UACM

Universidad Autónoma de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA DE SOFTWARE

Cifrado de Texto Imprimible Basado en Grupos Cuánticos

T E S I S

PARA OBTENER EL TÍTULO DE

LICENCIADO EN INGENIERÍA DE SOFTWARE

P R E S E N T A

EMMANUEL DE JESÚS GONZÁLEZ RAMOS

D I R E C T O R: **MTRO. FEDERICO JUÁREZ ALMARAZ**

C O D I R E C T O R: **DR. CARLOS ADOLFO DÍAZ RODRÍGUEZ**

Ciudad de México, octubre de 2025.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

Abstract

A stream cipher model is presented, based on algebraic structures derived from Quantum Mechanics, specifically within the framework of Quantum Group Theory. The encryption process relies on operations corresponding to braiding and coantipode, which serve as the core mechanisms for data transformation. The proposed methodology employs a symmetric key to generate dynamic subkeys, and applies Base64 encoding to ensure a legible representation of the encrypted output. To validate the model, statistical metrics such as Shannon Entropy, the Coincidence Index, Correlation Coefficients (Chi-Square and Pearson), along with Key Sensitivity analysis and criteria proposed by NIST. The evaluation results demonstrate chaotic behavior, high diffusion, and successful recovery of the original text.

Keywords: Symmetric Encryption, Braided Categories, Hopf Algebras, Base64, NIST.

Resumen

Se presenta un modelo de Cifrado en Flujo basado en estructuras algebraicas provenientes de la Mecánica Cuántica en su principal apartado de la teoría de Grupos Cuánticos, y sus operaciones correspondientes al Trenzado y la Coantípoda, que sirven como el proceso principal de la transformación de datos. A su vez la metodología emplea una llave simétrica para generar sub-llaves dinámicas y codificación Base64 como una herramienta para la presentación legible de los datos cifrados. Con la finalidad de validar el modelo se aplicaron métricas estadísticas como lo son la Entropía de Shannon, el Índice de Coincidencia, Coeficientes de Correlación: Chi-Cuadrado y Pearson, así como pruebas de Sensibilidad de Llave y criterios propuestos por NIST. Los resultados obtenidos en la evaluación demuestran un comportamiento caótico, alta difusión y la correcta recuperación del texto original.

Palabras Clave: Cifrado Simétrico, Categorías Trenzadas, Álgebras de Hopf, Base64, NIST.

Agradecimientos

Quiero expresar mi más sincero agradecimiento al M. en C. Federico Juárez Almaraz, quien no solo ha sido mi director de tesis, sino también, una inspiración a lo largo de mi carrera demostrando su compromiso y vocación. Agradezco también a mi Codirector el Dr. Carlos Adolfo Días Rodríguez, por brindarme su confianza, así como su acompañamiento académico y personal, fundamental en cada etapa de este proceso.

A los docentes que, a lo largo de mi trayectoria, influyeron decisivamente en mi formación, y a la Universidad Autónoma de la Ciudad de México (UACM), por brindarme un lugar digno y una educación de calidad para mi desarrollo profesional.

A mis padres, Jesús González del Ángel y Ernesta Ramos Justino, quienes son los pilares fundamentales en mi vida. Gracias por su amor incondicional, por su sacrificio, por su paciencia y su constante apoyo en cada paso de mi camino. Este logro, es sin duda, también el suyo.

Extiendo mi gratitud al jurado evaluador, por sus observaciones críticas y enriquecedoras que contribuirán significativamente al fortalecimiento de esta investigación y conocimientos futuros.

Dedicatoria

A mi hermana, Xaxiry Magaly González Ramos, por ser una inspiración constante de fortaleza y resiliencia. Me siento profundamente orgulloso de ti y agradecido por tenerte a mi lado. Tu dedicación, tus esfuerzos y tu capacidad para superar cualquier obstáculo son un ejemplo para mí, así como una fuente de motivación en mi vida.

A Analuisa Medina Ramos, por tu amistad sincera y valiosa. Me alegra enormemente ver tus logros personales y el crecimiento que has tenido, y me enorgullece haber compartido esta etapa universitaria. Gracias por tu apoyo y por ser una amiga tan incondicional y presente.

A Ángel Cruz Olvera, por tu constante motivación y por el amistoso espíritu competitivo que compartimos en este camino académico.

A mis mejores amigos, Francisco Torres Aguilar y Helmut Uriel Teco Coronel, por sus más de 20 años de amistad, por estar conmigo, por escucharme, por apoyarme e inspirarme a seguir adelante y por creer en mí.

Contenido

Índice de Figuras

Índice de Tablas

1. Introducción

El presente trabajo de tesis propone el diseño y construcción de un modelo de cifrado basado en la abstracción de conceptos algebraicos inspirados en la mecánica cuántica, en particular, de estructuras propias de las categorías trenzadas y las álgebras de Hopf [1]. A través de esta reinterpretación matemática, se busca representar propiedades fundamentales de dichas estructuras dentro de un entorno de computación clásica, con el objetivo de desarrollar un esquema criptográfico funcional que no dependa del uso de hardware especializado.

Por consiguiente, se plantea un algoritmo orientado al cifrado de texto imprimible, cuya implementación se realiza en el lenguaje de programación MATLAB. Este entorno de desarrollo no solo permite validar su funcionamiento operativo, sino también establecer pruebas comparativas frente al estándar criptográfico más consolidado: el Advanced Encryption Standard (AES), avalado por el National Institute of Standards and Technology (NIST) [2].

Para el desarrollo de esta propuesta, el trabajo se encuentra estructurado bajo tres ejes fundamentales:

1) Fundamentación teórica que sustenta la construcción del modelo.
2) Diseño e implementación computacional del algoritmo propuesto.
3) La evaluación experimental de su desempeño.

Estos aspectos serán abordados en los capítulos siguientes, los cuales, conforman la presente tesis de investigación.

1.1. Planteamiento del Problema

En la actualidad, los algoritmos criptográficos como los estándares AES, RSA o ECC constituyen soluciones ampliamente aceptadas y aplicadas en sistemas de seguridad digital. Estos modelos se encuentran fundamentados en estructuras algebraicas que han sido rigurosamente analizadas en términos de seguridad, eficiencia y viabilidad computacional [3] [4].

De forma paralela, en las últimas décadas ha emergido con relevancia el campo de estudio de criptografía cuántica, con un particular desempeño enfocado en la Distribución Cuántica de Claves (QKD), siendo esta una línea de investigación prometedora para revolucionar los mecanismos de seguridad en el escenario del cómputo cuántico [5].

Sin embargo, el aprovechamiento real de los principios formales provenientes de la mecánica cuántica, requieren dispositivos altamente especializados, entornos controlados y recursos tecnológicos que, en el contexto actual, se encuentran fuera del alcance de los sistemas convencionales. Esto plantea una brecha entre la teoría cuántica y su aplicación cotidiana, lo que ha motivado la búsqueda de metodologías alternativas que faciliten su integración.

Dentro de este marco, se considera pertinente investigar las posibilidades de extraer conceptos provenientes de estructuras algebraicas asociadas a los Grupos Cuánticos, para trasladarlos a un entorno computacional clásico. De esta forma, se propone un modelo criptográfico que incorpore las abstracciones de la mecánica cuántica sin depender del uso de hardware especializado.

Por lo tanto, la problemática de estudio se relaciona con la formalización de la teoría cuántica dentro de una arquitectura computacional coherente, capaz de integrar la lógica algebraica en módulos funcionales, y la validación de su comportamiento mediante pruebas estructurales y métricas funcionales. Más allá de ser una simple propuesta, se trata de una alternativa exploratoria con el objeto de contribuir al diseño de nuevas estructuras criptográficas en el ámbito de la computación actual.

1.2. Justificación

Este proyecto tiene como propósito la exploración de conceptos provenientes de la teoría de Grupos Cuánticos, que a través de su reinterpretación permitan modelar la estructura de un algoritmo de cifrado innovador, adaptado al ámbito de la computación convencional [1].

Por ende, no se pretende emular ni replicar fenómenos físicos, sino más bien, establecer una analogía funcional de la teoría cuántica, mediante la representación lógica sustentada en operaciones modulares, campos finitos y matrices [3] [4]. Dando origen a un modelo de cifrado versátil, basado en estructuras cambiantes y flujos dinámicos, que puede resultar conceptualmente novedoso respecto a los esquemas criptográficos modernos.

Además, esta investigación resulta pertinente desde una perspectiva formativa y metodológica, al consolidar conocimientos multidisciplinarios que integran la Ingeniería de Software: desde la concepción y abstracción matemática, hasta el análisis de requisitos, el modelado de diagramas, la implementación y la aplicación de pruebas funcionales.

1.3. Hipótesis

El marco teórico basado en estructuras algebraicas de Grupos Cuánticos permitirá modelar de manera coherente la arquitectura interna del cifrado propuesto, y los resultados experimentales y comparativos demostrarán su funcionalidad, aplicabilidad al texto imprimible y viabilidad, que posibilita la idea de una alternativa potencial frente a los métodos criptográficos modernos.

1.4. Objetivo General

Desarrollar un algoritmo criptográfico dinámico y no convencional, fundamentado en operadores derivados de los Grupos Cuánticos, aplicable en el procesamiento seguro de texto imprimible.

1.5. Objetivos Específicos

1) Analizar los fundamentos teóricos de los Grupos Cuánticos, con principal énfasis en las Álgebras de Hopf y las Categorías Trenzadas sobre los operadores Trenzado y Coantípoda.

2) Elaborar e implementar los operadores matemáticos asociados al Trenzado y la Coantípoda, mediante el uso de aritmética modular, campos finitos y matrices, para dirigir el mecanismo central del proceso de cifrado.

3) Construir un esquema de cifrado simétrico en el que cada carácter sea transformado según reglas asociadas a la sub-llave y la selección alternante de los operadores cuánticos.

4) Adaptar el modelo de cifrado para operar sobre entradas compuestas por texto imprimible, considerando el rango de caracteres definido sobre la codificación ASCII y su extensión Latin-1.

5) Desarrollar un prototipo del algoritmo criptográfico propuesto, orientado a la validación del mismo.

6) Establecer criterios, métricas y procedimientos de validación para el algoritmo, con el fin de garantizar su robustez estadística y su pertinencia en el contexto de texto imprimible.

7) Evaluar el desempeño del algoritmo frente al estándar AES, empleando el conjunto de métricas definido en el punto anterior, para determinar sus ventajas y limitaciones.

1.6. Alcances y Limitaciones

Se determinan los siguientes alcances de esta tesis:

1) El algoritmo criptográfico se centra en la entrada de un formato de texto compuesto por los caracteres imprimibles pertenecientes a la codificación estándar ASCII y Latin-1, excluyendo saltos de línea y caracteres de control.

2) Los datos cifrados serán presentados mediante el uso de codificación Base64, siendo el propósito de facilitar su manipulación, portabilidad y almacenamiento en formato imprimible.

3) La programación del algoritmo de cifrado será implementada mediante el entorno de desarrollo MATLAB, aprovechando su facilidad para la manipulación de estructuras matriciales y aplicación de operaciones módulo.

4) Se propone un modelo de cifrado en flujo, es decir, cada carácter es transformado mediante la selección de los operadores propuestos (Trenzado o Coantípoda), considerando el rango perteneciente al estándar de texto Latin-1 (0 a 255).

5) Se realizarán evaluaciones de desempeño mediante la aplicación de métricas comúnmente utilizadas en estudios criptográficos provenientes del procesamiento de la información, como lo son:

- Entropía de Shannon
- Coeficiente de Correlación
- Índice de Coincidencia
- Sensibilidad de Llave
- Pruebas de Aleatoriedad NIST

6) Se aplicará una comparación experimental estadística del modelo propuesto y el estándar AES, haciendo uso de las métricas antes mencionadas. La comparación permitirá identificar fortalezas y posibles mejoras respecto a un cifrado ya establecido.

En consecuente, se establecen las siguientes limitaciones:

1) El sistema no contempla el procesamiento de otros formatos como lo pueden ser archivos binarios, imágenes o estándares textuales no mencionados.

2) No se incluyen pruebas de resistencia ante ataques criptográficos específicos (diferenciales, lineales, etc.).

3) La comparación de desempeño se limita únicamente al algoritmo AES, sin extensión a otras familias criptográficas.

2. Estado del Arte

2.1. Criptografía

Según Menezes et al. (1996): “La criptografía es el estudio de técnicas matemáticas relacionadas con aspectos de la seguridad de la información como la confidencialidad, la integridad de los datos, la autenticación de entidades y la autenticación del origen de los datos.” [4]

Se puede definir como una disciplina que tiene como objetivo la protección de información mediante la aplicación de técnicas y métodos matemáticos – como el álgebra y la aritmética – capaces de transformar los datos de forma que se impida el acceso al contenido original por parte de personas no autorizadas. Siendo esta área de estudio conocida como criptografía, y a los procedimientos empleados en dicha transformación denominados como sistemas/algoritmos criptográficos o simplemente de cifrado [3].

Fig. 1. Proceso General de Cifrado y Descifrado.

La Fig.1., ejemplifica el proceso básico de cifrado y descifrado mediante la comunicación de un mensaje. El mensaje original al cual nos referiremos como “texto plano” está representado por la palabra “Hola”, este es cifrado por el emisor para enviar

un mensaje ilegible representado por símbolos incomprensibles “#@&%”. Esta forma alterada del mensaje se conoce como “texto cifrado” o “criptograma”.

Para que el receptor pueda acceder al mensaje original, se aplica el proceso inverso llamado descifrado, siendo reversible únicamente mediante el uso de una “llave” secreta dada por el emisor, sirviendo como el elemento autenticador que permite a los participantes autorizados (quienes poseen la llave) interpretar correctamente el mensaje, evitando el acceso a la información original de posibles “intrusos” que intercepten el texto cifrado.

De acuerdo con Menezes et al. (1996) [4] y Stallings (2017) [3], los métodos de cifrado se pueden clasificar en dos categorías:

Criptografía de Llave Simétrica (2.1.1)

El cifrado simétrico, también conocido como cifrado convencional o cifrado de llave única, era el único tipo de cifrado utilizado antes del desarrollo del cifrado de llave pública en la década de 1970. Sigue siendo, con diferencia, el más utilizado de los dos tipos de cifrado. Este consiste en la transformación del texto plano en texto cifrado mediante la llave secreta y un algoritmo de cifrado, haciendo uso de la misma llave se aplica el descifrado para la recuperación del texto [3] [4].

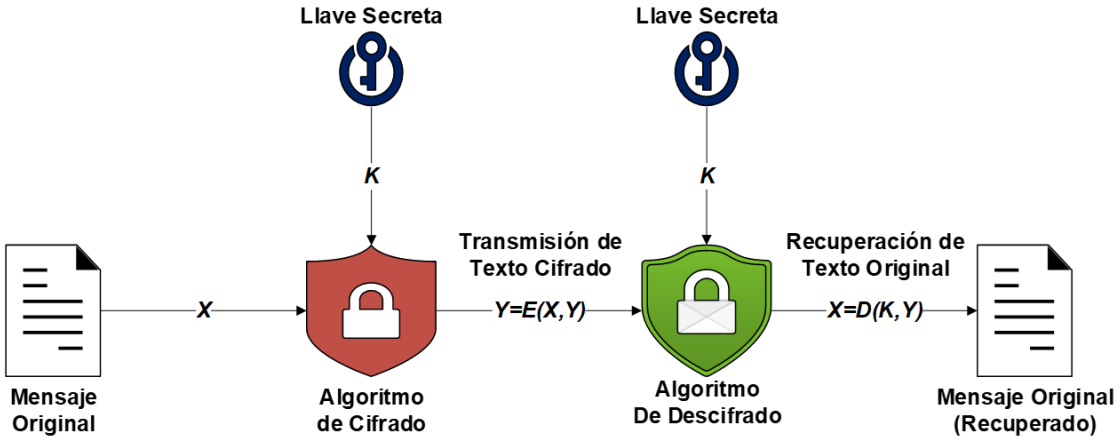


Fig. 2. Proceso de Cifrado de Llave Simétrica. Adaptado de *[3]*

Expresamos el modelo matemático de la Fig.2., tal que, dado un mensaje X, una llave K, entonces:

$$Y = E(K, X), \quad X = D(K, Y) \tag{1}$$

Donde:

- E es la función de cifrado.
- D es la función inversa o descifrado.
- Y es el criptograma.

La ecuación (1) modela el proceso convencional de cifrado simétrico [3].

A continuación, se presentan ejemplos destacados de este tipo de cifrado, los cuales soportan el formato de texto:

#	Algoritmo	Estructura/ Modo	Ventajas	Limitaciones
1	AES [2]	Bloque (SNP), CBC	Alta seguridad; estándar ampliamente aceptado; eficiente en hardware y software. Resistencia de ataques diferenciales	Requiere relleno (padding); más lento que cifrados de flujo para datos pequeños. Cifrado lento en imágenes grandes.
2	ChaCha20 [6]	Flujo (ARX)	Muy rápido; alta seguridad; resistente a ataques de canal lateral; sin necesidad de padding.	Menos soporte en hardware antiguo; tamaño de llave fijo (256 bits). Alto consumo de memoria.
3	3DES [7]	Bloque	Fue estándar durante décadas y es didáctico para comprender cifrados de bloque. Estructura clara y bien documentada.	Sustituido por AES en la mayoría de aplicaciones modernas. En desuso porque NIST lo declaró obsoleto (desde 2023).

Tabla 1. Cifradores de Llave Simétrica.

2.1.2. Criptografía de Llave Asimétrica

El cifrado asimétrico o también conocido como cifrado de llave pública, hace uso de dos tipos de llave: una pública y una privada. La transformación de texto plano a texto cifrado hace uso de una de las dos llaves y un algoritmo de cifrado. La recuperación del texto plano a partir del texto cifrado hace uso del algoritmo de descifrado en conjunto a las dos llaves antes mencionadas. Este enfoque tiene un fundamento matemático descrito en problemas de factorización o logaritmos discretos [3] [4].

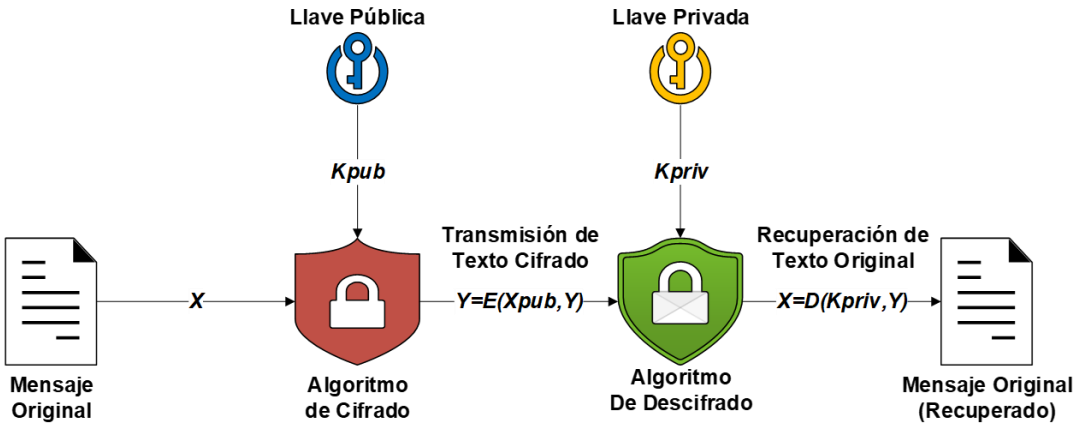


Fig. 3. Proceso de Cifrado de Llave Asimétrica. Adaptado de *[3]*

Su formulación básica es:

$$Y = E[K_{pub}, X], \quad X = D[K_{priv}, Y] \tag{2}$$

- K_{pub} Llave Pública para cifrar.
- K_{priv} Llave Privada para descifrar.

La ecuación (2) modela el proceso convencional de cifrado asimétrico [3].

Algunos ejemplos conocidos en esta categoría de cifrado son:

#	Algoritmo	Estructura/ Modo	Ventajas	Limitaciones
1	RSA [8]	Bloque	Alta seguridad; ampliamente adoptado; compatibilidad con muchos formatos.	Lento para grandes volúmenes; consumo alto de CPU; tamaño de clave grande.

2	ECC [9]	Variante de Flujo	Seguridad alta con claves pequeñas; más rápido que RSA; menor uso de memoria.	Más complejo de implementar; dependiente de parámetros de curva seguros; menor soporte en hardware antiguo.
3	ElGamal [10]	Bloque	Base de muchos esquemas modernos, pero casi no se usa de manera directa. Ofrece la misma seguridad que RSA con claves mucho más pequeñas.	Cifrado más largo que el mensaje; menor eficiencia en comparación con ECC. Vigente solo en su forma derivada ECDSA (la versión en curvas elípticas)

Tabla 2. Cifradores de Llave Asimétrica.

2.2. Métricas de Evaluación Criptográfica

Como en todo sistema de seguridad el enfoque no solo radica en el diseño de algoritmos, sino también en verificar su resistencia ante posibles vulnerabilidades. Para ello, el criptoanálisis nace como una herramienta en función de buscar debilidades que permitan recuperar información sin conocer la llave [3].

En el presente trabajo, no se contemplan ataques prácticos activos o pasivos; en su lugar, se opta por un enfoque criptoanalítico estadístico, orientado a la evaluación de propiedades como la difusión, la aleatoriedad y el desorden del cifrado propuesto. Este análisis cuantificable permitirá inferir el grado de fortaleza del algoritmo frente a los ataques básicos de predicción y análisis estructural [4].

A continuación, se describen las métricas adoptadas para evaluar el comportamiento criptográfico propuesto:

2.2.1. Pruebas de Evaluación NIST STS

Desde finales del siglo XX, el National Institute of Standards and Technology (NIST) se ha consolidado como entidad central en la estandarización de la criptografía, estableciendo métricas y protocolos de validación para algoritmos simétricos y

asimétricos. Entre sus aportes destacan la selección del Data Encryption Standard (DES) y, posteriormente, del Advanced Encryption Standard (AES) en 2001, vigente hasta la actualidad como referente de seguridad [2].

Asimismo, el NIST lidera la transición hacia la criptografía postcuántica y proporciona herramientas de evaluación estadística como la utilizada en este trabajo para medir la robustez y seguridad del algoritmo propuesto.

Esta suite, conocida como NIST STS 800-22, fue diseñada originalmente para validar generadores de números pseudoaleatorios mediante el análisis de secuencias binarias, secuencias que deben mostrar ausencia de patrones repetitivos y estructuras identificables, siendo este criterio hoy adoptado en la evaluación de algoritmos criptográficos [11] [12].

La suite NIST STS 800-22 incluye un conjunto de 15 evaluaciones estadísticas, de las cuales, se pueden destacar [11]:

#	Prueba	Descripción
1	Frequency (Monobit)	Verifica si la proporción de unos y ceros en la secuencia es equilibrada.
2	Runs	Analiza la cantidad y longitud de secuencias consecutivas de unos o ceros.
3	Autocorrelation	Detecta correlaciones entre bits separados por distintos desplazamientos.
4	Poker	Evalúa la distribución uniforme de combinaciones de n bits en la secuencia.
5	Longest Run of Ones in a Block	Mide la longitud máxima de secuencias de unos en bloques.
6	Approximate Entropy	Calcula el nivel de incertidumbre o entropía de la secuencia.
7	Cumulative Sums (Cusum)	Revisa la desviación acumulada respecto a la media esperada.
8	Serial	Verifica la uniformidad en la aparición de pares o tuplas de n bits.

Tabla 3. Métricas de NIST.

2.2.2. Criptoanálisis Estadístico

En este trabajo, el criptograma resultante además de las pruebas NIST, es evaluado mediante un conjunto de métricas estadísticas y de calidad adicionales, seleccionadas según las recomendaciones dadas por Menezes [4] y Stallings [3].

Estas medidas permiten el análisis sobre propiedades como la uniformidad en la distribución, la independencia entre símbolos y la ausencia de patrones residuales. De esta manera, se obtiene una aproximación objetiva sobre aleatoriedad y fortaleza del algoritmo propuesto.

Métrica	Formulación	Que mide
Entropía de Shannon [13]	$H = -\sum_{i=1}^{n} p_i * log\ _2(p_i)$	Nivel promedio de incertidumbre o impredecibilidad en la secuencia.
Correlación Pearson [14]	$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 * \sum_{i=1}^{N}(y_i - \bar{y})^2}}$	Relación lineal entre dos secuencias.
Chi-Cuadrado [14]	$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$	Comparación entre la distribución observada y la distribución esperada (uniforme ideal).
Índice de Correlación (IC) [15]	$I_c(x) = \frac{\sum_{i=1}^{n} f_i\,(f_i - 1)}{n(n-1)}$	Mide el grado de dependencia entre las intensidades de una imagen evaluando la distribución de sus niveles.

Tabla 4. Métricas de Criptoanálisis Estadístico.

2.2.3. Evaluación de Sensibilidad de Llave

La sensibilidad de llave es una propiedad criptográfica que permite evaluar el comportamiento del cifrado a través de pequeños cambios en la llave secreta, lo que debe generar salidas (criptogramas) completamente diferentes [4].

Esta descripción se apega al Principio de Avalancha propuesto por Horts Feistel, y se establece que los cambios (por ejemplo, de un solo bit) deben alterar la salida en al menos un 50% de los bits, lo que garantiza un sistema altamente resistente ante ataques por llaves similares [3] [16].

Para ello, obtenemos los criptogramas a partir del proceso de cifrado (1):

$$C = E(K, X), \;\; C' = E(K', X) \tag{3}$$

- X= Mensaje (Mismo para ambos procesos de cifrado).
- K = Llave Secreta Original.
- K'= Llave Secreta Modificada (Cambio de 1 bit)
- C= Criptograma usando K.
- C'=Criptograma usando K'.

La evaluación de sensibilidad (Key Sensitivy) se realiza calculando la distancia de Hamming (4) entre las dos salidas cifradas (3) del mismo mensaje, tal que [17]:

$$Key\ Sensitivy = \frac{H(C,C')}{n} \tag{4}$$

Para:

- $H(C, C') =$ Cantidad de bits diferentes, distancia de Hamming.
- $n =$ Total de bits del criptograma.

2.3. Distribución Cuántica de Claves

La criptografía cuántica surge como una propuesta que responde a las limitaciones del cifrado moderno y su poder computacional ante a los ordenadores de tipo cuántico. Su base radica en los principios fundamentales de la mecánica cuántica, para establecer un alto grado de seguridad, teóricamente inviolable [18].

La idea es aprovechar las propiedades físicas del sistema cuántico el cual no tiene equivalente con los procesos de cifrado moderno, pues hace uso de [19]:

- **El Principio de Superposición:** Se refiere a la representación de múltiples estados simultáneamente, lo que dificulta la predicción precisa del contenido de una clave sin perturbar el sistema [18].
- **El Entrelazamiento Cuántico:** Cualquier intento de observar un estado entrelazado desde el exterior altera el sistema de forma detectable, impidiendo la intercepción pasiva [18].
- **El Principio de No Clonación**: Establece que es imposible copiar un estado cuántico arbitrario sin introducir perturbaciones, lo cual asegura que un adversario no puede duplicar la clave ni el canal de comunicación sin ser detectado [5] [18].

Estas propiedades convierten a la criptografía cuántica en una propuesta única, donde cualquier intento de espionaje puede ser identificado de manera inmediata, preservando así la integridad del sistema [18].

El campo más consolidado dentro de esta área es la Distribución Cuántica de Claves (Quantum Key Distribution, QKD) que, permite a dos partes generar una llave secreta compartida con seguridad absoluta, incluso en presencia de un ente con capacidad ilimitada de cómputo [5].

A continuación, se describen tres protocolos representativos dentro esta área:

#	Protocolo	Principio Cuántico	Ventajas	Limitaciones
1	BB84 [5]	Polarización de fotones + Principio de no-clonación	Seguridad teórica contra ataques computacionales, detección de eavesdropping.	Limitado a distancias cortas sin repetidores cuánticos.
2	E91 [20]	Entrelazamiento cuántico	No requiere canal de comunicación seguro inicial, detección intrusos mediante desigualdades de Bell.	Complejidad experimental y requerimiento de pares entrelazados estables.
3	QDS [21]	Distribución cuántica de claves + No-clonación	Seguridad contra falsificación y repudio, aplicable a mensajes múltiples.	Infraestructura costosa, todavía en etapa experimental.

Tabla 5. Protocolos de Cifrado Cuántico.

2.4. Cifrado Postcuántico

La criptografía postcuántica (Post-Quantum Cryptography, PQC) es un campo emergente que busca desarrollar algoritmos seguros ante la amenaza que representan las computadoras cuánticas, pero que pueden implementarse en sistemas clásicos [22]. A diferencia de las QKDs, que requieren dispositivos cuánticos especializados, la criptografía postcuántica se centra en métodos resistentes a algoritmos como Shor o Grover, manteniendo compatibilidad con la infraestructura actual [22] [23].

Los algoritmos postcuánticos más destacados incluyen:

#	Algoritmo	Aplicación/ Base Matemática	Ventajas	Limitaciones
1	CRYSTALS Kyber [24]	• Key Encapsulation Mechanism (KEM) • Redes Euclidianas (Lattices, LWE).	Eficiente, clave corta, ya estandarizado (FIPS-203).	Cifrado más pesado que métodos clásicos pero aceptable.
2	CRYSTALS Dilithium [25]	• Firmas Digitales. • Redes Euclidianas (Lattices, LWE).	Buena eficiencia y robustez; estandarizado (FIPS-204).	Firmas relativamente grandes; requiere precisión técnica.
3	SPHINCS+ [26]	• Firmas Digitales. • Funciones Hash.	Seguridad muy alta, resistencia comprobada a largo plazo.	Firmas de gran tamaño y lentitud relativa.

Tabla 6. Algoritmos de Cifrado Postcuántico.

En el contexto de este trabajo, el análisis de la criptografía postcuántica permite enmarcar los desafíos modernos en seguridad digital, y destacar la relevancia de buscar alternativas de cifrado que puedan complementar los enfoques modernos ante los nuevos paradigmas tecnológicos.

3. Marco Teórico

El presente capítulo tiene como propósito sustentar los principios matemáticos que forman la base del modelo de cifrado propuesto. Partiendo del estudio de los Grupos Cuánticos y las estructuras derivadas que hacen posible delimitar los operadores de Torsión y Coantípoda, a representaciones diagramáticas que sirven como un apoyo visual para facilitar su entendimiento como un proceso de transformación de datos.

Más adelante, se abordan los fundamentos aritméticos y algebraicos que posibilitan modelar estos operadores para desarrollar su implementación en computación convencional, mediante el uso de matrices, funciones módulo y campos finitos. Como fase final, se presenta un método de codificación dotando así de un formato seguro de impresión para el criptograma.

3.1. Grupos Cuánticos

Los grupos cuánticos son estructuras matemáticas que amplían el concepto clásico de simetría (entiéndase como una transformación que preserva ciertas propiedades esenciales de un sistema, como lo puede ser una figura geométrica, una función o una estructura algebraica) y lo adaptan al contexto de la física cuántica [27] [28].

Aunque recibe el nombre de “grupos”, en realidad tiene una mayor relación con objetos algebraicos llamados “álgebras de Lie”, que es una estructura matemática para el estudio de simetrías continuas, que se define como un espacio vectorial compuesto de una operación llamada conmutador o corchete de Lie, es decir, mide como el orden de las operaciones o elementos influye en el resultado [29] [30].

Por lo anterior, se puede entender a los grupos cuánticos, como una deformación que adapta ciertas operaciones de las álgebras de Lie, que dan lugar a propiedades más adecuadas en la descripción de fenómenos cuánticos, siendo de nuestro interés [31]:

- **No Conmutatividad:** El orden en que se combinan los factores altera el resultado, es decir $ab \neq ba$.
- **No Coconmutatividad:** Podemos entender esta coálgebra como una división o descomposición con propiedad no conmutativa.

Estas características convierten a los Grupos Cuánticos en herramientas ideales para modelar situaciones en las que una transformación no se comporta de manera predecible, como sucede en diferentes modelos de física teórica.

Las Álgebras de Hopf, por su parte, integra operaciones análogas a la suma, el producto y la división, tal que, estas operaciones básicas sirven para expresar procesos sobre la combinación de elementos, su separación y la inversión de los mismos, lo que permite describir comportamientos característicos provenientes de los estados cuánticos [1] [31].

Esta estructura algebraica a su vez, da paso a una formulación más general en el marco de los Grupos Cuánticos, las conocidas como Categorías Trenzadas, que permiten la organización de transformaciones particulares, en donde, el intercambio entre dos objetos no es trivial. Es decir, si se intercambian dos elementos, el resultado puede ser diferente que si se los intercambia en sentido inverso [1] [31] [32].

3.1.1. Algebras de Hopf

Las Álgebras de Hopf se entiende como estructura central en la teoría de los Grupos Cuánticos, las cuales proporcionan explicaciones sobre las simetrías en el contexto no convencional y no conmutativo, ya que permiten unificar operaciones tanto de álgebras como de las coálgebras en un mismo espacio algebraico [1] [31].

Entendemos por Álgebra en Grupos Cuánticos como una estructura que actúa sobre un espacio vectorial con una operación de producto bilineal (es decir, se respetan las reglas de la suma y la multiplicación por escalares en cada uno de sus argumentos), que tienen la propiedad de combinar dos elementos para obtener un tercero dentro del mismo espacio. Describiendo así, un modelo de “descomposición” cuántica [32].

Por otro lado, la Coálgebra se puede entender como una estructura dual o inversa de este tipo de álgebra. En el que, al contrario de combinar elementos, este describe como un elemento puede ramificarse o dividirse en diferentes partes. Siendo la coálgebra una modelación de “descomposición” en el contexto de comportamiento cuántico [1] [31].

Esta combinación de Álgebra y Coálgebra es la base para representar las formulaciones matemáticas de las estructuras de Grupos Cuánticos, describiendo las interacciones entre estados cuánticos y sus posibles combinaciones y descomposiciones [1] [31].

En términos formales, las Álgebras de Hopf se definen como un espacio vectorial H sobre un cuerpo base $\mathbb{K}$, es decir, un conjunto numérico dotado de operaciones de suma, multiplicación y sus inversos (con excepción del cero). Tal que, $\mathbb{K}$ pertenece a un conjunto [31] [32]:

- $\mathbb{R}$: el conjunto de los números reales.
- $\mathbb{C}$: el conjunto de los números complejos.

Dentro del marco completo de las Álgebras de Hopf, existen operaciones conocidas como Unidad y Counidad, las cuales permiten, la introducción y eliminación coherente de elementos neutros dentro de una estructura. Sin embargo, no se hace uso de estas operaciones dentro de la presente propuesta, ya que el objetivo no es preservar elementos invariantes.

A continuación, se describen las cuatro operaciones fundamentales de las Álgebras de Hopf, de las que se hace uso en esta investigación:

3.1.1.1. Producto

En el Álgebra de Hopf, el Producto es una operación bilineal, esto implica que cada uno de sus elementos por separado son lineales. Esta propiedad es esencial en espacios vectoriales, donde una función se considera lineal si satisface dos condiciones básicas: la adición y la homogeneidad (escalamiento) [1] [31].

Formalmente, el producto se representa mediante el morfismo [32]:

$$m: H \otimes H \to H \tag{5}$$

Entonces, el producto (m) descrito por (5), implica que "H" es un espacio vectorial sobre un cuerpo $\mathbb{K}\,(\mathbb{C}\,o\,\mathbb{R})$, y "$\otimes$" denota el producto tensorial, que combina dos elementos para formar un par ordenado, generando así la composición "$\to$" de un nuevo valor H que pertenece al mismo cuerpo $\mathbb{K}$.

Lo anterior (12) implica que el producto respeta la bilinealidad en ambos argumentos, tal que [31]:

$$m(a \otimes b + a' \otimes b') = m(a \otimes b) + m(a' \otimes b') \quad (6)$$

$$m(\lambda a \otimes b) = \lambda m(a \otimes b) \quad (7)$$

En donde, (6) representa la propiedad de adición dentro de una operación Producto (m) con $a, a', b, b' \in H$. Mientras que (7) expresa la homogeneidad con $\lambda \in \mathbb{K}$. Además, dado un segundo argumento $c \in H$, también se cumple:

$$m(a \otimes (b + c)) = m(a \otimes b) + m(a \otimes c) \quad (8)$$

Esta propiedad (8) generaliza la distribución y linealidad observadas en la multiplicación algebraica convencional.

3.1.1.2. Coproducto

Es una operación dual al Producto (m) y se considera una coálgebra que integra el Álgebra de Hopf. Mientras que el producto "combina", el coproducto "duplica". Su valor puede interpretarse como una forma de "descomposición" de un elemento de H en partes más simples. Se define a este morfismo como [1] [31]:

$$\Delta: H \to H \otimes H \quad (9)$$

Para (9), el Coproducto (Δ) implica tomar un elemento H y realizar una descomposición mediante una combinación de pares tensoriales. De igual forma, H pertenece a un cuerpo base $\mathbb{K}$ del conjunto $\mathbb{C}$ o $\mathbb{R}$, con $\otimes$ que denota producto tensorial.

A su vez, una propiedad fundamental que debe cumplir el Coproducto (Δ) es la coasociatividad, la cual garantiza consistencia en descomposiciones múltiples dentro del espacio H. Esto se expresa mediante la siguiente igualdad [31]:

$$(\Delta \otimes id) \circ \Delta = \Delta \circ (id \otimes \Delta) \quad (10)$$

En donde, id en (10) representa la aplicación identidad, es decir, un morfismo que actúa como función neutra, sin alterar el valor del elemento al que se aplica. De esta forma, la coasociatividad asegura una descomposición estable (el resultado no

depende del orden en que se agrupan las operaciones) y estructurada (definida por la operación Coproducto Δ) de los elementos H [1] [32].

3.1.1.3. Antípoda

La Antípoda (S), es una operación fundamental que introduce el concepto de inverso de un grupo en la teoría de Álgebras de Hopf. Siendo una extensión en donde las operaciones de Producto y Coproducto coexisten, y su papel es garantizar una simetría entre ambos, haciendo posible una "inversión estructurada" [1].

Su morfismo se define como [31]:

$$S: H \to H \tag{11}$$

(11) se puede entender como un reflejo algebraico, que permite "revertir" una combinación de elementos. En consecuente, la Antípoda debe cumplir con una propiedad de compatibilidad, tal que [31]:

$$m \circ (S \otimes id) \circ \Delta(h) = \eta \circ \varepsilon(h) = m \circ (id \otimes S) \circ \Delta(h) \tag{12}$$

Donde:

- $h \in H$
- m representa el producto.
- Δ es el coproducto.
- η es la unidad.
- ε es la counidad.
- id es el morfismo identidad.

(12) se denota como una composición de funciones, similar a la representación en el cálculo diferencial $f \circ g(x) = f(g(x))$, permitiendo una expresión compacta del encadenamiento de operaciones.

Lo anterior, expresa también, que la ecuación (12) es análoga a la relación: $a^{-1} * a = e$, en donde el producto con el inverso devuelve la identidad (e) [32].

3.1.1.4. Coantípoda

De forma similar, en la coálgebra, puede entenderse a la Coantípoda (S^*) como el equivalente de la Antípoda (S). Esta operación se puede entender como una aplicación que inicia con un elemento, lo descompone en partes a través del Coproducto (Δ) y posteriormente lo combina mediante el Producto (m).

De esta forma, mientras que la Antípoda se relaciona con revertir el efecto del Producto, la Coantípoda aparece como una estructura para revertir o contrarrestar el Coproducto [1] [33].

Se puede expresar el morfismo de Coantípoda de la forma [33]:

$$S^*: H^* \to H^* \tag{13}$$

La expresión (13), considera que esta operación actúa sobre el espacio dual H^*, manteniendo la coherencia estructural dentro de la coálgebra de Hopf. Similar a la Antípoda en (11), siendo un reflejo algebraico, pero en este caso se tiene la finalidad de "revertir" una división de elementos.

En contraste a la Antípoda, la Coantípoda implica la propiedad fundamental de compatibilidad, mediante [31]:

$$\Delta \circ S^*(h) = (S^* \otimes S^*) \circ \Delta(h) \tag{14}$$

(14) indica que da lo mismo aplicar la Coantípoda antes del Coproducto, que aplicar el Coproducto antes de la Coantípoda, a los elementos duales.

Aunque no todas las Álgebras de Hopf requieren definir una Coantípoda explícita, en algunas formulaciones se utiliza para extender la simetría de manera más profunda, y es útil para ilustrar la dualidad del comportamiento inverso respecto a la Antípoda [33].

En forma de resumen, se puede identificar:

Concepto	**Antípoda**	**Coantípoda**
Actúa Sobre	El producto y se comporta como un inverso multiplicativo.	El coproducto y se interpreta como una especie de inverso con respecto a la descomposición.

Analogía	Invierte el flujo de composición (como en revertir una función o un producto)	Invierte el flujo de descomposición.
Ejemplo	Dos elementos se combinan y luego se descomponen en dos nuevos estados.	Un elemento se descompone y luego sus partes se vuelven a unir en una nueva entidad.

Tabla 7. Diferencia entre Antípoda y Coantípoda.

3.1.2. Categorías Trenzadas

Una Categoría en este contexto, es una estructura que organiza objetos y las relaciones entre ellos, conocidos como morfismos. Estos pueden representar distintos elementos: desde conjuntos y espacios vectoriales, hasta estructuras algebraicas más complejas [34]. Por otra parte, en la física y matemáticas cuánticas, el término de "Trenzar" se refiere al comportamiento de cambiar elementos (como partículas, operadores o estados) [1].

Surgiendo las Categorías Trenzadas como estructuras algebraicas que incorporan una regla particular para intercambiar dos objetos, mediante una transformación llamada "Trenzado". Esta interacción no corresponde a una simple permutación, sino a un entrelazamiento algebraico que modifica simultáneamente a los elementos involucrados [31].

Formalmente, el Operador de Trenzado (Ψ) se expresa mediante el morfismo [1]:

$$\Psi: A \otimes A \to A \otimes A \tag{15}$$

Para (15), A es un álgebra sobre un cuerpo $\mathbb{K}$ (como $\mathbb{C}$ o $\mathbb{R}$), y el tensor $\otimes$ señala que estamos considerando productos combinados de elementos. Para comprender mejor este morfismo se puede simplificar como [31] [32]:

$$\Psi(a \otimes b) = b' \otimes a' \tag{16}$$

(16) indica que, al aplicar el Trenzado (Ψ) sobre dos elementos $a \otimes b$, se produce un intercambio de posiciones acompañado por una posible transformación interna de dichos elementos, resultando en $b' \otimes a'$, que no son necesariamente igual a los elementos originales (a y b), lo cual refleja la asimetría inducida por el Trenzado.

3.2. Estructuras Algebraicas Aplicadas

Este apartado pretende establecer la transición formal de las estructuras cuánticas consideradas (el Trenzado y la Coantípoda), exponiendo las herramientas algebraicas necesarias para el diseño de una implementación funcional reversible, eficiente y segura.

3.2.1. Diagramas para el Modelado de Operadores

En el campo de los Grupos Cuánticos, Shahn Majid ha realizado trabajos sobre la interpretación de Categorías Trenzadas mediante el uso de Diagramáticas. En lugar de limitarse al uso de expresiones algebraicas abstractas, propone representaciones visuales mediante diagramas de cuerdas, que capturan el flujo, la simetría y la dirección de las transformaciones en una categoría trenzada [1].

Estas representaciones visuales, que funcionan como cuerdas o caminos, permiten ilustrar propiedades como la no conmutatividad, el flujo de información o su interacción algebraica, para así obtener una noción más natural de estas estructuras, lo que también facilita el entendimiento de conceptos físicos como el entrelazamiento y su reversibilidad.

Así, estas representaciones actúan como puentes, que proporcionan un lenguaje visual riguroso y flexible para describir comportamientos cuánticos sin recurrir necesariamente al formalismo físico. Siendo una herramienta más que ilustrativa, de la cual se puede hacer uso al expresar operadores mediante grafos dirigidos o transformaciones simbólicas, en una posible traducción de rutinas computacionales discretas.

3.2.1.1. Diagramática Propuesta – Coantípoda

A continuación, se muestra una tabla que describe el morfismo, su abstracción diagramática y por último la interpretación en relación a la transformación de datos desde una perspectiva de computación en relación al cifrado.

Morfismo	Abstracción	Interpretación
$S^*: H^* \to H^*$		Se puede entender a este proceso como una transformación de dos fases, la cual consiste en una expansión y su posterior reducción, que funciona como una operación de generación caótica de valores.
La Coantípoda se puede entender como una aplicación que inicia con un elemento, lo descompone en partes a través del Coproducto (Δ) y posteriormente lo combina mediante el Producto (m).		

Tabla 8. Diagramática Propuesta – Coantípoda.

Como se expresa en la Tabla.8., la Coantípoda es un proceso compuesto por dos operadores básico en la secuencia Coproducto – Producto, y que de forma individual se entienden de la siguiente forma:

Morfismo	Abstracción	Interpretación
$m: H \otimes H \to H$		Esta composición se puede ver como una reducción de datos en donde dos valores interactúan bajo una regla de combinación, resultando en un nuevo elemento.
El Producto consiste en una operación de composición entre dos elementos para obtener un tercero.		
$\Delta: H \to H \otimes H$		La descomposición representa una expansión de datos, en el que un dato se ve afectado para originar dos elementos que conservan su información estructural respecto al dato inicial.
El Coproducto es una descomposición, dado un elemento lo separa en dos.		

Tabla 9. Diagramática Propuesta – Coproducto/Producto.

La construcción de este enfoque no aplica un formalismo estricto de los Grupos Cuánticos, pero sí se conservan ideas metafóricas como “inverso cuántico” y se refleja una secuencia controlada de transformaciones que permiten tanto cifrar como descifrar.

3.2.1.2. Diagramática Propuesta – Trenzado

De forma similar a la Coantípoda, la siguiente tabla describe el morfismo, la abstracción diagramática y su interpretación respecto al Trenzado sobre el cifrado de datos.

Morfismo	Abstracción	Interpretación
$\Psi: A \otimes A \to A \otimes A$ $\Psi(a \otimes b) = b' \otimes a$		En este tipo de transformación los datos son reordenados mediante cruces que intercambian las posiciones de los datos contenidos, y a su vez, son alterados simulando el concepto de trenzado.
Su propósito es intercambiar o trenzar los elementos, puede involucrar factores adicionales, ajustando la posición de los elementos de forma no trivial.		

Tabla 10. Diagramática Propuesta – Trenzado.

3.2.2. Aritmética Modular

La aritmética modular es un sistema de cálculo en el cual se realizan operaciones bajo un sistema cíclico, similar al comportamiento de un reloj, en este sistema los números "vuelven al inicio" al alcanzar cierto valor de equivalencia llamado módulo. Este enfoque es de uso común en diversas ramas de la matemática aplicada, incluyendo criptografía, teoría de códigos, y estructuras algebraicas finitas [35] [36].

La operación módulo se define para a un número entero y n un número entero positivo, tal que, al dividir a entre n se obtiene un residuo r, y cumple con la relación [3]:

$$a \ mod \ n = r \ \ donde \ \ a = qn + r, con \ 0 \leq r < n \tag{17}$$

- Con q el cociente entero.

Asimismo, se dice que dos números enteros a y b son "congruentes módulo n" si tienen el mismo residuo al dividirse entre n. Se expresa [3] [37]:

$$a \equiv b (mod \ n) \tag{18}$$

Estas congruencias (18) dan lugar a las siguientes propiedades [3] [35]:

1) Reflexiva – Todo número es congruente consigo mismo: $a \equiv a(mod\ n)$

2) Simétrica – Si $a \equiv b(mod\ n)$, entonces: $b \equiv a(mod\ n)$

3) Transitiva – Si $a \equiv b(mod\ n)$ y $b \equiv c(mod\ n)$, entonces: $a \equiv c(mod\ n)$

Por las propiedades anteriores, se pueden definir operaciones modulares fundamentales [35] [38]:

- Suma: $[(a\ mod\ n)\ +\ (b\ mod\ n)]\ mod\ n\ =\ (a\ +\ b)\ mod\ n$
- Resta: $[(a\ mod\ n) -\ (b\ mod\ n)]\ mod\ n\ =\ (a -\ b)\ mod\ n$
- Multiplicación: $[(a\ mod\ n) *\ (b\ mod\ n)]\ mod\ n\ =\ (a *\ b)\ mod\ n$

La aritmética modular es una herramienta fundamental en criptografía y en sistemas de seguridad informática porque proporciona un marco matemático para trabajar con números bajo una operación de módulo, que limita los valores dentro de un rango fijo.

3.2.3. Grupos, Anillos y Campos

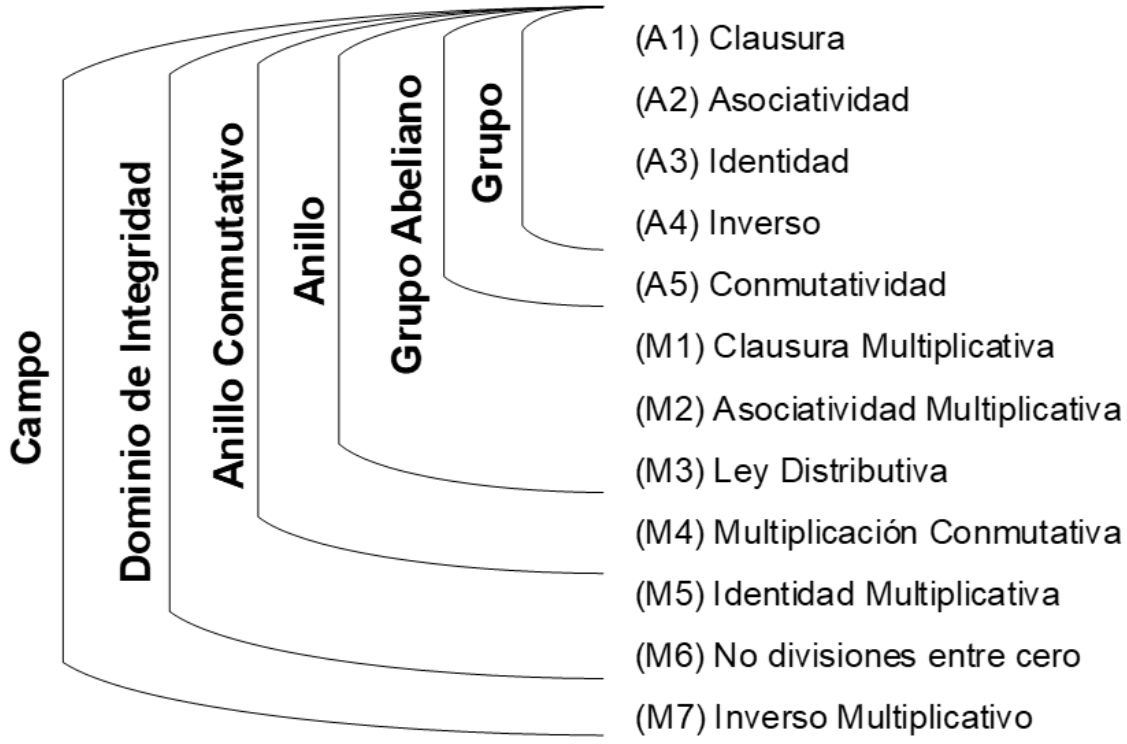


Fig. 4. Jerarquía de estructuras algebraicas: grupo, anillo y campo. Adoptado de *[3]*

Los Grupos, Anillos y Campos constituyen una estructura fundamental en el área de la matemática conocida como álgebra abstracta o moderna. Este tipo de álgebra abstracta, debe cumplir con ciertas propiedades como lo es la asociatividad, la existencia de neutros y en algunos casos sus inversos [39].

En comparación a la aritmética elemental, las operaciones que se emplean no son únicamente numéricas, sino que se generalizan dichas operaciones a estructuras más complejas, en las que se incluyen matrices, polinomios, funciones y otros objetos matemáticos, por convención, las dos operaciones principales se relacionan a la suma $(+)$ y multiplicación $(*)$ [27] [40].

A continuación, se aborda una breve descripción de las tres estructuras y las propiedades asociadas a la relación jerárquica entre Grupos, Anillos y Campos Fig.4.

3.2.3.1. Grupos

Un Grupo (G), es comúnmente denotado por $\{G,*\}$, es una operación binaria que denota · que para cada par ordenado (a, b) de elementos $(a * b)$, se cumple [3]:

(A1) Clausura: Si a y b pertenecen a G, entonces $a * b$ también está en G.

(A2) Asociatividad: $a * (b * c) = (a * b) * c$ para toda a, b, c en G.

(A3) Identidad: Existe un elemento en G tal que $a * e = e * a = a$ para toda a en G.

(A4) Inverso: Para cada a en G, hay un elemento a' en G, tal que $a * a' = a' * a = e$.

Si el Grupo (G) está formado por un número finito de elementos, se denomina como "Grupo Finito", por el contrario, se trata de un "Grupo Infinito".

A su vez, un Grupo se considera "Abeliano" si cumple:

(A5) Conmutatividad: $a * b = a * b$ para toda $a, b\ en\ G$.

Además, se implica que para todos los elementos enteros tanto positivos y negativos, bajo la operación de adición y multiplicación (números reales distintos de cero) son un Grupo Abeliano.

Algunas características especiales son:

- Grupo Aditivo: Identidad 0, con inverso $-a$, la resta se define $a - b = a + (-b)$.
- Grupo Multiplicativo: Identidad 1, inverso a^{-1}.
- Grupo Cíclico: Se define como una operación exponencial, de modo que para $g \in G$, se tiene g^n con n un número entero, elemento neutro $a^0 = e$ y $a^{-n} = (a')^n$ su inverso. Este Grupo siempre es Abeliano y puede ser Finito o Infinito.
-

3.2.3.2. Anillos

Un Anillo (R), se denota como $\{R, +, \times\}$ es un conjunto de elementos con dos operaciones binarias: Adición y Multiplicación. De modo que R cumple [3]:

(A1 – A5): R es un Grupo Abeliano con respecto a la Adición, por ende, satisface las propiedades asociadas al mismo.

(M1) Clausura Multiplicativa: Si a y b existen en R, entonces ab también está en R.

(M2) Multiplicación Asociativa: $a(b * c) = (a * b)c$ para toda a, b, c en R.

(M3) Ley Distributiva: $a(b + c) = ab + ac, (a + b)c = ac + bc$ para toda a, b, c en R.

Un anillo es “Conmutativo” si cumple las siguientes condiciones [3]:

(M4) Multiplicación Conmutativa: $ab = ba$ para toda a, b en R.

Un “Dominio de Integridad” es un Anillo Conmutativo que además cumple [3]:

(M5) Identidad Multiplicativa: Existe 1 en R, tal que $a1 = 1a$ para todo a en R.

(M6) No Divisiones entre 0: Si a, b existen en R y $ab = 0$, entonces $a = 0$ ó $b = 0$.

3.2.3.3. Campo

Un Campo (F), a veces denotado $\{F, +, \times\}$ es un conjunto de elementos con operaciones binarias, la suma y la multiplicación, tal que a, b, c existen en F y cumple [3]:

(A1 – M6): F es un Dominio de Integridad que satisface las propiedades asociadas al mismo.

(M7) Inverso Multiplicativo: Para toda a que existe en F, excepto el 0, hay un elemento a en el conjunto F, tal que $a(a^{-1}) = (a^{-1})a = 1$.

A su vez, se pueden realizar operaciones de suma, resta y multiplicación sin salir del conjunto, y se define a la división como una regla: $a/b = a(b^{-1})$.

3.2.3.4. Campo Finito

En la sección anterior, se definió el conjunto de propiedades por los cuales se conforma un Campo (F). Si bien, los cuerpos infinitos no son de particular interés en este contexto criptográfico, los cuerpos finitos, por el contrario, desempeñan un papel crucial en muchos algoritmos [3].

La notación empleada para la construcción de Campos Finitos es comúnmente denotada mediante $\mathbb{F}, \mathbb{Z}$ o $\mathbb{GF}$ (Campo de Galois). Entonces, sea un número primo de orden p, definimos el Campo Finito $\mathbb{GF}(p)$, como el conjunto de enteros $\mathbb{Z}_p = \{0,1, \dots, p-1\}$, con operaciones aritméticas módulo p [3] [41].

De manera general, los Campos Finitos se pueden clasificar en dos tipos:

- De orden p primo $\mathbb{GF}(p)$.
- orden potencia primo $\mathbb{GF}(p^m)$, con m un número entero positivo.

3.3. Codificación

En la informática, la codificación se refiere al proceso de representar datos (como números, caracteres o secuencias de bits) en un formato adecuado que una computadora pueda interpretar, almacenar o transmitir eficientemente [3]. Es importante mencionar que, este proceso no tiene como objetivo reducir el tamaño de los datos (como en la compresión) ni proteger su contenido (como en la criptografía) [42].

Aunque existen diversas aplicaciones de la codificación, este trabajo considera específicamente el formato de texto, el cual puede dividirse en dos categorías principales [43]:

3.3.1. Codificación de Caracteres

Este tipo de codificación transforma caracteres humanos (lenguajes, alfabetos y símbolos) en representaciones binarias. Este proceso se lleva a cabo asignando un valor numérico a cada símbolo textual (letras, números, signos, etc.).

Siendo los estándares de codificación de interés:

#	Estándar	Descripción
1	ASCII (American Standard Code for Information Interchange) [44]	Usa 7 bits para representar 128 caracteres básicos (letras inglesas, dígitos, signos de puntuación y caracteres de control). Fue el primer estándar ampliamente adoptado para intercambio de texto.
2	Latin-1 (ISO/IEC 8859-1) [45]	Extiende ASCII a 8 bits, representando 256 caracteres. Incluye símbolos adicionales para lenguas de Europa Occidental (acentos, diéresis, eñes). Es común en sistemas antiguos y en textos en español.
3	Unicode (UTF-8) [46]	Es una codificación de longitud variable que usa entre 1 y 4 bytes para representar cualquier carácter Unicode. Compatible con ASCII, es el estándar dominante en Internet dando soporte a todos los idiomas.

Tabla 11. Estándares de Codificación de Caracteres.

3.3.2. Codificación de Transferencia

Es un método de transformación de datos binarios (como archivos cifrados, imágenes o certificados digitales) en texto plano mediante una representación de caracteres imprimibles. Es útil para transmitir esta información a través de canales que solo aceptan texto, como el correo electrónico, XML o formularios web.

#	Estándar	Descripción
1	Base64 [47]	Convierte datos binarios en texto ASCII usando 64 símbolos. Es muy usado en correo electrónico (MIME) y en JSON/XML para transmitir imágenes, claves o archivos en entornos que solo aceptan texto.
2	Quoted-Printable [48]	Es un método de codificación que representa caracteres no imprimibles o especiales en un formato seguro de texto ASCII. Se usa principalmente en correos electrónicos para garantizar la transferencia de caracteres internacionales.
3	UUEncode (Unix-to-Unix Encoding) [49]	Fue uno de los primeros estándares de transferencia de datos, usado en sistemas Unix para enviar archivos binarios a través de canales de solo texto. Convierte bloques de 3 bytes en 4 caracteres imprimibles.

Tabla 12. Estándares de Codificación de Transferencia.

3.3.3. Codificación Base64

Los sistemas criptográficos modernos, sean simétricos o asimétricos, producen flujos de bytes que, al ser impresos o transmitidos directamente, pueden generar errores de interpretación en los sistemas que esperan texto plano. Para abordar esta incompatibilidad, Base64 se emplea como una capa de representación que garantiza la integridad del contenido binario [3].

La codificación Base64 se desempeña dividiendo la secuencia de bytes en bloques de 3 bytes (24 bits), separándose en 4 grupos de 6 bits cada uno, estos nuevos grupos se traducen en un carácter dentro del conjunto propio de 64 símbolos. Según RFC 4648, la implementación estándar (MIME/Base64) compatible con Matlab, hace uso de la representación de los siguientes caracteres [47] [50]:

Rango	Elementos
0 a 25	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
25 a 51	a b c d e f g h i j k l m n o p q r s t u v w x y z
52 a 61	0 1 2 3 4 5 6 7 8 9
62 a 63	+ /

Tabla 13. Índice de Codificación – Base64.

4. Metodología

En este capítulo se describe la metodología seguida para el diseño y validación del modelo de cifrado propuesto. Se combinan los fundamentos teóricos de estructuras algebraicas cuánticas con el desarrollo de software, detallando a continuación las etapas planificadas, las técnicas de diseño utilizadas y los criterios de validación aplicados al prototipo experimental.

4.1. Enfoque General

El presente trabajo de tesis tiene un enfoque metodológico de tipo investigación aplicada y experimental, cuya intención principal es el diseño y validación del modelo criptográfico basado en operadores derivados de la teoría de Grupos Cuánticos. Para ello, se abordan 3 ejes fundamentales:

1) Formalización Matemática del Modelo Criptográfico

A través de la definición de las estructuras cuánticas seleccionadas (trenzado y coantípoda) y su abstracción inicial como un proceso de cifrado simétrico se busca generar un modelo que tenga un fundamento matemático claro y comprobable.

2) Modelación Funcional mediante Diagramas

El uso de diagramas permite establecer de manera sencilla las etapas que componen al sistema, así como la relación de entradas y salidas. Haciendo posible mediante este modelado gráfico la representación de los operadores y su interacción dentro del proceso de cifrado/descifrado

3) Implementación Experimental y Validación

El modelo producido y su formalización matemática se traducen en un prototipo computacional de prueba, desarrollado en un entorno básico sin interfaz gráfica, siendo la finalidad la aplicación de pruebas con el fin de determinar si el algoritmo cumple con los criterios de cifrado esperados.

En conjunto, este enfoque metodológico integra el diseño teórico, el modelado conceptual y la validación práctica que sustenta el algoritmo propuesto.

4.2. Etapas de Desarrollo

El proceso metodológico seguido en esta investigación se organiza en tres etapas principales:

1) Diseño

A partir de la abstracción propuesta mediante diagramáticas en la sección 3.2.1, sobre los operadores de Trenzado y Coantípoda, se elaboraron diagramas de flujo y de actividades. Estos diagramas describen de manera visual la interacción de cada operador y la secuencia completa del proceso de cifrado y descifrado.

2) Formalización Matemática

Mediante la definición de los fundamentos teóricos, abstracción y modelado gráfico, se establecieron las reglas y operaciones para guiar la transformación de los datos en el proceso de cifrado orientado en el entorno de computación convencional. Estas reglas adaptan técnicas matemáticas coherentes, verificables y compatibles dentro del campo de la criptografía.

3) Implementación y Validación Experimental

El modelo diseñado se implementó como un prototipo computacional desarrollado exclusivamente para la aplicación de pruebas de validación. Estas pruebas se fundamentaron en métricas criptográficas (aleatoriedad, distribución estadística y sensibilidad de llave) y comparativas frente a un estándar criptográfico ya establecido AES, permitiendo determinar la factibilidad y solidez del algoritmo planteado.

4.3. Técnicas de Diseño

Para la construcción del modelo funcional de cifrado se recurre al uso de técnicas de modelado propias del desarrollo de software (UML y de IEEE), las cuales constituyen una base conceptual fundamental para describir la arquitectura del sistema. Este enfoque permite establecer un puente entre la abstracción teórica inspirada en estructuras algebraicas no convencionales a una representación como proceso computacional [51] [52].

Por consiguiente, se hace uso de diagramas de modelado estandarizados, de los cuales se han seleccionado los siguientes:

- Diagrama de Actividades (UML): Describe el flujo de control y actividades dentro de un proceso, incluyendo decisiones, paralelismo y ciclos, útil para describir comportamientos funcionales complejos [53].
- Diagrama de Flujo de Datos (ISO/IEC/IEEE): Representa el movimiento y la transformación de datos en un sistema, modelando procesos, almacenes de datos, flujos y entidades externas, sin detallar la lógica interna [54].

Este conjunto de herramientas se emplea para describir el enfoque estructural propuesto:

- Diseño de Operadores Funcionales.
- Diseño Funcional del Algoritmo.

Posteriormente, se realiza la formalización matemática de cada operador, definiendo las operaciones concretas que regulan su comportamiento dentro del algoritmo. Este enfoque gradual, desde el modelado gráfico hasta la expresión matemática, facilita una implementación estructurada y coherente en el entorno de computación convencional.

4.4. Especificación de Requerimientos de Validación

Con el fin de garantizar que el modelo criptográfico propuesto cumpla con propiedades mínimas de seguridad y robustez, se definen los siguientes requerimientos de validación que guiarán el proceso experimental:

4.4.1. Aleatoriedad

Como antes se ha expresado en la sección 2.2.1, el conjunto de NIST STS 800-22 es una herramienta que brinda un conjunto de 15 pruebas relacionadas con la evaluación de aleatoriedad. Para los cuales, una validación se considera positiva cuando $p - value \geq 0.01$, valores debajo de este umbral o indicaciones explícitas "$Non - Random$", $p - value = -1.0$, indican que la secuencia no es aprobatoria [11].

En formalidad, la NIST STS 800-22 no establece un porcentaje fijo como umbral aprobatorio. No obstante, para los fines de este trabajo se proponen los siguientes rangos de validación:

- Menor al 70% de pruebas aprobadas: Secuencias de Cifrado no Aceptable.
- ≥ 70% de pruebas aprobadas: Secuencias de Cifrado Aceptable.
- 100% de pruebas aprobadas: Secuencias de Cifrado Ideales.

Además, aunque el conjunto incluye 15 pruebas, algunas son más críticas porque evalúan propiedades fundamentales de la aleatoriedad, por lo que se destacan [NIST]:

1) Monobit Test (frecuencia de 0s y 1s).
2) Runs Test (longitud de secuencias consecutivas).
3) Approximate Entropy Test (uniformidad en patrones locales).
4) Linear Complexity Test (grado de complejidad de la secuencia).

Siendo el objetivo ideal de esta validación de aleatoriedad sobre el cifrado propuesto, aprobar estas 4 pruebas críticas y aprobar con al menos un 70% del total de pruebas (≥10 de 15), lo que consideraremos un indicador mínimo de calidad sobre la generación de secuencias cifradas.

4.4.2. Distribución y Correlación

Estas pruebas son complementarias al NIST STS, la intención es incluir otro tipo de métricas que permitan evaluar de manera más amplia las propiedades estadísticas del criptograma. Estas pruebas se orientan al análisis sobre la distribución de símbolos y la correlación entre ellos, lo cual proporciona evidencia adicional sobre la calidad del cifrado.

Histogramas

El objetivo de este tipo de evaluación es verificar la distribución de frecuencias tal que sea lo más uniforme posible. Esto se traduce en, una distribución plana indica un alto nivel de aleatoriedad, lo cual es deseable en un sistema de cifrado [3] [55].

Los criterios de aprobación, son obtener un criptograma que alcance el rango completo de valores posibles 0 a 255 y una distribución de frecuencias equilibrada que no muestre sesgos de similitud con el histograma del texto original, siendo más una evaluación objetiva.

Conjunto de Métricas Complementarias

Métrica	Definición	Criterio de Aceptación
Entropía de Shannon	Medida de la incertidumbre promedio por símbolo; cuantifica cuán impredecible es la fuente.	• Ideal (Optimo) ≈ 1.0 • Aceptable ≥ 0.975 • Mínimo ≥ 0.94 • No aprobatorio < 0.94
Chi-Cuadrado	Contrasta la distribución observada de símbolos frente a la distribución uniforme esperada.	• $p - value > 0.05$
Índice de Coincidencia (IC)	Probabilidad de que dos símbolos tomados al azar sean iguales; informa sobre repetitividad y redundancia.	• Ideal (Optimo) ≈ 0.0039 • Aceptable ≤ 0.01 • No Deseado > 0.01
Correlación Pearson	Mide dependencia lineal entre valores contiguos (o entre posiciones separadas por un desplazamiento fijo) del criptograma.	• Ideal cercano a 0 • Aceptable ≤ 0.02

Tabla 14. Criterio de Evaluación – Métricas de Criptoanálisis Estadístico.

Los criterios de aceptación y los umbrales aplicados en Tabla.14., se adoptan como pautas prácticas para la evaluación experimental y están sustentados en la guía técnica de NIST para pruebas estadísticas de aleatoriedad y en trabajos metodológicos sobre evaluación de entropía y pruebas de bondad de ajuste.

Así mismo, para las métricas Chi-Cuadrado e IC, se considerará la cercanía del valor obtenido de la evaluación en caso de no ser aprobatorio, para determinar si el comportamiento resultante es aún lejano del umbral necesario.

4.4.3. Sensibilidad de Llave

Como se menciona en la sección 2.2.3, un principio fundamental en la criptografía simétrica es la alta sensibilidad sobre la llave, esto implica que un algoritmo seguro debe ser caótico ante el más mínimo cambio en alguno de los elementos que conforman la llave de cifrado, y se establece un porcentaje de similitud aproximado del 50% [16].

Para nuestro caso de análisis, esta prueba la consideraremos aprobatoria si se cumple con un porcentaje de variación cercano al establecido (≥45%) y optimo si obtenemos un valor igual o superior al umbral teórico del 50%.

4.5. Implementación del Prototipo

La implementación del modelo criptográfico se llevó a cabo mediante un prototipo funcional desarrollado en un entorno de programación de alto nivel (MATLAB). Este prototipo integra los módulos de cifrado y descifrado, organizados en una arquitectura modular que permiten invocar de manera independiente los distintos procesos de transformación de datos.

Siendo el objetivo central de este trabajo la validación teórica y experimental del modelo, el código completo del prototipo no se incluye en el cuerpo del documento. No obstante, con el propósito de mostrar la naturaleza computacional del diseño, se han incluido fragmentos representativos de la implementación en la sección de Anexos (B). Los cuales corresponden a:

- Operadores Trenzado y Coantípoda (núcleo funcional del cifrado).
- Generación y Renovación de Sub-llave (gestión de la llave simétrica).
- Hexpansión (soporte para la estructura de procesamiento).

Estos elementos reflejan los componentes esenciales del diseño criptográfico y se presentan en los Anexos con el propósito de ilustrar la traducción de los operadores matemáticos a rutinas computacionales concretas.

La inclusión parcial del código responde a un criterio de claridad y relevancia: el análisis metodológico y de resultados se centra en el funcionamiento del modelo y su validación experimental, aspectos que ya se documentan mediante diagramas, descripciones formales y métricas de evaluación.

Por lo tanto, disponer únicamente de fragmentos representativos permite complementar la discusión sin sobrecargar el cuerpo principal del documento con detalles de implementación que, aunque necesarios para la ejecución, resultan secundarios frente a los objetivos analíticos de esta investigación.

5. Diseño y Formalización Matemática

Esta fase metodológica establece el puente entre la abstracción matemática y su implementación práctica, a través del diseño estructurado de componentes funcionales que conforman el modelo criptográfico propuesto.

Asimismo, como parte de la formalización matemática del modelo, se adopta la siguiente notación para representar los distintos espacios y conjuntos involucrados en el procesamiento de datos:

- $\mathbb{Z}_{256}$: Espacio modular que representa al conjunto de 256 valores posibles, asociados a los caracteres definidos en la codificación Latin-1.
- $\mathbb{Z}_{16}$: Espacio modular empleado para representar y manipular los dígitos hexadecimales que componen cada carácter.
- $\mathbb{I}_4 = \{1,2,3,4\} \subset N^+$: Conjunto finito de índices enteros positivos, utilizados como referencias direccionales en operaciones de mapeo sobre matrices de dimensión 4×4.

5.1. Diseño Funcional del Modelo de Cifrado

5.1.1. Diseño de Operadores Funcionales

En este apartado se modelan de forma individual los operadores fundamentales para la construcción del algoritmo criptográfico. En esta propuesta, se consideran los Operadores relacionados con la Coantípoda y el Trenzado. El objetivo es alcanzar una compresión integral del comportamiento lógico y computacional de estos operadores, sirviendo como punto de partida del diseño general.

Como elemento clave, también se agrega el diseño del módulo de generación de sub-llave. Para ello, se describe el comportamiento y funcionamiento lógico sobre la transformación estructural de los datos. Con este fin, se hace uso de dos tipos de diagramas complementarios:

a) Diagrama de Actividades (UML)

b) Diagrama de Flujo de Datos

5.1.1.1. Diagramas – Operador de Trenzado

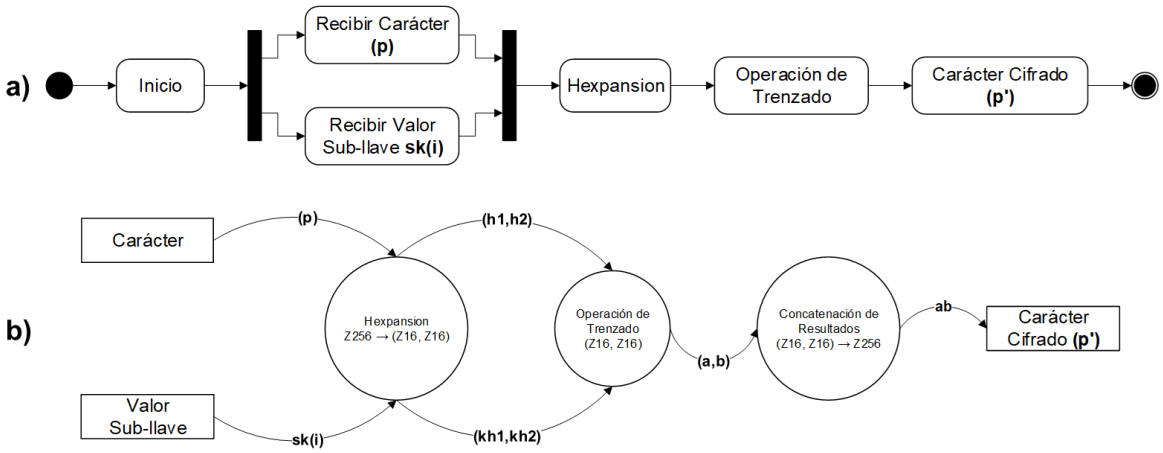


Fig. 5. Diagrama – Operador de Trenzado.

La Fig.5., representa el flujo funcional del Operador Trenzado. Este operador realiza una transformación estructurada a un carácter de entrada $p \in \mathbb{Z}_{256}$, haciendo uso de un elemento de la sub-llave $sk(i) \in \mathbb{Z}_{256}$, para generar un nuevo carácter cifrado $p' \in \mathbb{Z}_{256}$. Este procedimiento estructurado de funciones actúa sobre espacios modulares más pequeños, los cuales están contenidos en $\mathbb{Z}_{16}$.

El procedimiento es desarrollado de la siguiente manera:

1) Datos de Entrada: Se recibe un carácter del mensaje y un valor de sub-llave, ambos pertenecientes al espacio $\mathbb{Z}_{256}$.

2) Transformación Inicial – Hexpansion: A través de esta función, tanto el carácter p como la sub-llave $sk(i)$ se descomponen en sus dos dígitos hexadecimales, generando pares, de forma que, $p = (h_1, h_2)$ y $sk(i) = (kh_1, kh_2)$, acotando estos nuevos elementos individuales en el conjunto $\mathbb{Z}_{16}$.

3) Aplicación de operación Trenzado: Los valores expandidos se combinan mediante la operación de trenzado definida sobre el espacio modular $\mathbb{Z}_{16}$, que devuelven como resultado un par (a, b).

4) Reconstrucción del Carácter Cifrado: Se realiza una concatenación de los resultados a y b, devolviendo un valor $p' = ab \in \mathbb{Z}_{256}$, que representa el carácter cifrado.

5.1.1.2. Diagramas – Operador Coantípoda

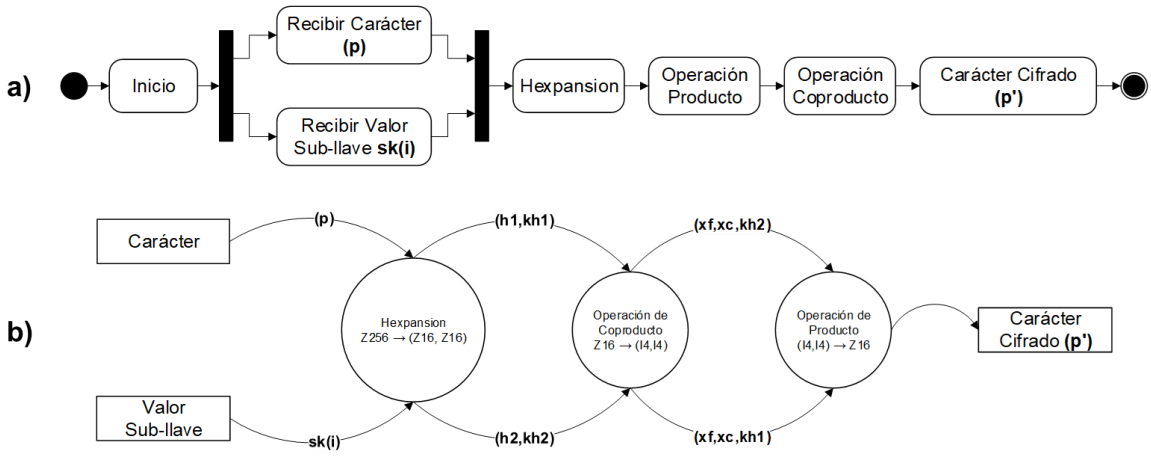


Fig. 6. Diagrama – Operador Coantípoda.

La Fig.6. de forma similar, al Trenzado (Fig.5), representa el flujo funcional de la transformación de un carácter (p), en conjunto a un elemento de la sub-llave $sk(i)$, para obtener un dato cifrado (p'). El proceso está organizado de la forma:

1) Entrada de Datos: Carácter p y valor sub-llave $sk(i)$, pertenecientes al conjunto $\mathbb{Z}_{256}$.

2) Transformación Inicial – Hexpansion: Tal como se presenta en el operador de trenzado, ambos valores son descompuestos en sus componentes hexadecimales. Tal que, $p = (h_1, h_2)$ y $sk(i) = (kh_1, kh_2)$, ambos en $\mathbb{Z}_{16}$.

3) Transformación – Coproducto: A diferencia del trenzado, este operador utiliza una estructura matricial para transformar datos. Se realiza un mapeo dentro de una matriz 4×4 para regresar los índices fila (x_f) y columna (x_c) donde se ubica el valor h buscado. Con $(x_f, x_c) \in \mathbb{I}_4 = \{1,2,3,4\}$, este procedimiento se lleva a cabo para cada (h_1, h_2).

4) Transformación – Producto: Esta operación similar al paso anterior, aplica una búsqueda, en este caso para obtener el valor que se encuentra en los índices de fila y columna de (x_f, x_c), recuperando un valor $\delta \in \mathbb{Z}_{16}$. Este proceso se aplica por cada par (x_f, x_c) obtenido de la transformación coproducto de (h_1, h_2), obteniendo como resultado el par (δ_1, δ_2).

5) Reconstrucción del Carácter Cifrado: Finalmente, se reconstruye el carácter cifrado p',realizando la concatenación del resultado obtenido en la transformación de Producto (δ_1, δ_2), conformando un elemento $p' \in \mathbb{Z}_{256}$

5.1.1.3. Generación de Sub-llave

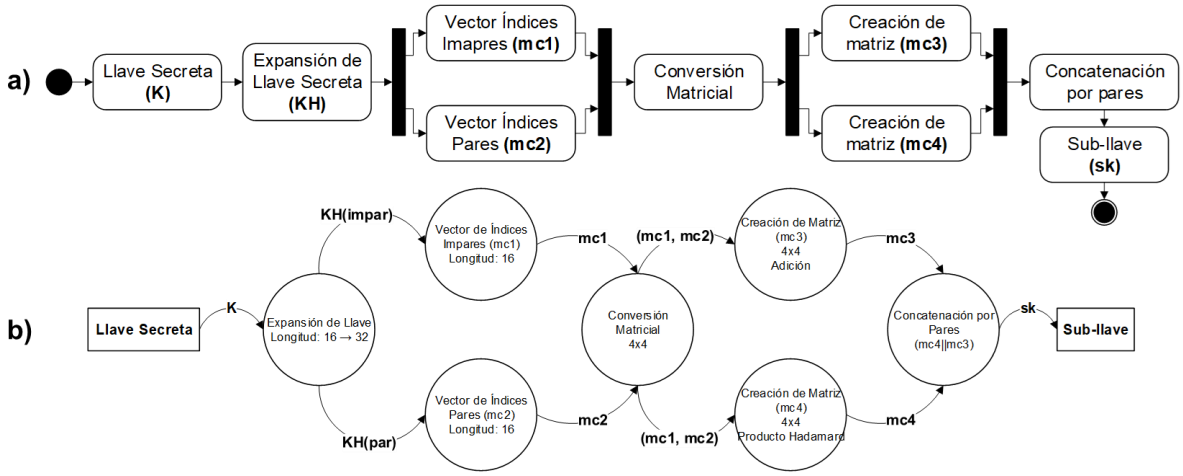


Fig. 7. Diagrama – Generación de Sub-llave.

La Fig.7., ilustra el procedimiento para la generación de la sub-llave sk, la cual es formada a partir de una llave secreta (formada por caracteres alfanuméricos) $K \in \mathbb{Z}_{256}$, de longitud 16. Este procedimiento permite derivar $sk \in \mathbb{Z}_{256}$, y de misma longitud, mediante manipulaciones de datos en el espacio $\mathbb{Z}_{16}$. El proceso es compuesto de los siguientes pasos:

1) Entrada de Datos: Se recibe una clave K constituida por 16 caracteres, cada uno perteneciente al espacio $\mathbb{Z}_{256}$.

2) Expansión de Llave: La llave es expandida a través de una transformación que convierte cada carácter en sus dos dígitos hexadecimales (Hexpansion), resultando en un vector KH de 32 elementos en $\mathbb{Z}_{16}$.

3) Separación por Índices:

El vector KH es separado en dos subconjuntos:

- m_{c1} = valores en posiciones impares (índices 1,3, 5, …,31)
- m_{c2} = valores en posiciones pares (índices 2,4, 6, …,32)

4) Conversión Matricial: Los vectores (m_{c1}, m_{c2}) se ordenan en una representación de matriz con dimensión 4×4.

5) Generación de Nuevas Matrices:

- m_{c3}=Resultado de aplicar módulo 16 a la suma entre (m_{c1}, m_{c2}).
- m_{c4}=Resultado de aplicar módulo 16 al producto Hadamard entre (m_{c1}, m_{c2}).

6) Generación de Sub-llave: Para finalizar, se realiza una concatenación por pares entre los elementos de ambas matrices mediante: $sk(i) = m_{c4}(i,j) \;||\; m_{c3}(i,j)$. Para todo $i, j \in \{1,2,3,4\}$, siguiendo el recorrido fila por fila. Lo que produce 16 elementos finales en $\mathbb{Z}_{256}$, conformando la sub-llave sk.

5.1.2. Diseño Funcional del Algoritmo Criptográfico

En este nivel de diseño, se integran los operadores previamente definidos para la construcción formal del algoritmo de cifrado y descifrado. De forma que, los diagramas presentados en esta sección describen el comportamiento del sistema y la interacción entre sus distintos componentes funcionales.

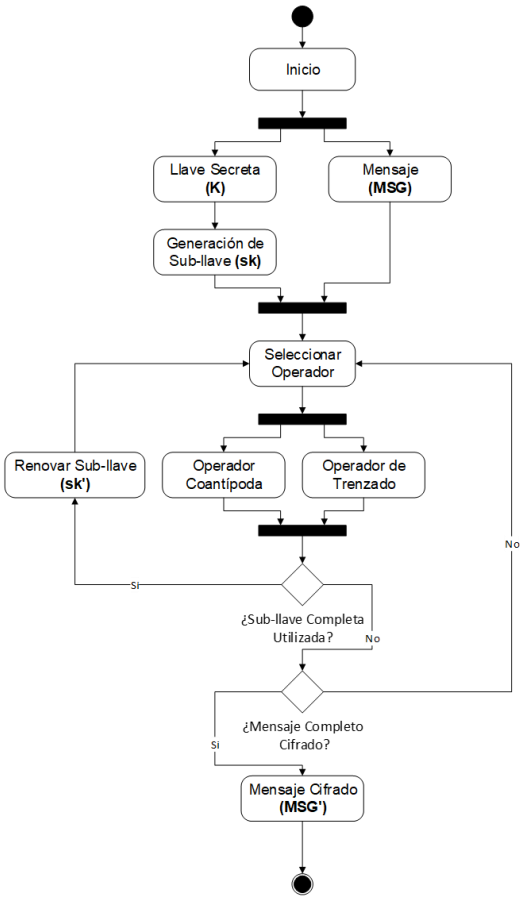


Fig. 8. Diagrama de Actividades – Cifrado/Descifrado.

La Figura Fig.8., representa un diagrama de actividades que modela el flujo funcional del sistema criptográfico, integrando los operadores previamente definidos y el procedimiento de generación de sub-llave. Interpretando el proceso como:

1) Entrada de Datos: Mensaje Original (MSG) y Llave Secreta (K).
2) Generación de Sub-llave: Se aplica el procedimiento correspondiente para obtener sk mediante K.
3) Proceso de Cifrado: Se aplica una selección para determinar el tipo de operador (Trenzado o Coantípoda) que transforma cada carácter correspondiente de MSG.
4) Renovar sub-llave: Si los 16 elementos de la sub-llave (sk) han sido utilizados, se recurre a una función para generar una nueva (sk').
5) Salida: Si se ha cifrado la totalidad del mensaje (MSG), se devuelve el mensaje cifrado (MSG')

El descifrado sigue la misma secuencia mostrada por Fig.8., pero proporcionando como entrada el Criptograma (MSG'), obteniendo como salida resultante a MSG.

5.2. Descripción General del Modelo

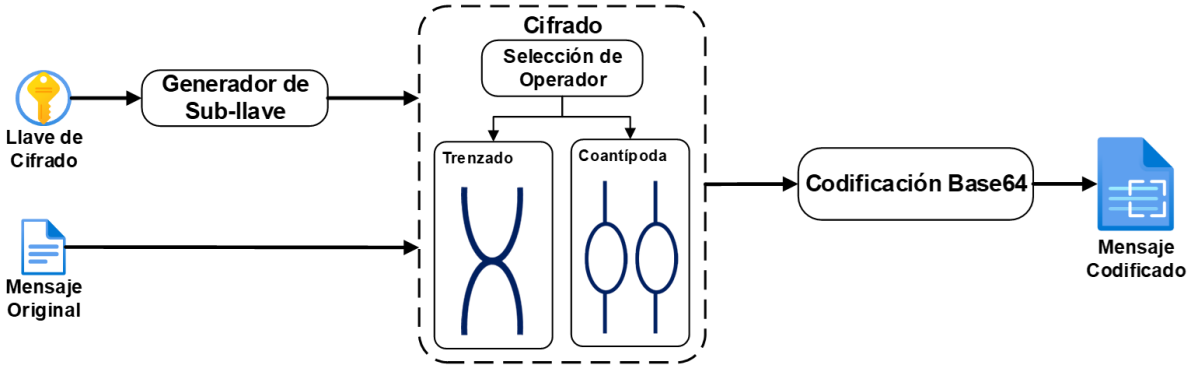


Fig. 9. Diagrama de Cifrado – Algoritmo Propuesto.

El diagrama Fig.9., ilustra el funcionamiento general del sistema de cifrado propuesto, el cual se basa en una arquitectura de estructuras algebraicas no conmutativas.

El proceso inicia con una llave de cifrado, la cual es procesada por un generador de sub-llaves que define los parámetros de operación interna. Posteriormente, el mensaje original y la sub-llave generada ingresan al módulo de cifrado, donde se selecciona

uno de los operadores definidos (Trenzado y Coantípoda) para transformar los datos de manera no lineal, introduciendo así propiedades de difusión y confusión.

Una vez obtenido el criptograma, se transforma mediante una etapa de codificación Base64, la idea es presentar los datos cifrados en un formato seguro y legible, siendo también una finalidad el almacenamiento o su portabilidad. Aunque este paso no conforma una parte interna del cifrado, se puede atribuir como una capa extra de postprocesamiento que facilita la manipulación del resultado.

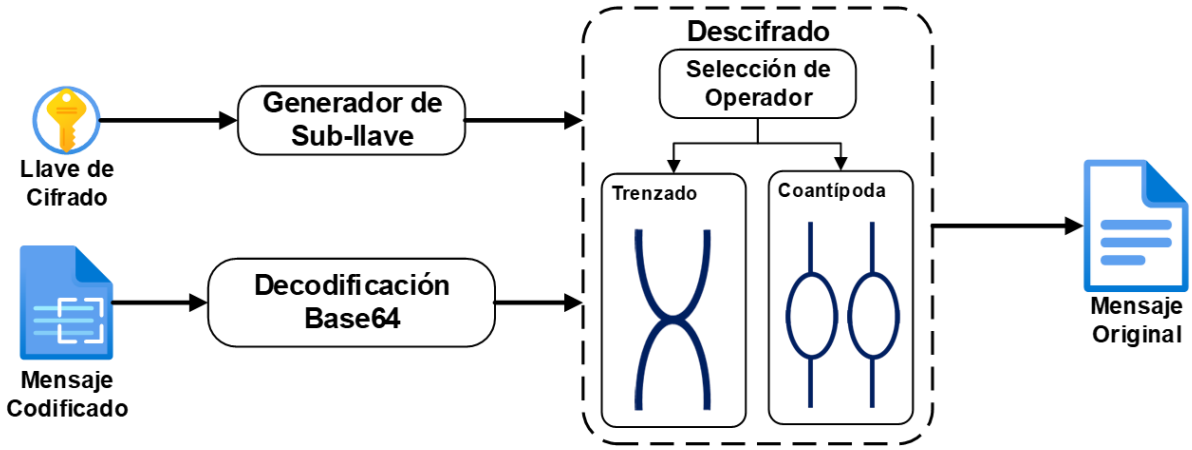


Fig. 10. Diagrama Descifrado – Algoritmo Propuesto.

El proceso de descifrado sigue exactamente los pasos inversos: primero se aplica la decodificación Base64 como etapa de preprocesamiento, recuperando el mensaje cifrado original. Luego, se utiliza la misma llave de cifrado para regenerar la sub-llave, asegurando la coherencia del proceso. Con dicha sub-llave y el mismo operador aplicado durante el cifrado, se ejecuta la operación inversa de forma controlada, restaurando el contenido original del mensaje.

Dado que los operadores definidos son reversibles y la estructura es simétrica, se garantiza que el mensaje original pueda ser recuperado sin pérdida.

5.3. Hexpansion

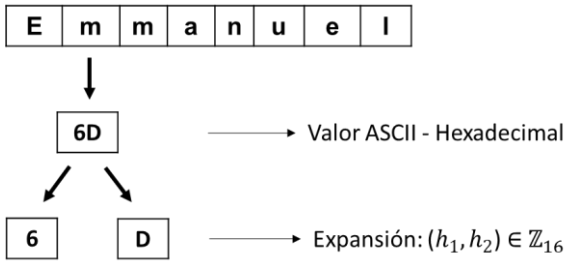


Fig. 11. Ejemplo – Expansión por Valor Hexadecimal de Carácter.

Previo a la aplicación de los operadores, cada valor de carácter p, que pertenece al espacio modular $\mathbb{Z}_{256}$, es transformado a su representación en notación hexadecimal. Esta conversión permite descomponer cada valor en dos dígitos hexadecimales independientes h_1 y h_2, restringidos al conjunto $\mathbb{Z}_{16}$.

$$p \rightarrow (h_1, h_2) \in \mathbb{Z}_{16} \tag{19}$$

De forma idéntica, este proceso se aplica a la sub-llave (sk), en donde cada elemento requerido $sk(i)$, con i la posición actual del recorrido de la llave, es expandido:

$$sk(i) \rightarrow (kh_1, kh_2) \in \mathbb{Z}_{16} \tag{20}$$

5.4. Selección de Operadores

La selección del operador cuántico a aplicar se realiza mediante una función módulo sobre el valor actual de la sub-llave:

$$selección = sk(i)\ mod\ 2 \tag{21}$$

Donde:

- $sk(i)$ es el valor de la sub-llave en la posición i.
- Si $selección = 0$, se aplica el Operador de Trenzado.
- Si $selección = 1$, se aplica el Operador Coantípoda.

Este mecanismo establece una lógica binaria que alterna entre el trenzado y la coantípoda, generando una dinámica no lineal dentro del espacio de procesamiento de los datos cifrados.

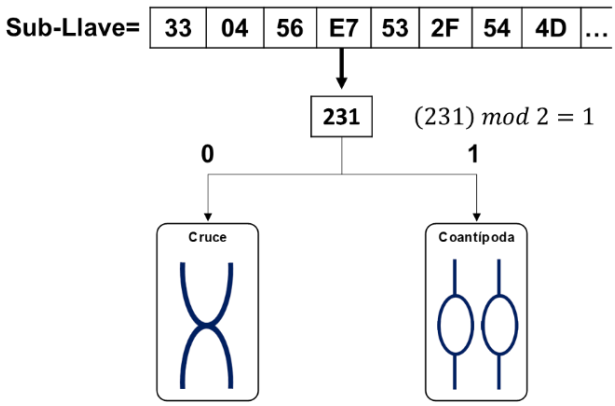


Fig. 12. Selección de Operador.

5.5. Módulo I: Sub-llave

La llave secreta o de cifrado, se sustenta como una entrada de longitud de 16 (caracteres), apegándose al estándar de seguridad definido por la Federal Information Processing Standard (FIPS 197) para la validación del uso de llaves de 128 bits (16 bytes) [2]. Como una práctica recomendada en el diseño criptográfico, se genera una sub-llave derivada de la llave principal, siendo una operación para fortalecer al algoritmo al introducir variabilidad y con ello, evitar patrones predecibles.

5.5.1. Generación de Sub-llave

Sea una llave $K = \{k_1, k_2, \dots, k_{16}\}$, con $k_i \in \mathbb{Z}_{256}$, se realiza una conversión de cada elemento en su representación hexadecimal de dos dígitos ($Hexpansion$), tal que, K se expande en $KH = \{kh_1, kh_2, \dots, kh_{32}\}\, con\, kh_i \in \mathbb{Z}_{16}$, obteniendo así un vector de longitud 32. A continuación, este vector se divide en dos partes:

El primer subvector (sk_1) contiene los elementos en las posiciones impares:

$$sk_1 = \{kh_1, kh_3, \dots kh_{31}\} \in \mathbb{Z}_{16}$$

El segundo subvector (sk_2) incluye los elementos en las posiciones pares:

$$sk_2 = \{kh_2, kh_4, \dots, kh_{32}\} \in \mathbb{Z}_{16}$$

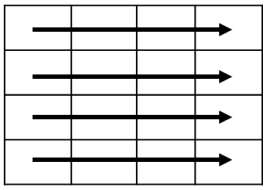
Ambos subconjuntos resultantes tienen longitud 16 y permiten construir las siguientes matrices:

Matriz $sk_1 \in \mathbb{Z}_{16}$: Se organiza como una representación de lectura de izquierda a derecha.

1) Se comienza desde la esquina superior izquierda.
2) Se avanza de izquierda a derecha a lo largo de la fila actual hasta alcanzar el final de la fila.
3) Al llegar al final de una fila, se baja a la primera columna de la siguiente fila y se repite el proceso hasta llenar la matriz.

$$sk_1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16] \longrightarrow Índices$$

$$Orden = \quad \longrightarrow \quad sk_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$



Matriz 4x4

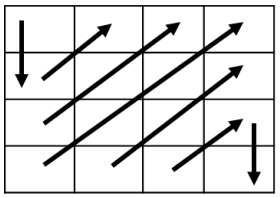
Fig. 13. Ordenamiento por Índices de Matriz sk_1.

Matriz $sk_2 \in \mathbb{Z}_{16}$: Se aplica un ordenamiento en zigzag para los elementos contenidos, partiendo de la primera columna, podemos definir un mapeo en términos de diagonales:

1) Se comienza desde la esquina superior izquierda.
2) Se avanza en diagonal hacia abajo hasta alcanzar el borde.
3) Se inicia una nueva diagonal desde la siguiente columna o fila disponible

$$sk_2 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16] \longrightarrow Índices$$

$$Orden = \quad \longrightarrow \quad sk_2 = \begin{bmatrix} 1 & 3 & 6 & 10 \\ 2 & 5 & 9 & 13 \\ 4 & 8 & 12 & 15 \\ 7 & 11 & 14 & 16 \end{bmatrix}$$



Matriz 4x4

Fig. 14. Ordenamiento por Índices de Matriz sk_2.

A partir de estas matrices se generan dos matrices derivadas:

- Matriz sk_3: $sk_3 = (sk_1 + sk_2)\ mod\ 16$ (22)

$$sk_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix} \qquad sk_2 = \begin{bmatrix} 0 & 2 & 5 & 9 \\ 1 & 4 & 8 & 12 \\ 3 & 7 & 11 & 14 \\ 6 & 10 & 13 & 15 \end{bmatrix}$$

$$sk_3 = (sk_1 + sk_2)\ mod\ 16$$

$$(12 + 6)\ mod\ 16 = \begin{array}{|c|c|c|c|} \hline 0 & 3 & 7 & 12 \\ \hline 5 & 9 & 14 & 3 \\ \hline 11 & 0 & 5 & 9 \\ \hline 2 & 7 & 11 & 14 \\ \hline \end{array} = (7 + 12)\ mod\ 16$$

Fig. 15. Ejemplo – Construcción de matriz sk_3.

- Matriz sk_4: $sk_4 = (sk_1 \odot sk_2)\ mod\ 16$ (23)

donde $\odot$ denota el producto elemento a elemento (Hadamard product) [56].

$$sk_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix} \qquad sk_2 = \begin{bmatrix} 0 & 2 & 5 & 9 \\ 1 & 4 & 8 & 12 \\ 3 & 7 & 11 & 14 \\ 6 & 10 & 13 & 15 \end{bmatrix}$$

$$sk_3 = (sk_1 \odot sk_2)\ mod\ 16$$

$$(12 \odot 6)\ mod\ 16 = \begin{array}{|c|c|c|c|} \hline 0 & 2 & 10 & 11 \\ \hline 4 & 4 & 0 & 15 \\ \hline 8 & 15 & 14 & 10 \\ \hline 8 & 2 & 6 & 1 \\ \hline \end{array} = (7 \odot 12)\ mod\ 16$$

Fig. 16. Ejemplo – Construcción de Matriz sk_4.

Finalmente, se genera la sub-llave de cifrado $sk \in \mathbb{Z}_{16}$ concatenando los elementos de ambas matrices por posición:

$$sk_i = (sk_4)_i \parallel (sk_3)_i, \qquad para\ i = 1, \dots, 16$$

$$sk_3 = \begin{bmatrix} 0 & 3 & 7 & 12 \\ 5 & 9 & 14 & 3 \\ 11 & 0 & 5 & 9 \\ 2 & 7 & 11 & 14 \end{bmatrix} = \begin{bmatrix} 0 & 3 & 7 & C \\ 5 & 9 & E & 3 \\ B & 0 & 5 & 9 \\ 2 & 7 & B & E \end{bmatrix} \quad sk_4 = \begin{bmatrix} 0 & 2 & 10 & 11 \\ 4 & 4 & 0 & 15 \\ 8 & 15 & 14 & 10 \\ 8 & 2 & 6 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & A & B \\ 4 & 4 & 0 & F \\ 8 & F & E & A \\ 8 & 2 & 6 & 1 \end{bmatrix}$$

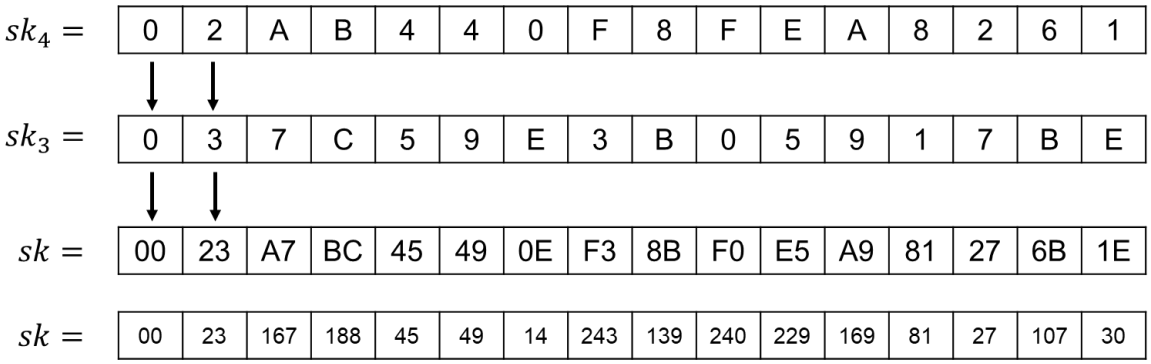


Fig. 17. Sub-llave sk longitud 16.

Este procedimiento proporciona un vector sub-llave (sk) de longitud 16 elementos, todos en $\mathbb{Z}_{256}$ que, al ser evaluado mediante una comparación de sensibilidad de llave, se obtuvo un porcentaje diferencial del 52% respecto de la llave inicial. Por lo anterior, se considera que el mecanismo de generación introduce una variación lo suficientemente significativa para incrementar la entropía y dificultar ataques basados en análisis de llave.

5.5.2. Renovación de Sub-llave

Una vez que la sub-llave ha sido completamente utilizada durante el procesamiento de cifrado, se ejecuta un procedimiento de renovación dinámica, con el objetivo de incrementar la variabilidad de las operaciones internas y minimizar patrones repetitivos que pudieran comprometer la seguridad del sistema.

Este procedimiento consiste en lo siguiente:

1) Se calcula la suma total de los elementos que conforman la sub-llave actual $sk \in \mathbb{Z}_{256}$, denotada por:

$$\sigma = \sum_{i=1}^{16} sk_i \tag{24}$$

2) A continuación, se genera una nueva versión de la sub-llave sk', sumando cada uno de sus elementos por el valor de σ, y aplicando módulo 256:

$$sk'_j = (\sigma + sk_j)\ mod\ 256,\ \forall_j = 1, \ldots, 16 \tag{25}$$

3) La sub-llave renovada sk' reemplaza a la sub-llave anterior y se utiliza de manera continua, siendo actualizada cada vez que se consume. Este proceso se repite de forma iterativa hasta completar el cifrado de la totalidad del mensaje.

Este enfoque proporciona una actualización autónoma y continua del material criptográfico, sin necesidad de recurrir a nuevas llaves externas. Así, se promueve una estructura de variabilidad interna, con sub-llaves que se adaptan y evolucionan a lo largo del cifrado.

5.6. Matrices de Cifrado/Descifrado

En el contexto de este algoritmo, las matrices de cifrado desempeñan un papel fundamental como transformaciones estructurales que permiten reorganizar los elementos sin repetir posiciones, que actúan como permutaciones del conjunto $\{0, 1, \ldots, 15\}$, reorganizando la posición de los elementos en una matriz de 4×4.

$$m_{c1} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix} \quad m_{c2} = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \end{bmatrix} \quad m_{c3} = \begin{bmatrix} 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 \end{bmatrix} \quad m_{c4} = \begin{bmatrix} 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \\ 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \end{bmatrix}$$

$$m_{c5} = \begin{bmatrix} 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix} \quad m_{c6} = \begin{bmatrix} 10 & 11 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \\ 15 & 14 & 13 & 12 \end{bmatrix} \quad m_{c7} = \begin{bmatrix} 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{bmatrix} \quad m_{c8} = \begin{bmatrix} 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

Fig. 18. Matrices de Cifrado Elaboradas.

Una característica muy importante es que ningún valor ocupa la misma posición en más de una matriz. Esto quiere decir que, por ejemplo, si el valor 0 está en la posición $(1,1)$ en m_{c1}, no volverá a aparecer en esa misma posición en las demás matrices.

Así mismo, cada matriz representa una permutación del conjunto $\{0, 1, \ldots, 15\}$, es decir, no hay repeticiones dentro de una misma matriz. Solo cambia el orden, pero no el contenido. Este tipo de reorganización puede verse como una función que “baraja” las posiciones de forma controlada.

5.6.1. Construcción de Matrices

Definir un vector base: Comienza con un vector de 16 elementos, m_{c1} o m_{c2}, que contendrán números en un orden específico.

- $m_{c1} = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]$
- $m_{c2} = [3,2,1,0,15,14,13,12,11,10,9,8,7,6,5,4]$

$$m_{c1} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{bmatrix} m_{c2} = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \end{bmatrix}$$

Fig. 19. Matrices de Cifrado Base m_{c1} *y* m_{c2}.

Para cada nueva matriz m_c con índice $i >= 3$, se aplica la operación de suma $+4$ a cada elemento del vector base, seguida de la operación módulo 16 para mantener los valores dentro del rango $0\ a\ 15$.

$$m_{c(i)} = \left(m_{c(i-2)} + 4\right) mod\ 16 \quad (26)$$

Orden de llenado de las matrices:

- Todos los vectores resultantes se ordenan en la matriz de izquierda a derecha. Es decir, cada número del vector se coloca en las posiciones de la matriz siguiendo el orden de izquierda a derecha.
- Una vez que llenamos la primera fila, continuamos con la siguiente fila de la matriz, colocando los elementos del vector en el mismo orden, fila por fila, hasta completar la matriz de 4×4. (Fig.13.)

Ejemplo de Generación de Matrices:

Matriz de Cifrado m_{c3}:

$$m_{c3} = \left(m_{c(3-2)} + 4\right) mod\ 16 = (m_{c1} + 4)\ mod\ 16$$

$$m_{c3} = [4,5,6,7,8,9,10,11,12,13,14,15,0,1,2,3]$$

$$m_{c3} = \begin{bmatrix} 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

Fig. 20. Ejemplo – Generación de Matriz de Cifrado Impar m_{c3}.

Matriz de Cifrado m_{c4}:

$$m_{c4} = (m_{c2} + 4)\ mod\ 16 = \left(m_{c(4-2)} + 4\right)\ mod\ 16$$

$$m_{c4} = [7,6,5,4,3,2,1,0,15,14,13,12,11,10,9,8]$$

$$m_{c4} = \begin{bmatrix} 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \\ 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \end{bmatrix}$$

Fig. 21. Ejemplo – Generación de Matriz de Cifrado Par m_{c4}.

5.7. Módulo II: Operadores de Grupos Cuánticos

Este apartado tiene como objetivo formalizar la lógica computacional y las operaciones matemáticas que sustentan el modelo criptográfico propuesto. Partiendo de la abstracción inicial representada mediante diagramas de modelado, se establece aquí una descripción detallada sobre los operadores de Trenzado y Coantípoda.

Estos operadores se pueden entender como transformaciones aplicadas a través de conjuntos o espacios modulares de caracteres $\mathbb{Z}_{256}$, expansión hexadecimal $\mathbb{Z}_{16}$, y mapeos mediante índices del conjunto $\mathbb{I}_4 = \{1,2,3,4\}$. Contemplando a su vez, el proceso inverso, que permite garantizar la recuperación estructurada y reversible de la información en las distintas etapas del cifrado.

5.7.1. Operador de Trenzado

En el modelo propuesto del operador de Trenzado se aplica sobre los pares expandidos $(h_1, h_2) \in \mathbb{Z}_{16}$, provenientes de un valor de carácter p. Este operador no cambia el dominio de los elementos, pero los transforma mediante una operación aditiva modulada, que se expresa formalmente como:

$$a = (h_1 + kh_1)\ mod\ 16,\ \ b = (h_2 + kh_2)\ mod\ 16 \tag{27}$$

Con (kh_1, kh_2) valores expandidos de la llave $sk(i)$ en ese instante.

Esta transformación permite generar diversas configuraciones que simulan el entrelazamiento, entre ellas:

- Estado estático: a y b no se intercambian. Con $a = h_1$ y $b = h_2$.
- Estado intercambiado: $a \leftrightarrow b$. Tal que, $a = h_2$ y $b = h_1$.
- Estado mixto I: a se conserva, $b \rightarrow b + k$. Con $a = h_1$.
- Estado mixto II: b se conserva, $a \rightarrow a + k$. Con $b = h_2$.
- Estado mixto III: $a \leftrightarrow b$, y se aplica $+k$ a ambos.
- Estado desplazado: Sin intercambio, pero ambos elementos se transforman mediante $+k$.
- Con k un valor constante.

Finalmente, los valores a y b se concatenan para reconstruir un nuevo valor de carácter (p') en el campo original:

$$p' = a||b, \qquad donde\ a, b \in \mathbb{Z}_{16}$$

Dado que cada valor hexadecimal representa 4 bits, la concatenación de a y b (ambos en $\mathbb{Z}_{16}$) permite reconstruir un único valor en $\mathbb{Z}_{256}$.

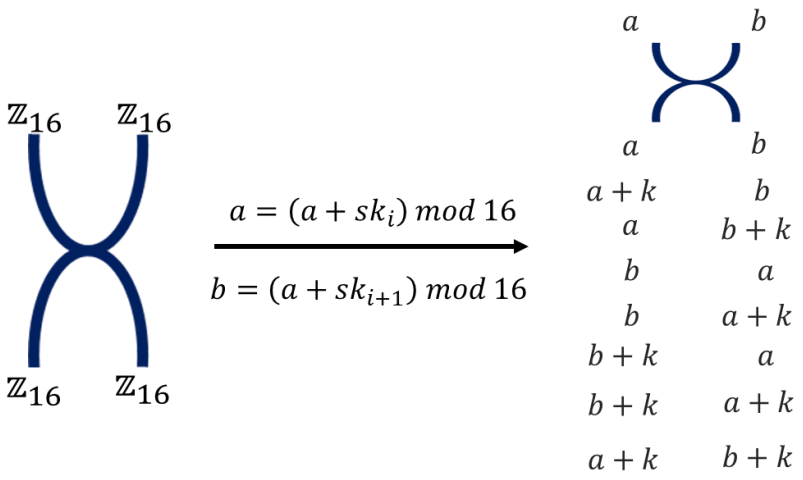


Fig. 22. Representación Diagramática – Operación de Cruce.

El proceso de Descifrado realiza la operación inversa a la adición positiva, sustituyendo las operaciones empleadas en (41), mediante:

$$a = (h_1 - kh_1)\ mod\ 16,\ \ b = (h_2 - kh_2)\ mod\ 16 \tag{28}$$

Esta operación introduce un cifrado no lineal apegándose a los conceptos clave de criptografía sobre la dispersión y confusión de los datos resultantes, así mismo, los posibles estados que pueden darse a través de esta aplicación emulan fenómenos de superposición y entrelazamiento dentro del marco de la mecánica cuántica, y así, mediante la transformación propuesta en conjunto al módulo de selección brinda un enfoque no convencional frente a los estándares actuales.

5.7.2. Operador Coantípoda

Como se muestra en la Fig.14., el proceso de Coantípoda se compone de las operaciones Coproducto → Producto, las cuales se aplican una por cada componente $(h_1, h_2) \in \mathbb{Z}_{16}$, permitiendo una expansión y la posterior reducción del dato original, lo que favorece la diversificación estructural de los valores intermedios. El proceso respectivo a este operador se divide entonces en dos fases:

Fase 1 – Descomposición (Coproducto)

Se aplica sobre un valor $\mathbb{Z}_{16}$, transformándolo en un par ordenado de índices ($\mathbb{I}_4$) provenientes de las matrices de cifrado/descifrado.

Para ejecutar dicha transformación, se recurre al conjunto de 8 matrices $m_c = \{m_1, m_2, \dots, m_8\}$, cada una de tamaño 4×4. Estas matrices contienen permutaciones únicas de los elementos de $\mathbb{Z}_{16}$, asegurando que cada número aparezca una sola vez por matriz. Su propósito es servir como mapeo determinista entre el valor original y su descomposición.

La selección de la matriz m_c usada en cada instancia está determinada por un valor de sub-llave $sk(i)$ en su representación de $Hexpansion$ (Dígitos Hexadecimales) como un par (kh_1, kh_2) y se calcula el índice mediante:

$$índice = (kh_1 \ mod \ 8) + 1, \ para \ índice \in \{1,2, \dots,8\} \tag{29}$$

Tal que:

$$m_{c(índice)}, \qquad para \ m_{c(índice)} \in \{m_{c1}, m_{c2}, \dots, m_{c8}\}$$

El $índice$ se reduce a módulo 8 con suma $+1$, para restringir el número de matrices disponibles. Una vez elegida la matriz, el operador actúa sobre un valor $h \in \mathbb{Z}_{16}$, localizando su posición dentro de la matriz M_c. Esta posición se expresa como un par de índices:

$$(x_f, x_c) = POS_{m_{c(indice)}}(h) \tag{30}$$

- $POS =$ Posición.
- $m_{c(indice)} =$ Matriz seleccionada.
- $h =$ Valor buscado.
- $x_f =$ Índice fila donde se ubica h.
- $x_c =$ Índice columna donde se ubica h.

$x_f \ y \ x_c$ son, respectivamente índices (fila y columna) donde se encuentra el valor h en m_c, y por lo tanto representan los componentes de la expansión del operador de creación en las matrices discretas de 4×4. Para h_1, se sustituye $h = h_1$ y para h_2, se sustituye $h = h_2$, $kh_1 = kh_2$.

Fase 2 – Recomposición (Producto)

En esta fase, se lleva a cabo una inversión del conjunto de índices obtenidos en la descomposición $\mathbb{I}_4 \rightarrow \mathbb{Z}_{16}$, replegando el par ordenado resultante en un único valor, siguiendo una estructura no lineal y gobernada por la simetría interna de las matrices m_c.

De forma análoga al proceso de descomposición, se selecciona otra matriz m_c mediante el cálculo del $índice$ (43), modificando el valor kh:

$$índice = (kh_2 \ mod \ 8) + 1, \ para \ índice \in \{1,2, \dots ,8\} \tag{31}$$

La operación consiste entonces, en acceder al valor ubicado en la posición (x_f, x_c) dentro de $m_{c(indice)}$, lo cual genera un nuevo valor dentro del conjunto $\mathbb{Z}_{16}$.

$$\delta = POS_{m_{c(indice)}}[x_f, x_c] \tag{32}$$

Donde,

- $POS =$ Posición.
- $m_{c(indice)} =$ Matriz seleccionada.
- $\delta =$ Valor encontrado.
- $x_f =$ Índice fila.
- $x_c =$ Índice columna.

Este procedimiento se aplica de forma individual a cada uno de los dos dígitos provenientes de la expansión hexadecimal del carácter original. Es decir, a partir del carácter $p \in \mathbb{Z}_{256}$, que fue descompuesto en dos valores $h_1, h_2 \in \mathbb{Z}_{16}$, y estos posteriormente mapeados a pares (x_f, x_c), ahora se recuperan dos nuevos valores $\delta_1, \delta_2 \in \mathbb{Z}_{16}$.

- Para obtener δ_1, se toma el par (x_f, x_c) resultante de la descomposición mediante h_1.
- Para obtener δ_2, se toma el par (x_f, x_c) resultante de la descomposición mediante h_2, y sustituyendo en la recomposición a $kh_2 = kh_1$.

Finalmente, para regresar al conjunto original $\mathbb{Z}_{256}$, se realiza una concatenación interpretando δ_1 como el dígito más significativo y δ_2 como el menos significativo:

$$p' = \delta_1 || \delta_2, \ donde \ \delta_1, \delta_2 \in \mathbb{Z}_{16} \tag{33}$$

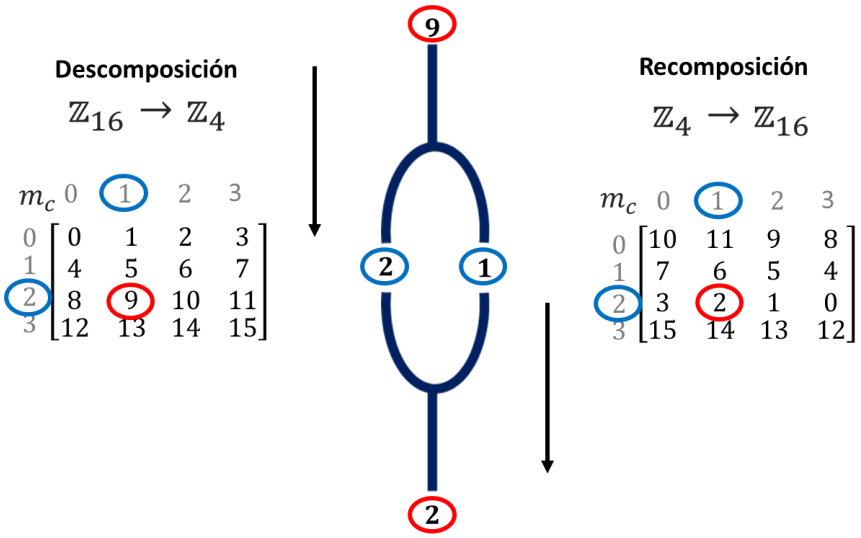


Fig. 23. Representación Diagramática – Operación Coantípoda.

El descifrado se puede ver como una inversión de la diagramática sugerida Fig.23. que sigue la misma secuencia Coproducto → Producto, pero en el que ahora se ven intercambiados los elementos (h_1, h_2), de forma que el recorrido que sigue h_1 ahora es realizado por h_2 y de igual manera h_2 sustituye el proceso de h_1.

La simulación de este proceso natural de descomposición y recomposición, es determinista, dependiente de sub-llave y reversible, lo cual lo convierte en una analogía funcional de la coantípoda algebraica. Y similar al cruce se pueden apreciar los fenómenos de superposición y entrelazamiento cuántico, además el uso de matrices para la transformación se aleja del enfoque de cruce agregando seguridad sobre el funcionamiento interno del cifrado.

6. Pruebas

Con la intención de validar la robustez y calidad estadística de la propuesta criptográfica, este capítulo desempeña su papel a través de la presentación de pruebas orientadas a evaluar aspectos de comportamiento del cifrado. Entre las cuales se encuentra: nivel de aleatoriedad, dispersión, sensibilidad de llave y distribución de datos.

Para finalizar, se incluyen pruebas comparativas que contribuyen a consolidar la validez funcional y estadística de la propuesta, reforzando su disposición como alternativa y extensión del área criptográfica, obteniendo también un panorama sobre el nivel de desempeño frente a un estándar ya consolidado (AES en modo CBC).

6.1. Entorno Experimental

Durante el desarrollo del algoritmo de cifrado propuesto, así como en la ejecución de las pruebas estadísticas y de validación comparativa frente al cifrado AES, se hizo uso de dos entornos de hardware. Ambos equipos fueron empleados de manera complementaria, tanto para la codificación del modelo como para la ejecución de pruebas relacionadas con el texto cifrado y su posterior evaluación.

Equipo de Escritorio de Alto Rendimiento:

Características	
Procesador	AMD Ryzen 5 7600 (6 núcleos, 3.80 GHz)
Memoria RAM	32 GB
Almacenamiento	2 TB SSD
Tarjeta Gráfica	NVIDIA GeForce RTX 4060
Sistema Operativo	Windows 11 Pro (Versión 24H2, 64 bits)

Tabla 15. Equipo de Escritorio.

Equipo Portátil (Gama Media-Alta):

Características	
Procesador	Intel Core i5 (11ª Gen, 2.70 GHz)
Memoria RAM	8 GB
Almacenamiento	1 TB + 512 GB SSD
Tarjeta Gráfica	NVIDIA GeForce RTX 2050
Sistema Operativo	Windows 11 Home (Versión 24H2, 64 bits)

Tabla 16. Equipo Portátil.

6.2. Elementos de Prueba

6.2.1. Llave de Secreta

Según la disposición normativa del FIPS 140 y la NIST STS 800, se recomienda una llave de 16 caracteres que incluya letras (mayúsculas y minúsculas), dígitos y símbolos especiales [2] [11]. Por lo anterior, se propone la siguiente Llave Secreta para el proceso de Pruebas de Cifrado:

3mm@Nue1/dE&3Su$

6.2.2. Texto de Prueba

Para la generación de este texto se tomó en consideración el conjunto de pruebas proporcionado por el National Institute of Standards and Technology (NIST), el Statistical Test Suite (STS), en el que se recomienda una cadena de al menos 10^6 bits para obtener resultados estadísticamente significativos y confiables [57].

Resumen: “La última pregunta" de Isaac Asimov.
"La última pregunta" es un relato de ciencia ficción escrito por Isaac Asimov, que explora los temas de la inteligencia artificial, el destino del universo y la eterna búsqueda de respuestas. La historia sigue a un grupo de científicos y humanos que, a lo largo de los siglos, le hacen a una supercomputadora, llamada Multivac, una pregunta fundamental: ¿Cómo se puede evitar la entropía y asegurar la supervivencia del universo? La computadora, en cada era,

responde que no es posible, pero conforme pasa el tiempo y la tecnología avanza, la humanidad sigue buscando una respuesta definitiva.

El relato abarca miles de años, con diferentes generaciones de humanos y máquinas. A lo largo de los siglos, la pregunta persiste, mientras que la capacidad de las computadoras para comprender el universo y realizar cálculos cada vez más complejos aumenta exponencialmente. Al final, la última pregunta es planteada una vez más, pero esta vez la respuesta llega de manera inesperada, revelando un concepto profundo sobre el tiempo, la creación y el destino de todo lo que existe.

La narrativa invita a reflexionar sobre el papel de la tecnología, la inteligencia artificial y nuestra comprensión del universo. Asimov plantea una pregunta filosófica sobre la naturaleza del tiempo y el futuro, mientras los personajes buscan una solución al eterno problema de la entropía. El cuento subraya la idea de que la tecnología, por avanzada que sea, no puede escapar de los límites inherentes al universo, pero también ofrece una mirada esperanzadora hacia lo desconocido.

(Asimov,1982)

6.3. Análisis de Resultados

Tanto el criptograma resultante del modelo de cifrado propuesto, al que en adelante nos referiremos como Cifrado TC (Trenzado – Coantípoda), como el generado mediante AES en modo CBC, fueron procesados fueron procesados de acuerdo con los requisitos establecidos (llave secreta y texto de prueba), así como por cada uno de los requerimientos de validación.

6.3.1. Análisis de Aleatoriedad

A continuación, se presentan los resultados obtenidos por el Cifrado TC y AES en modo CBC:

Métricas – NIST STS		Cifrado TC	Cifrado AES – CBC
1	Frequency (Monobit) Test	0.48476	0.58179
2	Frequency Test within a Block	0.52043	0.21685
3	Runs Test	0.06699	0.12281

4	Test for the Longest Run of Ones in a Block	0.26019	0.70516
5	Binary Matrix Rank Test	0.59697	0.00177 Non-Random
6	Discrete Fourier Transform (Spectral) Test	0.01246	0.33608
7	Non-overlapping Template Matching Test	0.74475	0.91733
8	Overlapping Template Matching Test	0.39205	0.83792

Tabla 17. Resultados – Pruebas Comparativas NIST STS [1 - 8].

Métricas – NIST STS		**Cifrado TC**	**Cifrado AES – CBC**
9	Maurer's "Universal Statistical" Test	-1.0 Non-Random	-1.0 Non-Random
10	Linear Complexity Test	0.98220	0.72507
11	Serial Test	0.40115	0.74837
12	Approximate Entropy Test	0.06002	0.10864
13	Cumulative Sums (Cusum) Test	0.14934	0.57007
14	Random Excursions Test	0.55459	0.56232
15	Random Excursions Variant Test	0.91097	0.92419

Tabla 18. Resultado – Pruebas Comparativas NIST STS [9 - 15].

Validación Preliminar

Según los criterios de aceptación definidos anteriormente (Sección 4.4.1), el cifrado propuesto TC supera satisfactoriamente el umbral de Secuencia Aceptable, obteniendo resultados favorables en 14 de las 15 pruebas posibles, lo que equivale a más del 90% del total de la suite NIST STS 800-22. Así mismo, entre el conjunto aprobatorio se encuentran las 4 pruebas fundamentales de aleatoriedad.

Este desempeño confirma un cumplimiento preliminar de los requisitos de aleatoriedad, validando que TC genera secuencias con propiedades estadísticas

aceptables, sobrepasando el indicador mínimo de calidad establecido para este trabajo.

Observaciones Específicas

- Prueba de Estadística Universal (Maurer, prueba 9): Ambos cifradores producen un resultado no aprobatorio. Esta observación no necesariamente invalida la seguridad, pero sí indica un área de estudio futuro.
- Prueba de Complejidad Lineal (Prueba 10): El cifrado propuesto TC muestra mejores resultados, lo que sugiere mayor resistencia a ataques basados en predicción de patrones.
- Prueba Espectral (Prueba 6): El $p - value$ del cifrado TC (0.01246) está muy próximo al umbral mínimo, lo que podría indicar un análisis complementario con otras secuencias o tamaños de entrada.
- En general, el cifrado TC demuestra un comportamiento altamente aleatorio y competitivo frente al estándar AES en modo CBC, lo que respalda su robustez frente a análisis estadísticos.

En resumen, el cifrado TC aprueba satisfactoriamente los requisitos de validación establecidos, descartando la presencia de sesgos significativos en las secuencias generadas. A pesar de fallar en la prueba Maurer (Prueba 9), los resultados globales respaldan una solidez estadística, reforzando la viabilidad del algoritmo como una nueva alternativa criptográfica.

6.3.2. Análisis de Histogramas

Los histogramas siguientes se interpretan como mensaje inicial o texto original en el apartado izquierdo y el criptograma resultante a su derecha, la Fig.24. presenta el cifrado propuesto TC, mientras que la Fig.26. corresponde al algoritmo AES en modo CBC.

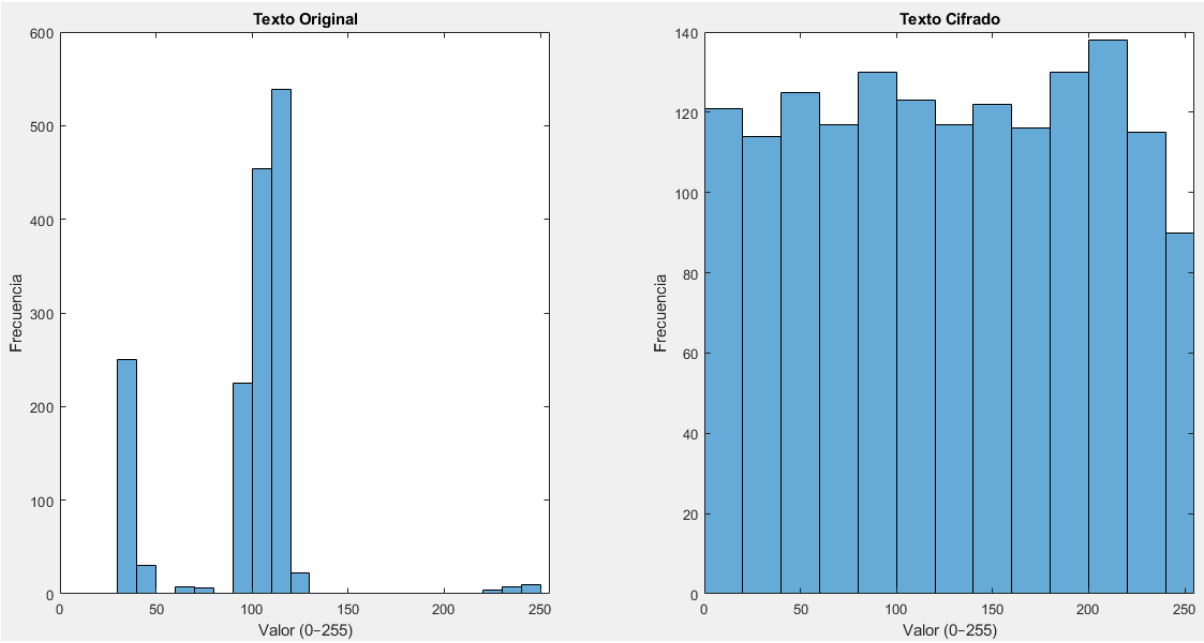


Fig. 24. Histograma – Cifrado TC Propuesto

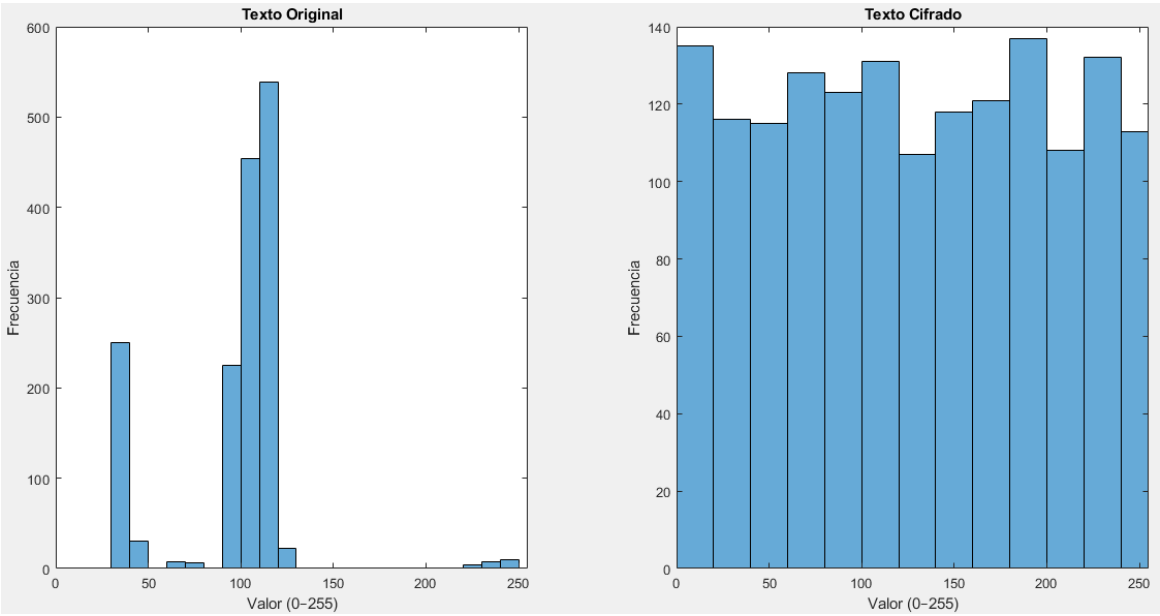


Fig. 25. Histograma – Cifrado AES modo CBC.

Observaciones Preliminares

- Texto Original: Este histograma en ambos casos, presenta una distribución sesgada y estructurada, siendo un comportamiento esperado, en donde los valores son marcadamente predominantes debido a su relación con el conjunto de caracteres imprimibles, en un rango acotado de 32 a 128 (ASCII) y en menor medida de 161 a 255 (Latin-1).
- Criptograma TC: Resulta en un histograma más uniforme y de distribución equiprobable deseada, tal que, esta ausencia de picos o concentraciones específicas no tiene relación con el Texto Original, lo cual es una característica indispensable en los esquemas criptográficos modernos.
- Comparativa TC vs AES-CBC: Para este caso, el estándar AES en modo CBC presenta un histograma ligeramente más plano en comparación con TC, lo que indica una difusión marginalmente superior en la ocultación de patrones estructurales.

De manera objetiva, el Cifrado TC alcanza un desempeño consistente bajo las premisas definidas en la Sección 4.4.2, validando su capacidad de dispersión y se considera como un resultado satisfactorio.

6.3.3. Análisis de Distribución y Correlación

	Cifrado TC	**Cifrado AES CBC**
Entropía de Shannon	9.9985	9.9982
Chi-Cuadrado	0.8825	0.9967
Índice de Coincidencia	0.0044	0.0038
Correlación Pearson	0.0238	0.0279

Tabla 19. Resultados – Análisis de Distribución y Correlación.

Validación Preliminar

Siguiendo los criterios de validación definidos en la Sección 4.4.2., se muestran los siguientes resultados:

- Entropía de Shannon: Se supera el umbral Aceptable (≥ 0.975), alcanzando valores cercanos al máximo teórico de 1.0, lo que confirma la ausencia de redundancia significativa y una alta aleatoriedad.
- Prueba Chi-Cuadrado: El resultado muestra alta uniformidad en la distribución de los caracteres cifrados, con predominancia en los obtenidos por el estándar AES en modo CBC y un desempeño del cifrado TC dentro del rango aprobatorio.
- Índice de Coincidencia (IC): Para un alfabeto de 256 símbolos, el valor ideal tiende a 0.0039, muy inferior al IC del idioma natural (~0.039). Los valores obtenidos por TC y se mantienen dentro de este rango, cumpliendo con el criterio de validación, acercándose al valor ideal.
- Coeficiente de Correlación Pearson: Los bajos valores (≤ 0.02) confirman que no existen relaciones lineales significativas entre los símbolos del criptograma, cumpliendo con el criterio establecido.

Por ende, el análisis conjunto de estas métricas confirma que el cifrado propuesto TC cumple satisfactoriamente con los criterios de dispersión y no correlación definidos en la sección 5.4.4., y se refleja que el algoritmo basado en Grupos Cuánticos mantiene una calidad competente, favoreciendo la idea de esta propuesta como una alternativa funcional desde este análisis preliminar estadístico.

6.3.4. Análisis de Sensibilidad de Llave

Para evaluar esta propiedad en el Cifrado TC, se utilizó la clave original:

$$3mm@Nue1/dE\&3Su\$$$

y se construyeron dos llaves ligeramente modificadas:

- $R1$: $3mm@Nue1/dE\&3Su\#$ (último carácter $ cambiado por #)
- $R2$: $3mm@Nue1/dE\&3Su\%$ (último carácter $ cambiado por %)

	Porcentaje de Similitud
Cifrado TC – R1	48.22 %
Cifrado TC – R2	48.45 %

Tabla 20. Resultado – Pruebas de Sensibilidad de Llave.

Para ambos casos, se obtuvieron valores próximos al 48% de bits modificados respecto al criptograma original. Estos resultados se encuentran dentro del rango esperado y próximos al umbral ideal del 50%, lo cual indica validación positiva sobre el alto nivel de difusión y sensibilidad ante variaciones mínimas de la llave simétrica.

6.4. Discusión

Los resultados obtenidos en las pruebas de validación demuestran que el cifrado propuesto TC cumple los criterios de aceptación establecidos, alcanzando un nivel estadístico adecuado para cada evaluación planteada. Obteniendo preliminares aprobatorios que confirman propiedades consistentes de un sistema criptográfico moderno.

Partiendo de las pruebas de aleatoriedad basadas en el conjunto proporcionado por la suite NIST STS 800-22, y la aprobación de más del 90% de las mismas, se confirma que el cifrado propuesto TC, produce secuencias deseables, con un criptograma libres de patrones estructurales, característica positiva de resistencia frente a ataques basados en análisis estadístico.

Complementando al análisis de aleatoriedad, las pruebas incluidas sobre correlación y uniformidad, aportan un nivel extra de aprobación estadística. Entre estas métricas se decide incorporar un respaldo visual y descriptivo, que se relaciona a su vez, en la interpretación de resultados obtenidos por los criterios de validación numérica.

El análisis de sensibilidad de llave por otro lado, es una métrica que se incluye para verificar el cumplimiento del Principio de Avalancha, centrado en la robustez del cifrado para resistir ataques diferenciales y de búsqueda cercana. Por lo que, el resultado aprobatorio cercano al 50% confirma esta capacidad del algoritmo para producir cambios significativos ante ligeras modificaciones en la llave.

Como primera conclusión, el Cifrado TC ha demostrado ser una opción funcional robusta, que cumple y se apega a los principios criptográficos estadísticos esenciales, y que en comparativa con el estándar AES presenta un desempeño competitivo, sosteniendo su viabilidad como alternativa potencial.

7. Conclusión y Trabajo a Futuro

7.1. Contribuciones de la Investigación

La presente investigación aporta diversas contribuciones tanto al ámbito teórico de la criptografía como a su aplicación práctica en entornos computacionales. Entre las principales se destacan:

- Formalización Computacional de Estructuras Algebraicas no Convencionales: Se expone una abstracción conceptual de las estructuras algebraicas de la teoría de Grupos Cuánticos, trasladando su formulación matemática original a un entorno de computación convencional, sin requerir de hardware especializado.
- Diseño de Operadores Criptográficos inspirados en Estructuras Cuánticas: Se implementaron los operadores de Trenzado y Coantípoda como procesos computacionales mediante el uso de la aritmética modular, representaciones matriciales y de campos finitos.
- Construcción de un Algoritmo Funcional y Verificable: El modelo propuesto se consolidó en un prototipo operativo, validado mediante métricas de criptoanálisis estadístico (aleatoriedad, distribución y sensibilidad de clave).
- Aporte Metodológico al Diseño Criptográfico: Se desarrolló un enfoque gradual que integra modelado gráfico, formalización matemática e implementación computacional, constituyendo un puente entre la abstracción algebraica y el diseño de procesos criptográficos verificables.
- Proyección hacia Investigaciones Futuras: Estas contribuciones abren la posibilidad de explorar mejoras en rendimiento, aplicaciones híbridas con algoritmos modernos, y su integración en sistemas de seguridad reales.

7.2. Trabajo a Futuro

A partir de los resultados obtenidos y del enfoque teórico desarrollado, se vislumbran varias líneas de trabajo que pueden ampliar y fortalecer el alcance del cifrado basado en Grupos Cuánticos:

- Extensión a otras Estructuras en el marco de Grupos Cuánticos: Explorar otras construcciones algebraicas como lo son las Categorías Monoidales y Quasi-Hopf Álgebras, con la finalidad de ampliar el repertorio de operadores disponibles.
- Optimización de los Operadores: Se considera que los operadores Producto y Coproducto, pueden funcionar de manera independiente en lugar de combinarse en Antípodas y Coantípodas, proporcionando un diseño granular, más controlado.
- Nuevos enfoques de Cifrado: Diseñar a partir de otros enfoques criptográficos (simétricos y asimétricos; de flujo o en bloque), con la intención de construir cifrados personalizados o adaptativos, que incluso varíen en el tiempo de ejecución según las reglas internas del modelo.
- Evaluación de Seguridad Avanzada: Realizar un estudio enfocado en la resistencia del cifrado frente a ataques de fuerza bruta, análisis diferencial o ataques estadísticos, para obtener una evaluación integral de la robustez y fiabilidad del algoritmo.
- Aplicación en otros Dominios de Datos: Adaptar el modelo a otros tipos de datos, como lo pueden ser imágenes en escala de grises, imágenes RGB e imágenes DICOM, así como, formatos de audio y de flujos de red.

7.3. Limitaciones

Si bien la investigación aporta un marco novedoso al introducir operadores inspirados en estructuras de Grupos Cuánticos dentro del cifrado simétrico, también se presentan ciertas limitaciones a considerar. En primer lugar, el desarrollo se ha centrado en un modelo teórico-funcional y su implementación prototípica, sin explorar su desempeño en escenarios de gran escala o en plataformas de uso intensivo.

Por otro lado, la validación se ha realizado principalmente mediante pruebas estadísticas de criptoanálisis, sin abarcar de forma exhaustiva otros tipos de ataque, ni escenarios más complejos. Por ello, aunque se evidencia un nivel inicial de robustez, aún se requiere un análisis más profundo que garantice una seguridad más completa.

7.4. Conclusiones Generales

El proyecto dio inicio con la mentalidad de desarrollar un prototipo funcional, con expectativas moderadas, considerando aceptable aprobar los requerimientos mínimos de validación establecidos, lo que permitía consolidar esta propuesta experimental. Sin embargo, en contraste a lo esperado, los resultados obtenidos son notablemente favorables, superando considerablemente el umbral mínimo previsto.

Estos resultados no solo validan la coherencia del diseño propuesto, también evidencian la solidez del modelo criptográfico desarrollado, mostrando un comportamiento competitivo frente al estándar de cifrado AES. Aunque aún existen aspectos por optimizar y un análisis más exhaustivo por realizar, la investigación evidencia que esta propuesta de innovación es viable y prometedora.

El desarrollo de este proyecto fue posible gracias a la aplicación de los conocimientos sobre la Ingeniería de Software, del cual, se emplearon herramientas metodológicas, tales como el análisis de requisitos, el modelado estructurado mediante diagramas y una implementación modular. Del mismo modo, se integraron métricas de evaluación y principios de aseguramiento de calidad (SQA), garantizando la verificación del comportamiento funcional del algoritmo y el alcance de los objetivos planteados.

De forma personal, este proyecto me permitió poner en práctica competencias clave de la formación en Ingeniería de Software, tales como el razonamiento algorítmico, la

formalización de modelos, el diseño de arquitecturas lógicas y la validación experimental. En el que, a través de un tratamiento adecuado, fue posible traducir estructuras algebraicas no convencionales en operadores funcionales concretos.

En conclusión, más allá de su función como prototipo, esta propuesta representa una contribución potencial en el ámbito de la criptografía computacional, con posibilidades reales de adaptación. De este modo, el trabajo no solo valida los aprendizajes adquiridos, sino que demuestra cómo estos pueden proyectarse hacia soluciones innovadoras y aplicables en tecnologías actuales.

8. Bibliografía

[1] S. Majid, Foundations of Quantum Group Theory, Cambridge University Press, 1995.

[2] NIST, « FIPS PUB 197: Advanced Encryption Standard (AES). National Institute of Standards and Technology,» 2001. [En línea]. Available: https://doi.org/10.6028/NIST.FIPS.197.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice (7ª ed.), Pearson, 2017.

[4] A. J. v. O. P. C. &. V. S. A. Menezes, Handbook of Applied Cryptography, CRC Press, 1996.

[5] C. H. &. B. G. Bennett, «Quantum Cryptography: Public Key Distribution and Coin Tossing,» de *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, 1984.

[6] D. J. Bernstein, «ChaCha, a variant of Salsa20,» Workshop Record of SASC 2008: The State of the Art of Stream Ciphers, 2008. [En línea]. Available: https://cr.yp.to/chacha/chacha-20080128.pdf.

[7] N. I. o. S. a. T. (NIST), Data Encryption Standard (DES) and Triple Data Encryption Algorithm (TDEA)., Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3). Gaithersburg, MD: U.S. Department of Commerce., 1999.

[8] R. L. S. A. &. A. L. Rivest, «Cryptographic communications system and method». U.S. Patent and Trademark Office Patente Patent No. 4,161,343, 1979.

[9] V. S. Miller, «"Use of elliptic curves in cryptography," in Advances in Cryptology — CRYPTO '85 Lecture Notes in Computer Science,» *Lecture Notes in Computer Science. Springer,* vol. 218, p. 417–426, 1986.

[10] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469–472, 1985.

[11] National Institute of Standards and Technology (NIST), «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,» NIST

Special Publication 800-22 Rev. 1a, 2010. [En línea]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf.

[12] M.-J. O. Saarinen, «SP 800-22 describes 15 statistical tests and suggests that they can be used for the evaluation of random and pseudorandom number generators in cryptographic applications,» Public Comments on SP 800-22 Rev. 1a, PQShield Ltd., Feb. 2022, 2022.

[13] C. E. Shannon, «A Mathematical Theory of Communication,» Bell System Technical Journal, vol. 27, pp. 379–423, 1948. [En línea]. Available: 10.1002/j.1538-7305.1948.tb01338.x.

[14] K. Pearson, On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling., Philosophical Magazine, 50, 157–175., 1900.

[15] W. F. Friedman, The Index of Coincidence and Its Applications in Cryptography, Riverbank Lab; reimpreso por Aegean Park Press, 1987, 1922.

[16] H. Feistel, Cryptography and Computer Privacy, Scientific American, vol. 228, no. 5, pp. 15–23, 1973.

[17] R. W. Hamming, Error detecting and error correcting codes, Bell Syst. Tech. J., vol. 29, no. 2, pp. 147–160, 1950.

[18] M. A. &. C. I. L. Nielsen, Quantum Computation and Quantum Information, Cambridge University Press, 2010.

[19] National Institute of Standards and Technology , «Post-Quantum Cryptography Standardization,» NIST, [En línea]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography. [Último acceso: 2025].

[20] A. K. Ekert, Quantum cryptography based on Bell's theorem, Physical Review Letters, 67(6), 661–663., 1991.

[21] D. G. a. I. Chuang, «Quantum digital signatures,» arXiv preprint, arXiv:quant-ph/0105032, 2001. [En línea]. Available: https://arxiv.org/abs/quant-ph/0105032.

[22] L. K. Chen, «Report on post-quantum cryptography,» NISTIR 8105. National Institute of Standards and Technology, 2016.

[23] D. J. B. a. T. Lange, «Post-quantum cryptography,» Nature, vol. 549, pp. 188–194, 2017. [En línea]. Available: https://doi.org/10.1038/nature23461.

[24] J. B. e. al., CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM, Proc. IEEE European Symp. Security and Privacy (EuroS&P), 2018,, 2018.

[25] D. L. a. L. Ducas, «CRYSTALS Cryptographic Suite for Algebraic Lattices,» 2017. [En línea]. Available: https://pq-crystals.org/dilithium/.

[26] D. J. B. e. al., SPHINCS: practical stateless hash-based signatures, Cryptology ePrint Archive, Report 2014/795, 2014.

[27] J. A. Gallian, Contemporary Abstract Algebra (9th ed.), Cengage Learning, 2016.

[28] J. Stillwell, Mathematics and Its History, Springer, 2002.

[29] B. C. Hall, Lie groups, Lie algebras, and representations: An elementary introduction (2nd ed.), Springer, 2015.

[30] J. Stillwell, Naive Lie Theory, Springer, 2008.

[31] C. Kassel, Quantum groups (Vol. 155), Springer-Verlag, 1995.

[32] V. &. P. A. Chari, A Guide to Quantum Groups, Cambridge University Press, 1994.

[33] M. Durdevic, «Geometry of quantum principal bundles I,» Communications in Mathematical Physics, 175(3), 457–521, 1996. [En línea]. Available: https://doi.org/10.1007/BF02099560.

[34] J. &. S. M. Baez, Physics, Topology, Logic and Computation: A Rosetta Stone, Springer, 2010.

[35] K. H. Rosen, Elementary Number Theory and Its Applications, Pearson, 2011.

[36] W. &. W. L. Trappe, Introduction to Cryptography with Coding Theory, Pearson, 2006.

[37] C. F. Gauss, Disquisitiones Arithmeticae, 1801.

[38] D. M. Burton, Elementary Number Theory (7th ed.), McGraw-Hill, 2010.

[39] D. S. &. F. R. M. Dummit, Abstract Algebra (3rd ed.), Wiley, 2004.

[40] J. B. Fraleigh, A First Course in Abstract Algebra (7th ed.), Pearson, 2002.

[41] H. Tapia-Recillas, «Sobre algunas aplicaciones de los campos de Galois.,» Miscelánea Matemática, 53, 81–100., 2011. [En línea]. Available: https://miscelaneamatematica.org/download/tbl_articulos.pdf2.b83a7656c55ef738.353330362e706466.pdf.

[42] J. F. &. R. K. W. Kurose, Computer Networking: A Top-Down Approach. 8th ed., Pearson, 2021.

[43] B. A. Forouzan, Data Communications and Networking. 5th ed., McGraw-Hill, 2023.

[44] A. F. f. N. Interchange, «“código estándar estadounidense para el intercambio de información”,» RFC 20, 1969.

[45] M. /. IANA, «Information Technology — 8-bit single-byte coded graphic character sets — Part 1: Latin Alphabet No. 1.,» ISO/IEC 8859-1:1998, 1998.

[46] U. Consortium, «Unicode Standard,» ISO/IEC 8859, 1991.

[47] S. Josefsson, «The Base16, Base32, and Base64 Data Encodings,» RFC 4648. Internet Engineering Task Force (IETF), 2006. [En línea]. Available: https://tools.ietf.org/html/rfc4648.

[48] I. N. B. N. Freed, «Multipurpose Internet Mail Extensions (MIME) Part One (sección 6.7),» RFC 2045, 1996.

[49] I. S. 1003.1, «uuencode - encode a binary file,» ISO/IEC 646:1991, 1991.

[50] MathWorks, «matlab.net.base64encode - Encode binary data using Base64,» [En línea]. Available: https://www.mathworks.com/help/matlab/ref/matlab.net.base64encode.html. [Último acceso: 2025].

[51] R. S. &. M. B. R. Pressman, «Software Engineering: A Practitioner’s Approach (9th ed.).,» McGraw-Hill, 2020.

[52] I. Sommerville, «Software Engineering (10th ed.).,» Pearson, 2016.

[53] ISO/IEC, «Information technology — Open Distributed Processing — Unified Modeling Language (UML) version 1.4.2,» de *ISO/IEC 19501*, ISO/IEC, 2005.

[54] ISO/IEC/IEEE, «Life Cycle Processes — Requirements Engineering,» de *ISO/IEC/IEEE 29148:2018*, ISO/IEC/IEEE, 2018.

[55] J. &. R. V. Daemen, The design of Rijndael: AES - The advanced encryption standard, Springer, 2002.

[56] R. A. &. J. C. R. Horn, Matrix Analysis (2nd ed.), Cambridge University Press, 2013.

[57] A. S. J. N. J. S. M. B. E. L. S. Rukhin, «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.,» NIST Special Publication 800-22 Revision 1a. National Institute of Standards and Technology. , 2010. [En línea]. Available: https://doi.org/10.6028/NIST.SP.800-22r1a.

[58] N. K. a. A. Menezes, «A Survey of Public-Key Cryptosystems,» *SIAM Review,* vol. 46, nº 4, p. 599–634, 2004.

[59] G. Vernam, «Cipher printing telegraph systems for secret wire and radio telegraphic communications,» *Journal of the IEEE,* vol. 55, nº 4, p. 109–115, 1926.

[60] D. J. Bernstein, «Salsa20 specification,» eSTREAM, ECRYPT Stream Cipher Project, 2005. [En línea]. Available: https://cr.yp.to/snuffle/salsafamily-20071225.pdf.

[61] OpenBSD Project, «OpenSSH: Secure Shell,» [En línea]. Available: https://www.openssh.com/. [Último acceso: 2025].

[62] T. D. a. C. Allen, «The TLS Protocol Version 1.0,» RFC 2246, 1999. [En línea]. Available: https://tools.ietf.org/html/rfc2246.

[63] B. R. a. S. Turner, «Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification,» RFC 5751, 2010. [En línea]. Available: https://tools.ietf.org/html/rfc5751.

[64] P. R. Zimmermann, The Official PGP User's Guide, MIT Press, 1995.

[65] N. Koblitz, «Elliptic curve cryptosystems,» *Mathematics of Computation,* vol. 48, nº 177, p. 203–209, 1987.

[66] C. E. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949.

[67] A. Agresti, Statistical Methods for the Social Sciences (5th ed.), Pearson, 2018.

[68] W. F. Friedman, Elements of Cryptanalysis, Riverbank Publications, 1920.

[69] Y. U. D. S. M. &. F. I. Gui, «Metrology Challenges in Quantum Key Distribution. Journal of Physics: Conference Series, 2416,» 012005 DOI, 2022. [En línea]. Available: https://doi.org/10.1088/1742-6596/2416/1/012005.

[70] S. Liao, Satellite-to-ground quantum key distribution, Nature, vol. 549, pp. 43–47, 2017.

[71] G. R. W. T. a. H. Z. N. Gisin, «Quantum cryptography,» Rev. Mod. Phys., vol. 74, pp. 145–195, 2002. [En línea]. Available: https://doi.org/10.1103/RevModPhys.74.145.

[72] P. J. Clarke, Experimental demonstration of quantum digital signatures, Nature Communications, 3, 1174, 2012.

[73] P. W. a. E. A. V. Dunjko, «Quantum digital signatures without quantum memory,» Phys. Rev. Lett., vol. 112, no. 4, p. 040502, 2014. [En línea]. Available: https://doi.org/10.1103/PhysRevLett.112.040502.

[74] W. Castryck, «Breaking SIDH in polynomial time,» Advances in Cryptology – EUROCRYPT 2023, vol. 14007, pp. 473–503, 2023. [En línea]. Available: https://eprint.iacr.org/2022/975.

[75] P. G. S. N. D. &. O. V. Etingof, Tensor Categories, American Mathematical Society, 2015.

[76] Unicode Consortium, «The Unicode Standard, Version 15.1.,» UNICODE, 2024. [En línea]. Available: https://www.unicode.org/versions/Unicode15.1.0/.

[77] International Organization for Standardization, ISO 646:1991 – Information technology – ISO 7-bit coded character set for information interchange, ISO, 1991.

[78] J. G. Myers, «Base64 Encoding,» RFC 4648, IETF, 2006. [En línea]. Available: https://datatracker.ietf.org/doc/html/rfc4648.

[79] J. B. a. N. S. M. Jones, «JSON Web Token (JWT),» RFC 7519, IETF, 2015. [En línea]. Available: https://datatracker.ietf.org/doc/html/rfc7519.

[80] P. Gutmann, «X.509 Style Guide,» University of Auckland, 2004. [En línea]. Available: https://www.cs.auckland.ac.nz/~pgut001/pubs/x509guide.txt.

[81] N. &. B. N. Freed, «Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,» RFC 2045. Internet Engineering Task Force, 1996. [En línea]. Available: https://datatracker.ietf.org/doc/html/rfc2045.

[82] T. Berners-Lee, «Hypertext Transfer Protocol -- HTTP/1.0,» RFC 1945, IETF, 1996. [En línea]. Available: https://datatracker.ietf.org/doc/html/rfc1945.

[83] I. Asimov, The Last Question en The Complete Robot, Doubleday, 1982.

Anexos

A. Glosario (A)

A.1. General

- Álgebra de Hopf: Es una herramienta matemática que combina operaciones de juntar, separar e invertir elementos, usada para describir procesos que pueden deshacerse, en comportamiento cuántico.
- Categorías Trenzadas: Es una forma de representar relaciones o transformaciones entre elementos, usando “trenzas” o cruces para mostrar cómo se conectan o cambian, en términos cuánticos.
- Computación Convencional: Es la forma tradicional de usar computadoras.
- Cómputo Cuántico: Es una forma nueva de computación que usa reglas de la física cuántica para realizar cálculos usando unidades llamadas qubits.
- Grupos Cuánticos: Son estructuras matemáticas que ayudan a entender ciertos comportamientos complejos, especialmente en sistemas que no siguen reglas normales de simetría.
- Qubit: Es la unidad básica de información en la computación cuántica. A diferencia de un bit normal, que solo puede valer 0 o 1, un qubit puede representar ambos al mismo tiempo (Principio de Superposición).

A.2. Criptografía y Cifrado

- AES (Advanced Encryption Standard): Es un estándar de cifrado ampliamente usado para proteger datos. Usa llaves secretas para transformar información legible en datos incomprensibles.
- Aleatoriedad: Se refiere a la impredecibilidad de los datos cifrados. Cuanto más aleatoria parece la salida, más difícil es predecir o romper el cifrado.
- Cifrado por Permutación: Es un tipo de cifrado que reorganiza el orden de los datos sin cambiarlos, para ocultar su contenido.
- Cifrado por Sustitución: Consiste en reemplazar cada parte del mensaje original por otro valor siguiendo una regla específica, sin conservar el orden original.

- Cifrado Simétrico: Es un método donde la misma llave se usa tanto para cifrar como para descifrar un mensaje.
- Cifrado TC: Es un cifrado propuesto que combina dos operadores inspirados en estructuras cuánticas: el Trenzado y la Coantípoda.
- Confusión: Es una propiedad del cifrado que busca dificultar la relación entre la llave secreta y los datos cifrados.
- Difusión: Es la propiedad que asegura que un pequeño cambio en el mensaje o llave afecte muchas partes del resultado cifrado.
- Funciones Hash: Son funciones que convierten datos de cualquier tamaño en un valor fijo.
- Reversibilidad: Es la capacidad de un sistema de volver a su estado original, es decir, poder descifrar el mensaje y recuperar los datos originales.
- Sub-llave: Es una parte derivada de la llave principal, generada dentro del algoritmo para realizar operaciones específicas de cifrado.

A.3. Álgebra y Estructuras Matemáticas

- Álgebra: Área de las matemáticas que estudia estructuras y operaciones como la suma y la multiplicación en distintos conjuntos.
- Anillo Conmutativo: Estructura matemática con operaciones de suma y multiplicación, donde el orden de los factores no altera el producto.
- Antípoda: Operador que actúa como un “inverso estructural” dentro de ciertas estructuras algebraicas, útil para revertir procesos.
- Aritmética Modular: Es una forma de hacer operaciones como suma o multiplicación, pero los resultados se ajustan a un valor fijo (llamado “módulo”).
- Campo Finito (Campo de Galois): Conjunto limitado de números donde se pueden hacer sumas, restas, multiplicaciones y divisiones siguiendo reglas específicas.
- Coálgebra: Estructura matemática que permite descomponer elementos, en contraste con un álgebra que los combina.
- Coantípoda: Versión dual de la antípoda; permite invertir un proceso desde una estructura separada, como la de la coálgebra.

- Coconmutatividad: Propiedad en una coálgebra donde el orden de descomposición no altera el resultado.
- Coproducto: Operación que separa un elemento en varias partes, de forma que mantengan una relación estructurada con el original.
- Espacio Modular: Es un sistema en el que los números “vuelven a empezar” después de alcanzar cierto límite. Este tipo de espacio permite trabajar con ciclos o repeticiones de forma ordenada.
- Grupo Cuántico Trenzado: Estructura matemática avanzada que combina simetrías y entrelazamientos, útil para describir comportamientos en física cuántica y criptografía.
- Morfismo: Es una forma de transformar o conectar elementos entre dos sistemas manteniendo sus reglas. Por ejemplo, si algo se suma o multiplica en un sistema, al pasar al otro a través del morfismo, esa operación sigue teniendo sentido.
- Operación Modular (o Módulo): Es una operación matemática en la que solo se considera el “residuo” o el “resto” al dividir.
- Operadores Cuánticos: Reglas o funciones que transforman estados o elementos, inspiradas en el comportamiento de sistemas cuánticos.
- Producto Tensorial: Es una forma de combinar dos elementos para formar uno nuevo, que contiene información de ambos.
- Producto: Operación que combina dos elementos en uno solo, respetando reglas dentro de una estructura algebraica.
- Representación Diagramática: Forma visual de expresar operaciones o estructuras matemáticas mediante gráficos, como líneas o nodos conectados.
- Trenzado: Operación que entrelaza dos elementos, intercambiando su posición y afectando su estado de forma no trivial.

A.4. Codificación y Datos

- ASCII: Es un sistema de codificación que asigna un número a cada letra, número o símbolo básico en inglés. Se usa para representar texto en computadoras.
- Base64: Es una codificación que convierte datos en texto usando solo letras, números y algunos símbolos.

- Bits: Es la unidad más pequeña de información en una computadora. Puede tener solo dos valores: 0 o 1.
- Byte: Es una unidad básica de información que contiene 8 bits. Normalmente, un byte representa un solo carácter, como una letra o número.
- Caracteres Imprimibles: Son los símbolos visibles como letras, números, signos de puntuación o espacios. Excluyen comandos invisibles como “enter” o “tab”.
- Codificación de Caracteres: Es el proceso de asignar un número a cada símbolo o letra para que una computadora pueda almacenar y mostrar texto correctamente.
- Codificación de Transferencia: Es una forma de transformar datos en texto para poder enviarlos o guardarlos sin errores, por ejemplo, al adjuntar archivos o enviar información cifrada.
- Latin-1: Es una extensión de ASCII que agrega más letras y símbolos usados en idiomas europeos como el español, francés o alemán.
- Secuencias Binarias: Son cadenas formadas solo por ceros y unos. Representan cualquier tipo de dato digital, desde texto hasta imágenes, en el lenguaje que entiende una computadora.
- Unicode: Es un sistema de codificación que permite representar texto de casi todos los idiomas del mundo, incluyendo símbolos, emojis y caracteres especiales.

A.5. Evaluación y Herramientas

- Análisis de Histogramas: Es un método visual para observar cómo se distribuyen los valores en una imagen o texto cifrado, ayudando a identificar si los datos están bien mezclados (uniformes) o tienen patrones visibles.
- Coeficientes de Correlación: Son medidas que indican qué tan relacionados están dos conjuntos de datos. En criptografía, ayudan a saber si los datos cifrados todavía guardan relación con los originales.
- Entropía: Es una medida del desorden o impredecibilidad de los datos. En cifrado, una entropía alta indica que el resultado es más aleatorio y por lo tanto más seguro.
- Índice de Coincidencia: Es una medida que compara la frecuencia de letras o símbolos en un texto. Ayuda a detectar si un mensaje ha sido cifrado correctamente o si sigue siendo predecible.

- Métricas Estadísticas: Son herramientas matemáticas que permiten evaluar y comparar comportamientos de datos, como su uniformidad, aleatoriedad o distribución, útiles para probar la calidad del cifrado.
- NIST: Es una organización de EE.UU. que crea estándares técnicos. En criptografía, proporciona pruebas y recomendaciones para validar si un sistema es seguro.
- Pruebas estadísticas: Son evaluaciones que ayudan a determinar si una secuencia de datos se comporta de manera aleatoria o sigue algún patrón.
- Sensibilidad de Llave: Es la capacidad de un sistema de cifrado para cambiar completamente el resultado si se modifica la llave, incluso en un solo valor.
- SP 800-22: Es un documento oficial del NIST que define un conjunto de 15 pruebas estadísticas usadas para verificar si una secuencia de bits es lo suficientemente aleatoria para aplicaciones seguras.

B. Codificación

En esta sección se incluyen fragmentos clave de la implementación del prototipo, seleccionados por su relevancia en la construcción del modelo criptográfico. Cada bloque de código corresponde a un operador o proceso fundamental dentro del algoritmo de cifrado y descifrado.

B.1. Hexpansión

Este procedimiento permite expandir los datos iniciales hacia una representación de valores hexadecimales individuales. Constituye la base sobre la cual se construyen las transformaciones posteriores.

```
%% Expansión de Digitos Hexadecimales (2)
function decimales = hex_pansion(dato)
    tam = size(dato,2); % Decimal
    bloque32hex = cell(1,tam*2); % Hexadecimal / Longitud Doble
        for i=1:tam
            x=dec2hex(dato(1,i),2); % Convertir Decimal a Hexadecimal
            bloque32hex{1,i*2-1}=x(1); % Hexadecimal Derecho
            bloque32hex{1,i*2}=x(2); % Hexadecimal Izquierdo
        end
    % Pasar Hexadecimal a Decimal
```

```
    bloque32dec = hex2dec(bloque32hex); %
    % Bloque HexDec Fila a Columna
    decimales = zeros(1,tam*2);
        for i=1:tam*2
            decimales(1,i) = bloque32dec(i,1);
        end
end
```

B.2. Generación de Sub-llave

La función encargada de derivar la sub-llave a partir de la llave secreta. Introduce seguridad y simetría del proceso de cifrado y descifrado.

```
%% Función - Generación de Sub-llave de Cifrado
function subkey = subkey_generator(key)
    % Llave de Cifrado -> Código/Valor ASCII -> Hexadecimal
    keyhex = key_hex(key); %disp(keyhex);
    % Generación de Matrices MC1 y MC2
    MC1 = mc_1(keyhex); %disp(MC1);
    MC2 = mc_2(keyhex); %disp(MC2);
    % Subllave -> Por Generación y Ordenamiento de (MC3, MC4)
    subkey = mc_34(MC1, MC2); % subkey - Valor Decimal [0-15]
end

%% Valor Hexadecimal de Llave de Cifrado por Código/Valor ASCII
function keyhex = key_hex(key)
    % Llave de Cifrado - Valor/Código ASCII
    keydec = zeros(1,16);
    for i=1:16
    keydec(1,i)=key(i);
    end
    % Código/Valor ASCII a Hexadecimal
    keyhex = cell(1,32);
    for i=1:16
    y=dec2hex(keydec(1,i),2);
    keyhex{1,i}=y(1); % Vector Hexadecimal
    keyhex{1,16+i}=y(2);
    end
end
%% Ordenamiento de MC1
function MC1 = mc_1(keyhex)
MC1 = zeros(4);
salto=1;
    for i=1:4
       for j=1:4
           MC1(i,j)=hex2dec(char(keyhex{1,salto}));
           salto = salto+1;
       end
    end
end

%% Ordenamiento de MC2
function MC2 = mc_2(keyhex)
```

```
MC2 = zeros(4);
salto=17;
    for i=1:4
       for j=1:i
           MC2(i-j+1,j)=hex2dec(char(keyhex{1,salto}));
           salto = salto+1;
       end
    end
salto=32;
    for i=1:3
        for j=1:i
           MC2(4-i+j,4-j+1)=hex2dec(char(keyhex{1,salto}));
           salto = salto-1;
        end
    end
end

%% Generación y Ordenamiento de Matrices MC3 y MC4
function M34 = mc_34(MC1, MC2)
M34 = zeros(1,32);
MC3 = mod(MC1+MC2,16); %disp(MC3);
MC4 = mod(MC1*MC2,16); %disp(MC4);
x = 1;
    for i=1:4
        for j=1:4
           M34(1,x*2)=MC3(i,j); % Segunda Parte - Subllave
           M34(1,x*2-1)=MC4(i,j); % Primera Parte - Subllave
           x=x+1;
        end
    end
end
```

B.3. Renovación de Sub-llave

Proceso para la actualización dinámica de las sub-llave durante la transformación iterativa del mensaje.

```
%% Generación de Subllave - Dinamica
function subkeyD = dinamic_subkey(subkey16)
    subkeyD = zeros(1,16);
    % Suma de Valores Subkey +
    sumkey = sum(subkey16);
    % Multiplicación sumkey * subkey módulo 16
    for i=1:16
        subkeyD(1,i) = mod(subkey16(1,i)+sumkey,256);
    end
end
```

B.4. Operador de Trenzado

El operador Trenzado constituye uno de los dos posibles mecanismos de transformación aplicados en el proceso de cifrado. Su diseño se inspira en la permutación no trivial propia de las Categorías Trenzadas, lo que le confiere un comportamiento no conmutativo. Se aplica una vez por cada entrada de carácter.

```
% Opción: 1.- Cifrado; 2.- Decifrado
%% Operador de Trenzado
function cruce = crossover_operator(caracter,subkey,option)
    % Expansión de Hexadecimales Carácter y Llave
    vcaracter = hex_pansion(caracter); % Retorna una lista con los dos digitos
del elemento hexadecimal
    isubkey = hex_pansion(subkey); % Retorna una lista con los dos digitos del
elemento hexadecimal
    vcruce = zeros(size(vcaracter,1));
    % Cruce Aditivo - Indirecto
        vcruce(1,1) = additive_crossover(vcaracter(1,1),isubkey(1,1),option);
        vcruce(1,2) = additive_crossover(vcaracter(1,2),isubkey(1,2),option);
    % Salida Final
    cruce = hex_compuesto(vcruce);
end
%% Additive Crossover - Encrypt & Decrypt
function adicion = additive_crossover(ivector,isubkey,option)
    % Opción - Cifrado
    if option == 1
        adicion = mod(ivector+isubkey,16);
    % Opción - Decifrado
    elseif option == 2
        adicion = mod(ivector-isubkey,16);
    end
end
```

B.5. Operador Coantípoda

El operador Coantípoda representa la segunda opción dentro de los mecanismos de transformación. Su construcción está inspirada en la estructura de las Álgebras de Hopf y consiste en un proceso bifásico que actúa sobre cada elemento del mensaje. Se aplica dos veces por cada entrada de carácter.

```
%% Operador Coantípoda
function vcoan = coantipoda_operator(caracter,subkey,option)
    % Expansión de Hexadecimales Carácter
    vcaracter = hex_pansion(caracter); % Retorna una lista con los dos digitos
    vcoantipoda = zeros(1,2);
    % Operación Coantípoda
    if option == 1
        vcoantipoda(1,1) = coantipoda(vcaracter(1,1),subkey,1);
        vcoantipoda(1,2) = coantipoda(vcaracter(1,2),subkey,2);
```

```
    else
        vcoantipoda(1,1) = coantipoda(vcaracter(1,1),subkey,2);
        vcoantipoda(1,2) = coantipoda(vcaracter(1,2),subkey,1);
    end
    % Salida Final
    vcoan = hex_compuesto(vcoantipoda);
end

%% Operación Coantípoda
function ab = coantipoda(vcaracter,subkey,vco)
    % Expansión de Hexadecimales Llave
    isubkey = hex_pansion(subkey); % Retorna una lista con los dos digitos del
elemento hexadecimal
    % Selección de Matriz de Cifrado Fase 1
    if vco == 1
        fase1 = mz4(isubkey(1,1));
    else
        fase1 = mz4(isubkey(1,2));
    end
    % Indices de Localidad
    [fila, columna] = find(fase1 == vcaracter); % Localización del Valor
    a = fila;
    b = columna;
    % Selección de Matriz de Cifrado Fase 2
    if vco == 1
        fase2 = mz4(isubkey(1,2));
    else
        fase2 = mz4(isubkey(1,1));
    end
    % Valor Final
    ab = fase2(a,b);
end

%% Matrices de Cifrado
function z4 = mz4(isubkey)
    if mod(isubkey,8) == 0
        vz4 = [0,4,8,12,1,5,9,13,2,6,10,14,3,7,11,15];
    elseif mod(isubkey,8) == 1
        vz4 = [3,15,11,7,2,14,10,6,1,13,9,5,0,12,8,4];
    elseif mod(isubkey,8) == 2
        vz4 = [4,8,12,0,5,9,13,1,6,10,14,2,7,11,15,3];
    elseif mod(isubkey,8) == 3
        vz4 = [7,3,15,11,6,2,14,10,5,1,13,9,4,0,12,8];
    elseif mod(isubkey,8) == 4
        vz4 = [8,12,0,4,9,13,1,5,10,14,2,6,11,15,3,7];
    elseif mod(isubkey,8) == 5
        vz4 = [11,7,3,15,10,6,2,14,9,5,1,13,8,4,0,12];
    elseif mod(isubkey,8) == 6
        vz4 = [12,0,4,8,13,1,5,9,14,2,6,10,15,3,7,11];
    elseif mod(isubkey,8) == 7
        vz4 = [15,11,7,3,14,10,6,2,13,9,5,1,12,8,4,0];
    end
    isubkey = mod(isubkey,9);
    vz4 = mod(vz4+isubkey,16);
    z4 = reshape(vz4,4,4);
end
```

C. Resultado: Criptograma Codificado (Base64)

AOL4iqYk+g242nrlzVvFKgIG2fxsAUjAz1IgVvPG45gU31feSpldq12tIcuKyjUDVUm5ht9l3UE
aqwORDLKrxCQtgiW5x0LnRqVTobK+Kr11Q/Sb0lxOWyix1axZghO0AW3082a2dYr5FgRS
5vRdeycTB2IDDwXq5YDhZy603EcltQ7VWWzeZAudBmmbjcbgsaZQB30tDb2EH6u8z+DG
vXlH3qvUzI1GnYFz5JvxeK3K/NMA0prRFtVfkV4FZD2q02vfH/kPzsn0Uo/zcyOpzh3G7Q7g
uWwb0H3UV9KbmphpLaFf4meup2Nvj2gx2Jju49W4G3MkSveR8l0noIp+Rnv0FjKLWFqw8
QopN+DZN8twYVtmm2zPMCLF0LlsNRa08QSP8w0b1BpU3JfFy0mwcMa/qPnCm8a6eR/L
Br17fjdOz2NFKTe+8VeRx9VLdN+Y5by4B/EbhMqKIcUVyNA0HIWbvxapL++3N0V0nqJVz
HhegYjdRzjNWlDvINOWnLC9jyn0Rt2Qj8DwWHdJBQA2ES3aE3eQVp4+64KaailacOyR8L
XVRzz1dgvS42/VLMBaOmtNvVIMZXc0yeO7MM6rWGiQ+M5hLC/jvA6+0pqtIjKGelF+Yqya
RDzX1c2VxhZHoC9ZTkMW2OlWBeuZcocOzFVWJ4GbrY9lDWZFkhB6aftvcFBdBcjCr2xH
nyUTNap+61+52gcB9O6kNf/k2iVXbIQ8jeZjejVGNTcBst+p3K5HMwlpO8667v2AsDI+Zxmi4
KvskplkZmWTDzRBGttQBw7u+XLk9O8496R+QDtE45Kty9+jj55V0WhzKTWNUTdzyrzlQ6I
grAPKU41DBik4MbdUoX05Uv82aUvQz3Mu6hJcgoXf2oxQqxL1xtkXOVeVTK03jGKPLBs6
EUyer9NTxu7dfiz2nc75qTFdKfd3mcZSRpCv1dOvIpz0Lq3PHJXrC/rB6kapD/msXAU5BXko
zQHVArx+zYk6y81VzuP8MDfLdJb93c6LwXkmDBrBcxh42oybOm2KN9VYoQclOKscaFlU6
I/5ijleA7rvplEpxu3gcnSe1u++cL5se+sRF5HVoaM222Hxa15u9z6imZbiwq8UCX4EGdxc83
mnhPhHuCWxXon+S35AoWQDY/Gm+JL+KfCIxtcOgFcef5Vds/CkFj72XsgPbxY6RNDPtY
1MWJ/gC4zgtLxv5ouwQ/EWnmsEDbr2RzA1TYnVGHlcC83YzPyMR3DSJNHUIIKxinJB1+
VvTTGkrRG3UhWhBVpYvIo3Wm5WNkEMAyoav+6JsHv2niQWTpPBVj1lHLwFaHduICtvx
PLvPk73bOWvyzlXZuRt9jcJbuP8PzfoRtB/NSva1UqKTvic67hFBwlzP33Wl5nJvlj+Za4UyXg
TXQiz1a1D3W1ogkkWJWiam+0tHpcARdDAwkkTIhYQ2mNHWxIFIHuRNDyDeURDinJlHO
hlyiNvLg8GFpGh8y7bMN5s2ryNcEUgAr1rMjyNAnQJAyqvFMnlWcPg8V1Jqq08viPkeJQDX
/IP2tQh4KHD7rsnw0/C8YUqXsehZM+KM4xju3PU9HHmGYdRgE5AvGB7NTtSwjly3bdcwL
43uMOPIHsawK2qvkcWika+vTolOJpFDLn+Y87ez6gNnZlm3YEp8SvR0VqdV3th1ZyrfEY2J
Thn69kQkauQOuAtIyuyW7ZqpqXV8UxIvuDSSxTWfPy86INrhwFGMCEbKo8pJ0Ppgo1+I7
8gtjVk3Pi4hpluB8ZrBwIJb6UVfha5P2gBgj3kjME8j43hCHurK+m0tvBXYIcCmiCuWp+O4cfA
mYGIBuPx1YiL9pCCir1y1/uhe3sWNLRVFYnxUkHypXUBP1YVbs+TW38HektnvElAUF6Ps
NUjibC8QGHg24AzRgEk+6c9oOzkFPP8ZjTxR7LzzydSoZGviiCp3zYnETACOKk/BI+dOj1d
Ux/jz7mSJT+qwXvGwypjYNcJWzXYPiXwwUO+rGGFRDzaBoptmQUDHd3Gil2tXalB0gUa
BapBjGHeJM1kdFcUwza34LOtVLXJRuhvIF5nc/ltMaOVcd7JnQ==