

Extending the Hill Cipher to be Asymptotic to a One-Time Pad

by Tony Patti

April 26, 2026

Abstract

The Hill Cipher, when known only as a mod 26 linear polygraphic substitution cipher, is often dismissed in modern cryptography due to its vulnerability to known-plaintext attacks. This paper presents a robust extension of the Hill Cipher that effectively removes its inherent linearity through four strategic modifications. By incorporating Galois Fields, affine transformations, S-P (substitution-permutation) layers, and (especially) probabilistic bit insertion, this cryptosystem achieves a state where a single plaintext block can map (without chaining) to over 10^{900} unique ciphertexts, making it asymptotically equivalent to a One-Time Pad (OTP), while remaining mathematically accessible to a high school audience. Ultimately, the traditional linear Hill Cipher encryption equation $C = M * P$ is extended through these four simple extensions to the encryption equation $C = SP(M * INS(P) + A)$ where SP is the substitution-permutation, "INS" is the insertion of random bits (for probabilistic encryption) into the plaintext vector, and "A" is the Affine Transformation.

1. Introduction to the Hill Cipher

Invented by Lester S. Hill in 1929, the original Hill Cipher was the first practical polygraphic encryption algorithm, using matrix-vector multiplication for encryption and decryption. The classic encryption equation is $C = M * P$, where "M" is the encryption matrix, "P" is the plaintext vector, and "C" is the ciphertext. While elegant, its purely linear nature means that if an attacker knows a few plaintext-ciphertext pairs, they can use simple algebra to recover the secret matrix "M". Modern cryptography requires "confusion and diffusion", properties the original Hill cipher lacks in the face of modern cryptanalysis.

For these reasons the Hill Cipher is frequently deprecated, but John Dooley says it best, in his book *"History of Cryptography and Cryptanalysis: Codes, Ciphers, and Their Algorithms"*, this is how Prof. Dooley starts Chapter 10 on page 167:

"Modern cryptology rests on the shoulders of three men of rare talents. William Friedman, Lester Hill, and Claude Shannon moved cryptology from an esoteric, mystical, strictly linguistic realm into the world of mathematics and statistics. Once Friedman, Hill, and Shannon placed cryptology on firm mathematical ground, other mathematicians and computer scientists developed the new algorithms to do digital encryption in the computer age."

2. The Four Pillars of the Extended Hill Cipher

To transform this classic algorithm into a secure modern cryptosystem, we implement four primary extensions:

1. **Galois Fields GF(p) and GF(p^n):** Rodney Cooper's 1980 Cryptologia article "Linear Transformations in Galois Fields and their Application to Cryptography" extended the "mod 26" Hill Cipher to Galois Fields, and we build on Cooper's foundation. Instead of non-prime modular arithmetic that Lester Hill utilized (mod 26), we operate within Galois Fields utilizing a prime modulus "p". This ensures every non-zero element has a multiplicative inverse, which is critical for generating the decryption matrix and maintaining algebraic structure. GF(p^n) is the math of polynomials over a prime field (mod p) and also modulo an irreducible polynomial.
2. **Affine Transformation (The "A" Vector):** We introduce an addition step after the traditional Hill cipher matrix-vector multiplication, which we can now label as the "Hill Affine Cipher". This ensures that even if the plaintext is a vector of all zeros, the ciphertext will not be zero, breaking the property of fixed-point linearity. Conceptually we now have the encryption equation: $C = M * P + A$

$$c_{ij} = \sum_{q=1}^f b_{iq} a_{qj}, \quad c_i = b_i + \sum_{q=1}^f b_{iq} a_q$$

In his 1931 paper, in the equation shown above (from page 144), Hill introduces the mathematical scaffolding for an affine transformation through the second expression, where the inclusion of the single-subscripted term b_i effectively acts as a translation vector added to the linear product. While Hill does not utilize modern terminology like "offset" or "affine," this notation reveals an early realization that a purely homogeneous system—where a null input vector would yield a null output—was cryptographically insufficient. However, because he uses the letter b to represent both the matrix elements (b_{iq}) and this additive component (b_i), the distinction remains opaque to any reader not closely scrutinizing the shift from double to single subscripts.

Toorani and Falahati in 2009 published "A Secure Variant of the Hill Cipher". They explicitly define the Affine Hill Cipher as $C = (KP + V) \text{ mod } m$, where V is a translation vector. They further extend this by making V dynamic (changing per block), which is the most robust published version of the affine extension.

3. **S-P (Substitution and Permutation):** We map numeric outputs to ASCII characters. This step utilizes the fundamental concepts of Substitution-Permutation Networks (SPNs), which are the building blocks of most modern block ciphers, including the Advanced Encryption Standard (AES). Details in the next section.
4. **Probabilistic Encryption (Random Bit Insertion):** By choosing a large prime modulus, we "hide" random bits within each element. For example, in $GF(997727)$, we can encrypt two bytes (16 bits) per element, leaving three bits to be filled with cryptographically secure random values. We place one random bit at the beginning, one random bit in the middle, and one random bit at the end. Conceptually "R12345678R12345678R" where "R" represents a random bit, and the digits are the two bytes.

3. The S-P (Substitution-Permutation) Layer: Combinatorial Explosion

Historically the Hill Cipher is viewed as simply a bunch of numbers which suffer from being a set of numbers-only linear equations, but that need not be the output of the Hill Cipher. The S-P layer serves as the final "scrambler," which randomly maps each vector element into a string of ASCII characters. This ensures that the mathematical output of the matrix operations is obscured before transmission. This architecture relies on the same principles as the Shannon-inspired S-boxes and P-boxes found in high-level cryptography but implemented in a way that remains conceptually simple.

3.A. Substitution (Digit-to-ASCII Mapping)

In this system, each of the 6 digits in an element is mapped (one-to-many) to one of 5 possible ASCII characters, which are mutually exclusive, here is one possible example, this map would be part of the secret key:

Each Digit "0"	maps randomly into one of these five:	NPMIz
Each Digit "1"	maps randomly into one of these five:	FuytR
Each Digit "2"	maps randomly into one of these five:	LUOms
Each Digit "3"	maps randomly into one of these five:	WnwJk
Each Digit "4"	maps randomly into one of these five:	HExq1
Each Digit "5"	maps randomly into one of these five:	QcGvo
Each Digit "6"	maps randomly into one of these five:	VbBDX
Each Digit "7"	maps randomly into one of these five:	apeAK
Each Digit "8"	maps randomly into one of these five:	rCgdf
Each Digit "9"	maps randomly into one of these five:	hSZYj

So, the 6-digit number 507610 in $GF(997727)$ might map digit-by-digit (one possible example):

5-0-7-6-1-0 into "Q-N-p-b-R-I"

3.A.1. The Complexity of the Unknown Mapping

This mapping is secret, and part of the key. The cryptanalyst is dealing with the **combinatorial explosion** of the character set.

We are using a standard set of 50 unique ASCII characters (10 digits * 5 characters each) assigned to the digits 0–9:

- **Partitioning the Set:** The attacker must determine which 5 characters belong to which of the 10-digit groups.
- **Key Space:** The number of ways to partition 50 unique characters into 10 groups of 5 is calculated by the multinomial coefficient:

$$\frac{50!}{(5!)^{10}}$$

This is a massive number, approximately 4.9×10^{43} , or roughly **145 bits of security** just for the mapping table itself, before even considering the underlying matrix or polynomial transformations.

This is why we embrace ciphertext expansion in the use of the random mapping, and having human-readable output — we are mapping a 20-bit number into six ASCII letters, each of which is 7 bits (stored in an 8-bit byte), so 48 bits total.

3.A.2. Cryptanalytic Difficulty

Without the mapping, the attacker faces two primary hurdles:

- **Loss of Mathematical Structure:** In a standard matrix cipher, the attacker can try to perform a Known Plaintext Attack (KPA) using linear algebra. Because the cryptanalyst cannot 'read' the underlying digits within the ASCII representation, they are unable to populate the vectors for a $C = MP + A$ algebraic attack. The unknown mapping creates a “semantic gap” that prevents the attacker from mapping observed ciphertext strings to the numerical elements required to solve the system of equations.
- **Homophonic Diffusion:** Because "5" can be "Q", "c", "G", "v", or "o", the same digit appears as different characters in different parts of the ciphertext. This breaks the 1:1 relationship required for simple substitution solvers.

3.B. Permutation (Position Shuffling)

Once the digits are mapped, their positions within the string are shuffled. For a 6-digit number, the number of ways to arrange those positions is defined by $6!$ (6-factorial) = 720 permutations.

For example, if we had the six digits "123456" we might permute and output as any of the following, with the permutation being stored in the secret key:

526134 or 246513 or 635214 or 431526 or 346152 or 324165 or 356241 or 642531

3.C. Total Element Entropy

Since the substitution and permutation steps are independent operations, their effects are multiplicative. So, $4.9 \times 10^{43} * 720 = 3.5 \times 10^{46} = 154$ bits of security.

4. The Extended Encryption Equation

The modified encryption process is represented by the following formula:

$$C = SP(M * INS(P) + A)$$

- **INS(P)**: Plaintext with inserted random bits.
- **M**: The secret encryption matrix.
- **A**: The secret Affine Transformation vector.
- **SP**: The final Substitution and Permutation layer.

5. This Probabilistic Encryption is Asymptotic to a OTP

The primary strength of the One-Time-Pad (OTP) is that each time you re-encrypt the same plaintext, you get completely different ciphertext output.

The primary weakness of standard block ciphers is that the same plaintext always produces the same ciphertext. In this system, the "INS" (insertion) and "SP" (substitution) steps introduce so much entropy that a single block can result in a staggering variety of outputs.

We introduce Probabilistic encryption into the Hill Cipher. Probabilistic encryption is the use of randomness in an encryption algorithm, so that when encrypting the same message several times it will, in general, yield different ciphertexts.

As mentioned above, we insert three random bits into each two bytes of plaintext.

Conceptually "R12345678R12345678R" where "R" represents a random bit, and the digits are the two bytes.

Specifically, with a vector of dimension 1000, **the number of possible different ciphertexts for ONE plaintext block (without chaining) is $2^{3000} = 10^{900}$** because there are 3,000 random bits inserted into the plaintext vector (of dimension 1000). This property makes the system

"asymptotic" to a One-Time Pad because, like a true OTP, the ciphertext provides virtually no information about the original message to an observer without the key. The sheer volume of possible outputs for a single input renders traditional pattern recognition and frequency analysis ineffective.

6. Mathematical Flexibility: The Agnostic (large) Matrix

The most obvious aspect of a matrix is that it can be of any size. In Hill's papers, he demonstrates using small 2x2, 3x3, or 4x4 matrices, but using modern computers, we can implement secret key matrices which vary over six orders of magnitude from 4x4 (with 320 key bits) to 4000 x 4000 (with 320 key million bits) in each matrix, or any size in-between (e.g. a 1000x1000 matrix is easy for a computer to handle today).

The discussion above is just a cryptographic framework.

The Hill Cipher is agnostic to the nature of the matrix elements as long as they form a field.

A primary strength of this extended system is the flexibility of the matrix M . While the original Hill Cipher operated on simple integers modulo 26 in a small matrix, the modern extension can utilize any elements that satisfy the properties of a **Field** (where addition, subtraction, multiplication, and division by non-zero elements are defined).

All four of the below have been implemented in various programming languages (see References 10 and 11).

6.A. Integers in $GF(p)$

The most straightforward implementation uses prime fields. By selecting a large prime " p ", we ensure that every matrix with a non-zero determinant has an inverse, allowing for guaranteed decryption. For large values of " p " such as 997727, 99.9999% of all randomly generated matrices are invertible. This provides a clean, high-speed computational base.

6.B. Polynomials in $GF(p^n)$

By moving to Extension Fields, elements are represented as polynomials. Multiplication and addition occur modulo a monic irreducible polynomial. This adds a layer of "algebraic complexity"; to an attacker, the relationship between "plaintext polynomials" and "ciphertext polynomials" is significantly harder to model than simple integer congruences.

6.C. Block Matrices (Nested Matrices)

If you read the last paragraph of Hill's 1931 paper, he wrote "... it is easy to set up an algebra for ranges of matrices whose elements are in turn matrices ..."

In other words, each "element" of the encryption matrix can itself be another matrix! This "Matrix-within-a-Matrix" structure (Block Matrices) exponentially increases the diffusion of a

single plaintext bit across the entire ciphertext block. We now have a four-dimensional (4D) data structure for the encryption and decryption matrices. See Reference 11.

6.D. Gaussian Integers ($a + bi$)

By using Gaussian integers (complex numbers where both the real and imaginary parts are integers), we introduce a two-dimensional numerical structure to every element. This is particularly useful for mapping complex data types or adding a "geometric" layer to the encryption, further obscuring the underlying plaintext through the interaction of real and imaginary components during matrix multiplication.

7. Security Analysis and PQC Resistance

The strength of this system lies in its resistance to modern and future threats:

7.A. Resistance to Chosen Plaintext Attack (CPA):

The insertion of random bits (the "INS" function) is the primary source of security. For a 1000×1000 matrix, this creates over $2^{3000} = 10^{900}$ variations for a single plaintext block. Because the ciphertext is different every time the same message is sent, the system achieves **Semantic Security**.

7.B. Post-Quantum Cryptography (PQC) Implications:

Quantum computers (via Shor's Algorithm) excel at solving linear systems. However:

1. **Grover's Algorithm:** The 10^{900} entropy space makes a quantum "brute force" search mathematically impossible.
 2. **Structural Obscurity:** The S-P layer and the use of complex elements (Polynomials/Gaussian Integers) break the periodicity required for quantum Fourier transforms. An attacker cannot solve a matrix they cannot mathematically reduce to a linear form.
-

8. Parallel Execution and Performance Optimization

To handle the computational complexity of matrix transformations and high-volume homophonic substitution, it is recommended to use the **Rust** programming language's **Rayon crate**. This provides a data-parallelism framework that ensures memory safety without the "data race" risks common in C or C++.

8.A. Parallelizing Matrix-Vector Multiplications

The core of the cryptosystem involves matrix-vector multiplication (and vector addition). Because each vector element in a large block is independent during the transformation phase, we utilize "par_iter()" to distribute these operations across all available logical CPU cores.

1. **Work Stealing:** Rayon's work-stealing scheduler dynamically balances the load, ensuring that if one calculation takes longer (due to specific complexity) other cores remain productive.
2. **Performance Gain:** On an 8-core/16-thread or 12-core/24-thread workstation, the transformation phase approaches a near-linear speedup, critical for maintaining high throughput despite the mathematical density of the cipher.

8.B. Concurrent Homophonic Mapping

The **Digit-to-ASCII mapping** (Substitution) is an "embarrassingly parallel" task. Since the mapping of one 6-digit element into 48 bits of ASCII does not depend on the state of the previous element, the entire ciphertext block is processed in parallel.

- **Vectorization:** By processing thousands of elements simultaneously, the ciphertext expansion (mapping 20 bits to 48 bits) incurs negligible latency.
- **Memory Safety:** Rust's ownership model ensures that the shared secret mapping table is accessed immutably across threads, preventing any corruption during the expansion process.

8.C. Impact on Cryptanalysis Resistance

While parallelism aids the legitimate user, it also provides a "performance floor" for the system. By leveraging high-performance hardware, the cipher can employ more complex transformations—increasing the "Work Factor" for a serial attacker—while remaining performant for the authorized user operating on multi-core server hardware.

9. Conclusion

The 1929 Hill Cipher provides a foundation upon which we can build a modern cryptosystem which is asymptotic to a One-Time Pad (OTP), because it provides different ciphertext when re-encrypting the same plaintext block (without chaining). We achieve this via four extensions: Galois Fields, Affine Transformation, Substitution-Permutation, and Probabilistic Encryption (inserting random bit into the plaintext). These straight-forward extensions are all accessible, without requiring advanced mathematics.

10. References

1. Hill, Lester S. (1929). "Cryptography in an Algebraic Alphabet". *The American Mathematical Monthly*.
2. Hill, Lester S. (1931) "Concerning Certain Linear Transformation Apparatus of Cryptography". *The American Mathematical Monthly*, March 1931, pages 135 - 154.
3. Cooper, Rodney H. (1980). "Linear Transformations in Galois Fields and their Application

- to Cryptography". *Cryptologia*
4. Dooley, John (January 1, 2018). "[10.1 The Shoulders of Giants: Friedman, Hill, and Shannon](#)". *History of Cryptography and Cryptanalysis: Codes, Ciphers, and Their Algorithms*. Springer. p. 167. ISBN 978-3-319-90442-9
 5. Wikipedia "Substitution–permutation network".
https://en.wikipedia.org/wiki/Substitution–permutation_network
 6. Wikipedia "Probabilistic encryption"
https://en.wikipedia.org/wiki/Probabilistic_encryption
 7. Wikipedia "Ciphertext expansion" https://en.wikipedia.org/wiki/Ciphertext_expansion
 8. Wikipedia "Semantic security" https://en.wikipedia.org/wiki/Semantic_security
 9. Wikipedia "Lester S. Hill" https://en.wikipedia.org/wiki/Lester_S._Hill
 10. Patti, Tony "An Interesting Example at the Intersection of Matrix Mathematics and Cryptography - Version 2". <https://cryptosystemsjournal.com/an-interesting-example-at-the-intersection-of-matrix-mathematics-and-cryptography-version-2.pdf>
 11. Patti, Tony "Introducing the Hill-GF-4D Cryptosystem! Four-dimensional Matrix-within-Matrix Cryptosystem inspired by the last paragraph of Lester Hill's 1931 paper!"
<https://cryptosystemsjournal.com/matrix-within-matrix.html>
 12. Shannon, Claude. (1948). "A Mathematical Theory of Communication". *The Bell System Technical Journal*, Volume 27, pages 379–423, 623–656, July, October, 1948.
<https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf>
 13. Shannon, Claude. (1949). "Communication Theory of Secrecy Systems". *Bell System Technical Journal*. Volume 28-4, pages 656-715, October 1949).
<https://pages.cs.wisc.edu/~rist/642-spring-2014/shannon-secrecy.pdf>
 14. Chhotaray et al; Encryption by Hill Cipher and by a novel method using Chinese remainder theorem in Galois field, January 2013, International Journal of Signal and Imaging Systems Engineering 6(1):38-45.
https://www.researchgate.net/publication/264814070_Encryption_by_Hill_cipher_and_by_a_novel_method_using_Chinese_remainder_theorem_in_Galois_field
cited in [US Patent# 12,609,809](#) issued April 21, 2026.
 15. Toorani, M., & Falahati, A. (2009). A secure variant of the Hill cipher. 2009 IEEE Symposium on Computers and Communications, 313–316.
<https://arxiv.org/pdf/1002.3567>
 16. Sastry, V. U. K., & Samson, C. (2012). A generalized hill cipher involving different powers of a key, mixing and substitution. International Journal of Advanced Research in Computer Science, 3(4), 110–114. Utilizes involutory (self-inverse) matrices and non-prime modular arithmetic "mod 256" to secure block data. It enhances the traditional linear transformation by incorporating different powers of a key matrix alongside specific mixing and substitution functions to increase cryptographic complexity.
https://www.researchgate.net/publication/268188054_A_MODERN_ADVANCED_HILL_CIPHER_INCLUDING_A_PAIR_OF_INVOLUTORY_MATRICES_AS_MULTIPLICANDS_AND_INVOLVING_A_SET_OF_FUNCTIONS