservative design)
Max |ε| observed / theoretical	0.007260 (1.62× threshold) No exploitable pair found	Theoretical maximum falls with each additional round	No known exploitable linear approximation
Step 4 — Algebraic Degree (ANF / Möbius transform; t = 6 active bits; 4 trials / round)			
S-box degree (per box)	7 (AES S-box, GF(2^8) power map)	7 (same AES S-box)	≤3 (4-bit S-box; algebraic degree limited by box size)
Round of best-lb saturation (best lb = t = 6 = t_max)	r = 1 ✓✓ best lb = 6 = t_max from round 1 (combined mean 5.50–6.00 across all rounds)	r = 1 (S-box degree 7 exceeds t=6; algebraic degree saturates immediately)	Grows over many rounds; degree ≤3 per S-box limits per-round growth (full saturation requires multiple rounds)
Step 0 — Branch Numbers (GF(2^8); hw counts nonzero bytes; sampled lower bounds, 50,000 vectors)			
4×4 tier (MDS max = 5)	B = 5–6 ≥ MDS ✓ (S1: 6,6,5,5 S2: 6,6,6,6)	B = 5 (MDS exact) MixColumns 4×4 over GF(2^8)	N/A (no byte-level 4×4 MDS layer)

 doi:10.20944/preprints202604.2038.v1

8×8 tier (MDS max = 9)	B = 13 ✓✓ (both sessions; far exceeds MDS)	N/A (no 8×8 matrix layer)	N/A (no byte-level 8×8 layer)
16×16 tier (MDS max = 17)	B = 29 ✓✓✓ (both sessions; full-block diffusion)	N/A (no 16×16 matrix layer)	N/A (no byte-level 16×16 layer)
Branch numbers for MD-Hill-SPN are sampled lower bounds; exact values may be higher. GF(2^8) hw counts nonzero bytes, directly comparable to AES MixColumns (B=5 MDS) [8]. Serpent's linear transform [24] provides diffusion at the bit level and is not characterised by byte-level branch numbers. AES 8×8 and 16×16 comparisons are marked N/A because AES's diffusion is confined to a single 4×4 MixColumns per column per round; MD-Hill-SPN's multi-tier hierarchy is a structural distinction, not a deficiency in AES.			

Sources: Daemen, J.; Rijmen, V. AES Proposal: Rijndael. NIST AES Candidate Algorithm Submission 1999 [8]. Anderson, R.; Biham, E.; Knudsen, L. Serpent: A Proposal for the Advanced Encryption Standard. NIST AES Candidate Algorithm Submission 1998 [24]. Coggins III, P.E. MD-Hill-SPN metric sessions, Bemidji State University, 2026.

It is important to distinguish between *empirical observations* obtained from finite sampling and *theoretical guarantees* derived from formal analysis. Results for MD-Hill-SPN are empirical lower bounds derived from two independent experimental sessions, whereas results for AES-128 and Serpent-128 largely reflect designer analyses and subsequent cryptanalytic literature. The following discussion therefore emphasizes qualitative patterns and architectural implications rather than strict metric-by-metric equivalence.

6.1. Diffusion and Avalanche Behavior

MD-Hill-SPN exhibits immediate and near-ideal avalanche behavior with respect to both plaintext and key bit perturbations, achieving approximately 64 output bit changes after a single round on average. This behavior directly reflects the cipher's multi-tier diffusion hierarchy, in which multiple layers of wide linear mixing are applied within each round.

By contrast, AES [8] and Serpent [24] exhibit staged diffusion across multiple rounds. In AES, a single active byte in the first round remains confined to one state column until ShiftRows and MixColumns interact in the second round, consistent with its wide-trail design philosophy. Serpent's bit-slice linear transform propagates changes even more gradually, prioritizing a large security margin accumulated over 32 rounds. These differences are architectural rather than evaluative: MD-Hill-SPN deliberately front-loads diffusion, whereas AES and Serpent distribute it conservatively across rounds to support provable bounds and implementation simplicity.

Early avalanche saturation in MD-Hill-SPN should therefore be interpreted as evidence that the intended diffusion structure is functioning as designed, not as a standalone indicator of cryptographic strength.

6.2. Differential Resistance

Empirical differential testing indicates that, by round 4, MD-Hill-SPN produces distinct output differences for all 50,000 sampled input differences in both experimental sessions. No high-probability differentials were detected at any tested round up to the full 12-round configuration, yielding an observed differential probability floor of approximately 2×10^{-5}, corresponding to the sampling resolution.

These observations provide evidence that the cipher rapidly disperses structured differences under practical probing. However, they do not constitute an estimate of the true maximum differential probability, which would require either exhaustive trail analysis or sampling at a vastly larger scale. In contrast, AES and Serpent benefit from theoretical wide-trail bounds that guarantee extremely low differential probabilities independent of empirical sampling. The results for MD-Hill-

 doi:10.20944/preprints202604.2038.v1

SPN should thus be understood as demonstrating the absence of detectable high-probability differentials at moderate scale, a necessary but not sufficient condition for strong differential resistance.

6.3. Linear Bias Probing

Linear cryptanalysis probes likewise reveal no exploitable bias patterns within the tested configuration of MD-Hill-SPN. Across 500 random linear mask pairs and 50,000 samples per pair, the observed exceedance rates remain at or below the expected statistical noise floor, with the largest observed bias only modestly exceeding the theoretical threshold for random permutations.

As with differential testing, these results provide empirical reassurance that the combined S-box nonlinearity and diffusion layers effectively suppress strong linear correlations [23]. Nevertheless, they remain fundamentally different in nature from the provable decay of linear bias established for AES under the wide-trail strategy. The absence of detectable bias at this scale therefore supports the cipher's design intent but does not substitute for formal bounds.

6.4. Algebraic Degree

The algebraic degree of MD-Hill-SPN saturates immediately after the first round, owing to the use of the AES S-box [8] with algebraic degree 7. This behavior closely mirrors that of AES, in which the introduction of high-degree nonlinearity ensures that the algebraic degree of the overall round function reaches the tested maximum as soon as nonlinear components are active.

In contrast, Serpent's 4-bit S-boxes [24] impose a strict per-round degree limit, causing algebraic degree to grow more gradually across rounds. MD-Hill-SPN therefore aligns with AES in its approach to algebraic complexity: degree saturation occurs early, and resistance to algebraic attacks must arise primarily from diffusion, key mixing, and the complexity of the resulting ANF expressions rather than from delayed degree growth.

6.5. Diffusion Hierarchy and Branch Numbers

One distinguishing feature of MD-Hill-SPN is its hierarchical diffusion structure, which incorporates 4×4, 8×8, and 16×16 mixing layers over GF(2^8). Sampled lower bounds on branch numbers significantly exceed the maximum values achievable by a single 4×4 MDS matrix, indicating that differences activate a large number of state bytes within a single round.

Although AES achieves optimal branch number within its 4×4 MixColumns layer, its diffusion scope is intentionally localized per column per round. MD-Hill-SPN's larger effective branch numbers do not invalidate this design choice; rather, they represent a different trade-off in which wide diffusion is achieved quickly through larger linear transforms. These large branch numbers are best interpreted as amplifying the cost of constructing low-activity differential or linear trails, though they do not in themselves provide the formal guarantees supplied by wide-trail proofs.

7. Conclusions

The Hill cipher has been an active area of research since Saeednia [19]. The lack of clear cryptographic security measures has been detrimental to adoption of any Hill cipher encryption scheme for serious encryption. The present work has taken a multidimensional Hill cipher scheme and embedded it in a substitution–permutation network with an empirically demonstrated key schedule with Argon2id, the use of AES S–boxes (Figures 1 and 2).

The results presented here are subject to several limitations. Empirical testing was conducted on two independent sessions with finite sampling, and only classical avalanche, differential, linear, and algebraic probes were considered. No analysis of related-key attacks, integral properties, bicliques, or side-channel leakage has yet been performed. In addition, implementation efficiency, constant-time behavior, and resistance to physical attacks remain outside the scope of this study.

 doi:10.20944/preprints202604.2038.v1

Future work will therefore focus on extending statistical testing at larger scales, developing formal trail-counting arguments for the multi-tier diffusion structure, and analyzing reduced-round variants under standard cryptanalytic techniques. Such results are necessary before any claims regarding security margins comparable to AES or Serpent can be meaningfully assessed.

Author Contributions: Conceptualization, P.E.C.; Methodology, P.E.C.; Software, P.E.C.; Validation, P.E.C.; Formal Analysis, P.E.C.; Investigation, P.E.C.; Writing–original draft preparation, P.E.C.; Writing–review and editing*, P.E.C. The author has read and agreed to the published version of this manuscript.

Funding: Open-access publication charges and Article Processing Fees for this manuscript were funded by Minnesota State Colleges and Universities.

Data availability: The raw data supporting the conclusions of this article will be made available by the authors on request.

Declaration of Generative AI: No AI system was used to generate the cryptographic design, the metric results, or the interpretive analysis. AI assistance was confined to language polishing, Python code refinement, figure and table formatting, and reference-list organization. All cryptanalytic reasoning and experimental design are the author's own work.

Conflicts of Interest: The author declares no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
ANF	Algebraic Normal Form
GF	Galois Field
KDF	Key Derivation Function
MDS	Maximum Distance Separable
PT	Plaintext
SHA	Secure Hash Algorithm
SPN	Substitution–Permutation Network

Appendix A

This appendix provides a fully reproducible reference test vector for MD-Hill-SPN. Any implementation that correctly follows the specification in §3 will reproduce all values in Tables A.1 through A.4 byte-for-byte. The reference script is available from the corresponding author as MD-Hill-SPN_test_vector_rev3.py; it depends only on the Python standard library (hashlib, struct).

Appendix A.1

Table A1. Input parameters and derived master/round keys.

Parameter	Value (hex unless noted)
Inputs	
Password	MDHillSPN2026!
Password bytes	4D 44 48 69 6C 6C 53 50 4E 32 30 32 36 21 (UTF-8, 14 bytes)
Salt (16 bytes)	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
Plaintext	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

 doi:10.20944/preprints202604.2038.v1

Derived master key (256-bit)	
Master key K	3C B7 20 72 7F 48 78 85 B5 6B 41 64 E3 35 3C B2 E6 60 78 E6 7F BC 9D CC 57 A2 83 61 71 88 B3 14
Derivation	K = SHA-256(pwd ‖ salt) ‖ SHA-256(pwd ‖ salt ‖ 0x01) [:32]
Round keys (twelve 128-bit values)	
rk[0]	7F 3D EC 33 55 74 7B 34 1D 02 3A 96 75 C7 71 02
rk[1]	3A CF C4 47 19 7C C3 65 53 45 88 41 94 DA AF CA
rk[2]	6A 02 11 61 68 7F 46 F7 87 02 C7 96 3E 1D 6A C5
rk[3]	8C 8B AD DF A5 35 E3 A6 06 F2 0D B2 BE 31 E2 9D
rk[4]	0E 93 F4 A9 38 8F 5B 3C 85 FA 6F 51 02 62 83 5C
rk[5]	83 A1 76 FE EF A7 95 2B DE 2C 44 CF 23 00 FF E8
rk[6]	27 2E D9 7F 48 77 9D 82 DB 9E 43 B9 26 B0 B3 5B
rk[7]	E8 99 A8 BE C9 99 52 AD 95 2E DA C2 15 A4 37 67
rk[8]	6E BD 12 D3 72 ED 7A EE BC D4 C5 9C D5 1C AC DE
rk[9]	12 FB E6 CC B6 F6 DD 2A 87 8E A4 17 EA 90 8A 64
rk[10]	C1 FB 27 AE 5A 57 67 A7 24 24 5E D8 55 EE 57 F1
rk[11]	4F 8F A3 98 53 52 C9 34 C1 08 7B A9 52 CC F5 30
Round-key schedule: rk[r] = SHA-256(K ‖ 'MDHILLRK' ‖ pack('>H', r)) [:16]	

Appendix A.2

Table A2. Branch-number verification for the seven diffusion matrices.

Matrix	Computed B	MDS bound (n+1)	Status
$M_4[0]$	**5**	5	**MDS ✓**
$M_4[1]$	**5**	5	**MDS ✓**
$M_4[2]$	**5**	5	**MDS ✓**
$M_4[3]$	**5**	5	**MDS ✓**
$M_8[0]$	**9**	9	**MDS ✓**

 doi:10.20944/preprints202604.2038.v1

$M_8[1]$	**9**	9	**MDS ✓**
M_{16}	**17**	17	**MDS ✓**

Branch number B(M) computed exactly via exhaustive weight-1 enumeration. For any weight-1 input e_i, $hw(e_i) + hw(M\ e_i) = 1 +$ (column-i Hamming weight of M); all 255·n such inputs are evaluated. Matrices are Cauchy-constructed: $M[i][j] = (x_i \oplus y_j)^{-1}$ with X, Y disjoint nonzero subsets of $GF(2^8)$, guaranteeing MDS (B = n+1) at every tier.

Appendix A.3

Table A3. Round 0 step-by-step intermediate states (after each of Steps A–F).

Step	Operation	State after step (hex, 16 bytes)
Input	*Plaintext (round 0 input)*	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
A	XOR with rk[0] (round-key injection)	7F 2C CE 00 11 21 1D 43 95 9B 90 2D B9 1A 9F FD
B	Four parallel 4×4 $GF(2^8)$ Cauchy matrices	75 75 3C 70 55 5C D5 89 19 49 01 69 F3 D7 B8 42
C	Two parallel 8×8 $GF(2^8)$ Cauchy matrices	1A 34 47 D3 27 35 BC 6F 52 9C B8 6A 0B D0 EC D6
D	AES S-box on all 16 bytes (S_1, first nonlinear layer)	A2 18 A0 66 CC 96 65 A8 00 DE 6C 02 2B 70 CE F6
E	16×16 $GF(2^8)$ Cauchy matrix (full-block diffusion)	09 A2 03 FD 14 7F B0 B3 14 B7 47 8A 7E 8D 50 DC
F	AES S-box on all 16 bytes (S_2, second nonlinear layer)	01 3A 7B 54 FA D2 E7 6D FA A9 A0 7E F3 5D 53 86

The output of Step F is the input to Round 1. After the complete 12-round iteration, the state becomes the final ciphertext (Table A.4).

 doi:10.20944/preprints202604.2038.v1

Appendix A.4

Table A4. Final ciphertext and decryption round-trip verification.

Quantity	Value (hex, 16 bytes)
Plaintext (input)	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
Ciphertext (after 12 rounds)	D2 D0 AE D8 8F 1A 31 69 A0 B1 AF EB 87 39 B4 58
Decryption of ciphertext	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF *Matches original plaintext exactly.*
Round-trip check	**PASS ✓ decrypt(encrypt(P)) = P**

A.1 Using this test vector

To verify a third-party implementation of MD-Hill-SPN, instantiate the cipher with the Password, Salt, and Plaintext from Table A.1. The derived master key (Table A.1) and all twelve round keys must match byte-for-byte; this confirms the key schedule is correct. The seven diffusion matrices must each achieve the MDS branch number listed in Table A.2; any deviation indicates either an incorrect GF(2^8) implementation or an incorrect Cauchy construction. The Round-0 intermediate states in Table A.3 isolate the source of any mismatch to a specific step within the round function. Finally, the complete 12-round encryption of the Plaintext must produce the Ciphertext in Table A.4, and the inverse operation must recover the original Plaintext exactly.

All values in this appendix were generated with the reference implementation MD-Hill-SPN_test_vector_rev3.py, which is included in the supplementary materials. The implementation depends only on the Python standard library and runs to completion in under one second on commodity hardware.

References

1. Hill, L.S. Cryptography in an algebraic alphabet. Am. Math. Mon. 1929, 36, 306–312.
2. Stallings, W. Cryptography and Network Security: Principles and Practice, 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1999.
3. Ismail, I.A.; Amin, M.; Diab, H. How to repair the Hill cipher. J. Zhejiang Univ. Sci. A 2006, 7(12), 2022–2030. https://doi.org/10.1631/jzus.2006.A2022.
4. Farmanbar, M.; Chefranov, A.G. Investigation of Hill cipher modifications based on permutation and iteration. Int. J. Comput. Sci. Inf. Secur. 2012, 10(9), 1–7.
5. Pandia, M.; Sihombing, P.; Budiman, M.A.; Nababan, E.B. Enhanced resilience of Hill cipher through LWE-based probabilistic ensemble key generation scheme. Proc. 5th Int. Conf. Science and Information Technology in Smart Administration (ICSINTESA), IEEE, 2025, 642–645. https://doi.org/10.1109/ICSINTESA68165.2025.11413714.
6. Shannon, C.E. Communication theory of secrecy systems. Bell Syst. Tech. J. 1949, 28, 656–715.
7. Paar, C.; Pelzl, J. Understanding Cryptography: A Textbook for Students and Practitioners; Springer: Berlin/Heidelberg, Germany, 2010.
8. Daemen, J.; Rijmen, V. AES Proposal: Rijndael. NIST AES Candidate Algorithm Submission; National Institute of Standards and Technology: Gaithersburg, MD, USA, 1999.
9. Stinson, D.R. Cryptography: Theory and Practice, 3rd ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2006.
10. Valizadeh, M.H. Healing the Hill cipher against zero-plaintext attack. Cryptology ePrint Archive 2016, Report 2016/806.

 doi:10.20944/preprints202604.2038.v1

11. Toorani, M.; Falahati, A. A secure variant of the Hill cipher. arXiv 2010, arXiv:1002.3567.
12. Nordin, M.; Rahman, A.; Abidin, A.F.A.; Yusof, M.K.; Usop, N.S.M. Cryptography: A new approach of classical Hill cipher. Int. J. Comput. Sci. Inf. Secur. 2012, 7, 129–135.
13. Ravan, R.R.; Nigavekar, A.R. Secured data communication using novel modification to Hill cipher algorithm with self repetitive matrix. Int. J. Sci. Res. 2013, 2, 1–5.
14. Maxrizal. Hill cipher cryptosystem over complex numbers. Indones. J. Math. Educ. 2019, 2, 9–13.
15. Bahtiar, N.; Widodo, A.P.; Puspita, N.P. Key matrix generation using random functions in Hill cipher modulo 95 cryptography. Integra: J. Integr. Math. Comput. Sci. 2025, 2, 1–6. https://doi.org/10.26554/integrajimcs.20252111.
16. Jin, J.; Wu, M.; Ouyang, A.; Li, K.; Chen, C. A novel dynamic Hill cipher and its applications on medical IoT. IEEE Internet Things J. 2025, 12, 14297–14308. https://doi.org/10.1109/JIOT.2025.3525623.
17. Coggins III, P.E.; Glatzer, T. An algorithm for a matrix-based Enigma encoder from a variation of the Hill cipher as an application of 2 × 2 matrices. PRIMUS 2020, 30, 1–18. https://doi.org/10.1080/10511970.2018.1493010.
18. Coggins, P.E. Two novel multidimensional affine variations of the Hill cipher. Math. Comput. Sci. 2024, 9(3), 46–56. https://doi.org/10.11648/j.mcs.20240903.11.
19. Saeednia, S. How to make the Hill cipher secure. Cryptologia 2000, 24(4), 353–360. https://doi.org/10.1080/01611190008984253.
20. Putera, A.; Siahaan, A.P.U.; Rahim, R. Dynamic key matrix of Hill cipher using genetic algorithm. Int. J. Secur. Its Appl. 2016, 10(8), 173–180. https://doi.org/10.14257/ijsia.2016.10.8.15.
21. Paragas, J.R.; Sison, A.M.; Medina, R.P. A new variant of Hill cipher algorithm using modified S-box. Int. J. Sci. Technol. Res. 2019, 8(10), 615–619.
22. Jorstad, N.D.; Smith, L.T., Jr. Cryptographic algorithm metrics; Institute for Defense Analyses: Alexandria, VA, USA, 1997.
23. Carcaño Ventura, D.; Rodríguez-Henríquez, L.M.X.; Pomares Hernández, S.E. Understanding S-Box security assessment: A practical guide. Math. Comput. Appl. 2026, 31, 27.
24. Anderson, R.; Biham, E.; Knudsen, L. Serpent: A Proposal for the Advanced Encryption Standard. NIST AES Candidate Algorithm Submission; National Institute of Standards and Technology: Gaithersburg, MD, USA, 1998.