Metodología para la adaptación de un algoritmo hash basado en SHA-256 mediante la validación de seguridad

Cevallos, Alejandro[1]; **Montalvo, Christian**[2]**;**

[1]*Instituto Superior Tecnológico 17 de Julio, Desarrollo de Software, ORCID: https://orcid.org/0009-0005-4215-618X, Ibarra, Ecuador*
[2]*Instituto Superior Tecnológico 17 de Julio, Desarrollo de Software, ORCID: https://orcid.org/0009-0005-3405-0043, Ibarra, Ecuador*

Recibido: 2025/11/10 Aceptado: 2026/01/05

Resumen: El artículo propone e implementa una metodología sistemática para la adaptación de un algoritmo hash basado en SHA-256 mediante la incorporación de un búfer bidimensional y etapas iterativas de amplificación, estructurada bajo un enfoque pedagógico y didáctico. A partir de una implementación experimental en Python se desarrolla la variante SECUREX siguiendo dos fases principales: preprocesamiento del mensaje con operaciones XOR y rotaciones dinámicas, y amplificación mediante transformaciones no lineales sobre una matriz 16x8. Se compara el algoritmo adaptado con SHA-256 original mediante pruebas unitarias funcionales, ensayos de resistencia a colisiones y análisis de entropía de Shannon en representaciones hexadecimal y Base64, empleando tamaños de muestra alineados con estándares NIST. Los resultados demuestran que SECUREX mantiene propiedades funcionales equivalentes a SHA-256 sin introducir colisiones, incrementando significativamente la entropía y diversidad de caracteres de las salidas, con el costo de mayor latencia computacional. Se concluye que la adaptación controlada y metodológicamente validada de algoritmos criptográficos existentes preserva la integridad de seguridad del algoritmo base, permitiendo explorar mejoras en robustez para contextos emergentes como la era poscuántica.

Palabras clave: hash; criptografía; validación; entropía; SHA-256; adaptación.

Methodology for adapting the SHA-256-based algorithm through security validation

Abstract: The article proposes and implements a systematic methodology for adapting a SHA-256-based hash algorithm by incorporating a bidimensional buffer and iterative amplification stages, structured under a pedagogical and didactic approach. Based on an experimental implementation in Python, the SECUREX variant is developed following two main phases: message preprocessing with XOR operations and dynamic rotations, and amplification through non-linear transformations over a 16×8 matrix. The adapted algorithm is compared with the original SHA-256 through functional unit tests, collision-resistance experiments, and Shannon entropy analysis in hexadecimal and Base64 representations, using sample sizes aligned with NIST standards. The results show that SECUREX maintains functional properties equivalent to SHA-256 without introducing collisions, significantly increasing the entropy and character diversity of the outputs, at the cost of higher computational latency. It is concluded that the controlled, methodologically validated adaptation of existing cryptographic algorithms preserves the security integrity of the base algorithm, enabling the exploration of robustness improvements for emerging contexts such as the post-quantum era.

Keywords: hash; cryptography; validation; entropy; SHA-256; adaptation.

1. Introducción

Una función hash criptográfica se define como un proceso que transforma datos de entrada de longitud arbitraria en una salida de longitud fija, denominada valor hash o resumen (AEPD & EDPS, 2019). La transformación que realizan estas funciones debe ser determinista, es decir, para una misma entrada siempre produce la misma salida, aunque una modificación mínima en los datos originales genere un valor hash completamente distinto (AEPD & EDPS, 2019). Un algoritmo hash debe tener ciertas características criptográficas principales para que este sea seguro. La primera característica es la resistencia a la preimagen, que establece la imposibilidad computacional de reconstruir un mensaje original a partir de su valor hash (Toledo Navarro, 2024). La segunda característica es la resistencia a colisiones, exigiendo que sea computacionalmente inviable encontrar dos mensajes cualesquiera que generen valores hash idénticos (AEPD & EDPS, 2019). Adicionalmente, una función hash debe presentar ciertas características secundarias incluyendo el efecto avalancha, mediante el cual cambios mínimos en la entrada producen variaciones significativas en la salida, asegurar una salida de datos fija independientemente del tamaño de entrada, facilidad de cálculo, no correlación entre entrada y salida, distribución uniforme de valores hash en el espacio de salida, pruebas unitarias enfocadas en la seguridad del algoritmo y la entropía de shannon (Toledo Navarro, 2024).

La aplicabilidad de funciones hash en sistemas de seguridad se fundamenta directamente en sus propiedades criptográficas. En firma digital, el valor hash permite verificar que un documento no ha sido alterado durante transmisión o almacenamiento; la resistencia a colisiones previene que un atacante sustituya el documento original por uno falsificado con el mismo hash, mientras que la resistencia a la preimagen impide la reconstrucción del documento desde su resumen (AEPD & EDPS, 2019; Toledo Navarro, 2024).

En sistemas de autenticación, las funciones hash transforman contraseñas en valores almacenables sin comprometer información original. Las pruebas de validación incluyen ataques de colisión mediante fuerza bruta, análisis de pruebas unitarias, evaluaciones estadísticas con el conjunto NIST SP 800-22 y análisis de entropía (National Institute of Standards and Technology, 2010).

El desarrollo de metodologías para el desarrollo de funciones hash personalizadas basadas en algoritmos existentes responde a requisitos específicos no satisfechos completamente por funciones estandarizadas. La construcción Merkle-Damgård proporciona un marco teórico para extender funciones de compresión a funciones hash de longitud variable, procesando iterativamente bloques de mensaje mediante esquemas como Davies-Meyer, Matyas-Meyer-Oseas y Miyaguchi-Preneel (Toledo Navarro, 2024). Adicionalmente, la metodología basada en el cifrado afín, aunque presenta limitaciones de seguridad cuando se emplea aisladamente, puede servir como componente base en diseños pedagógicos o en implementaciones híbridas que combinan múltiples transformaciones para alcanzar propiedades criptográficas adecuadas (Toledo Navarro, 2024).

La limitación identificada consiste en insuficientes metodologías explícitas y sistematizadas que articulen coherentemente el proceso de adaptación de algoritmos criptográficos, como el algoritmo SHA-256, con la validación rigurosa de propiedades de seguridad. Las modificaciones algorítmicas, sin protocolos estandarizados de evaluación, presentan dificultades para caracterizar de manera reproducible el comportamiento seguro de las adaptaciones realizadas. Esta problemática adquiere relevancia en propuestas orientadas a contextos operacionales emergentes, donde requisitos de seguridad específicos demandan ajustes funcionales que deben preservar las garantías criptográficas del algoritmo base.

Alejandro Ceballos

Autor por correspondencia

En la presente investigación se abordará este déficit mediante la implementación de un marco metodológico sistemático que integra la adaptación del algoritmo criptográfico existente con procedimientos estructurados de validación de seguridad. La metodología propuesta contempla la modificación controlada de una función hash conocida SHA-256, seguida de pruebas metodológicas comparativas con otro algoritmo hash de referencia. Este enfoque permite evaluar propiedades de resistencia a ataques, integridad funcional y robustez del comportamiento adaptado, facilitando la identificación temprana de fortalezas o debilidades que orienten el desarrollo de implementaciones más robustas y confiables para aplicaciones futuras.

2. Materiales y Métodos

Diseño experimental, entorno de implementación y especificaciones técnicas

El objetivo de esta investigación fue adaptar el algoritmo SHA-256 incorporando un búfer bidimensional y evaluar el impacto de esta mejora en atributos de seguridad esenciales mediante análisis comparativo cuantitativo, y se empleó una metodología experimental para contrastar el algoritmo SHA-256 original con nuestra variante modificada propuesta, denominada SECUREX.

El desarrollo y la experimentación se llevaron a cabo en Google Colab, utilizando las siguientes especificaciones, porque la plataforma en la nube proporcionó los recursos necesarios para el proyecto:

- Lenguaje de programación: Python 3.10
- Librerías clave: numpy (versión 1.26.4), hashlib, matplotlib, seaborn, pandas
- Hardware: instancia predeterminada en la nube de Google Colab (CPU Intel Xeon, 12.7 GB de RAM)
- Sistema operativo: Ubuntu 22.04.3 LTS.

Desarrollo del algoritmo base y del algoritmo modificado

SHA-256 como referencia

SHA-256 es una función hash criptográfica perteneciente a la familia SHA-2, que transforma entradas de longitud variable en salidas de longitud fija de 256 bits (NIST, 2015), y la implementación de referencia de SHA-256 utilizada en este estudio se obtuvo de la biblioteca hashlib en Python 3, cumpliendo con la especificación FIPS 180-4, por lo tanto, esta implementación sirvió como punto de referencia frente al cual se compararon el rendimiento y las características de seguridad del algoritmo modificado.

Arquitectura y ejecución de SECUREX

SECUREX fue diseñado como una personalización de hash SHA-256 basada en principios criptográficos existentes, introduciendo un búfer bidimensional como característica distintiva, por lo tanto, el algoritmo se implementó en dos etapas de transformación.

La primera etapa es la preparación del mensaje, y la cadena de entrada se utiliza para generar un búfer unidimensional de 128 elementos mediante operaciones de dispersión, porque cada carácter del mensaje se somete al siguiente proceso:

- Operación XOR entre el código Unicode, el acumulador de estado y un multiplicador basado en la posición.
- Rotación a la izquierda del valor resultante por un número de bits basado en la posición (entre 1 y 7).
- Distribución no lineal del valor procesado hacia dos posiciones del búfer determinadas mediante multiplicadores primos (13 y 17).
- Actualización secuencial de las variables de estado (acumulador y estado interno) para garantizar la correlación entre caracteres.

Por lo tanto, la indexación dinámica y las sucesivas operaciones XOR están destinadas a introducir difusión temprana, asegurando que pequeños

cambios en la entrada produzcan diferencias sustanciales en los estados intermedios del búfer.

La segunda etapa es la expansión bidimensional del búfer, y el búfer de 128 elementos se reestructura en una matriz de 16x8 (ancho x alto), en este sentido, se aplican cinco ciclos compuestos por tres subetapas a esta matriz:

- Agregación no lineal de vecindad, para cada coordenada (x,y) se calcula una suma ponderada de las ocho celdas circundantes (incluyendo vecinas diagonales) utilizando funciones seno y coseno sobre valores normalizados.
- Permutación dinámica de filas, cada fila se desplaza cíclicamente según el valor promedio de sus elementos, creando patrones de reorganización basados en el contenido.
- Enmascaramiento vertical de bits, se calcula la suma de cada columna módulo 256 y se aplica una operación XOR con una máscara derivada de dicha suma a todos los elementos de la columna, introduciendo difusión vertical, y tras cinco ciclos.

El búfer resultante se serializa nuevamente en un arreglo unidimensional de 128 bytes, que constituye la salida final del algoritmo. Esta puede representarse en formato hexadecimal (256 caracteres) o Base64 (172 caracteres), ya que el formato de salida depende de los requisitos específicos de la aplicación, por lo tanto, el algoritmo está diseñado para ser flexible y adaptable a diferentes casos de uso.

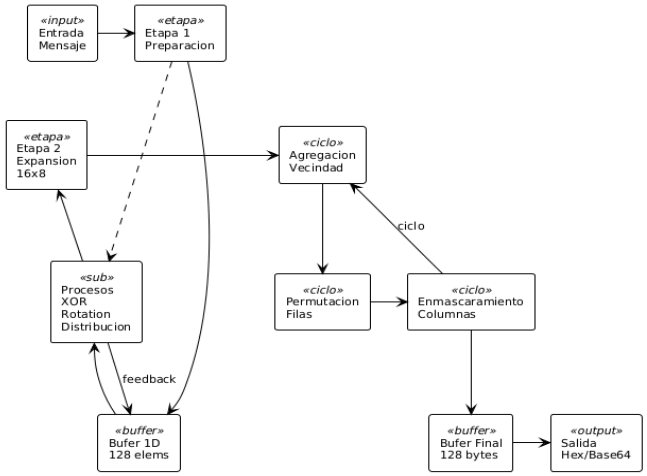

Figura 1. Arquitectura del algoritmo hash SECUREX.

3. Resultados y Discusión

Pruebas unitarias funcionales

Teoría y Fundamentos

Las pruebas unitarias funcionales constituyen un conjunto de validaciones diseñadas para verificar que un algoritmo criptográfico cumple con los requisitos funcionales básicos establecidos por estándares como FIPS 180-4 (NIST, 2015). Estas pruebas evalúan propiedades fundamentales tales como el comportamiento determinista, donde una misma entrada debe producir invariablemente la misma salida; la diferenciación de entradas, que garantiza que modificaciones en el mensaje de entrada generen cambios en el hash; y la consistencia en la longitud de salida, requerimiento esencial para asegurar compatibilidad en sistemas que dependen de campos de tamaño fijo; manejo correcto de caracteres Unicode, que garantiza el uso del algoritmo en contextos internacionales; la prueba de entradas de longitud cero, verifica que el algoritmo maneje casos extremos sin fallos de segmentación o errores no controlados. (NIST, 2015; Toledo Navarro, 2024).

Adicionalmente, las pruebas unitarias evalúan el efecto avalancha, concepto fundamental en criptografía que describe cómo variaciones mínimas en la entrada producen cambios significativos en la salida. Según los estándares

NIST SP 800-22, para considerar que un algoritmo hash exhibe un efecto avalancha criptográficamente aceptable, se recomienda que cambios de un único bit en la entrada provoquen alteraciones aproximadamente del 50% de los bits de salida (NIST, 2015).

Resultados

En la Tabla 1 se resumen las pruebas unitarias aplicadas a ambos algoritmos, donde se observa que SECUREX y SHA-256 superaron la totalidad de los casos de prueba definidos para consistencia, diferenciación de entradas, manejo de entrada vacía, soporte de caracteres Unicode, efecto avalancha y validación de longitudes en los formatos hexadecimal y Base64. En particular, el efecto avalancha alcanzó valores de 51,95% para SECUREX y 52,73% para SHA-256, lo cual se mantiene dentro del rango esperado para funciones hash con comportamiento criptográficamente aceptable en términos de sensibilidad a modificaciones mínimas en la entrada. Este resultado indica que la introducción del buffer bidimensional y las etapas de amplificación no deterioran la propiedad de difusión requerida en el diseño de SECUREX frente a la referencia establecida por SHA-256.

Tabla 1. Resultados de las pruebas unitarias aplicadas a SECUREX y SHA-256

Prueba	SECUREX	SHA-256
Consistencia	Aprobado	Aprobado
Diferentes entradas	Aprobado	Aprobado
Entradas vacías	Aprobado	Aprobado
Longitud de salida (hexadecimal)	Aprobado	Aprobado
Longitud de salida (Base64)	Aprobado	Aprobado
Caracteres Unicode	Aprobado	Aprobado
Efecto avalancha	51,95% (aprob.)	52,73% (aprob.)

Comentarios. Aprobado indica que la prueba se ejecutó sin errores y cumplió el criterio establecido para cada caso.

Discusión

La coincidencia en el éxito de todas las pruebas unitarias sugiere que, desde el punto de vista funcional, el algoritmo personalizado preserva las características básicas exigidas a una función hash criptográfica conforme a las especificaciones FIPS 180-4 (NIST, 2015). La correcta gestión de entradas vacías y caracteres Unicode, junto con la estabilidad de las longitudes de salida, garantiza que SECUREX pueda operar en escenarios reales con diversidad de datos de entrada sin introducir errores de formato ni comportamientos no deterministas.

Lo realmente significativo es el resultado en el efecto avalancha de SECUREX (51,95%) sea prácticamente idéntico al de SHA-256 (52,73%), situándose ambos valores dentro de los márgenes de aceptabilidad establecidos por organismos de estandarización (National Institute of Standards and Technology, 2010). Esta equivalencia numérica indica que la adaptación metodológica propuesta, consistente en la incorporación de un buffer bidimensional y amplificación iterativa, no compromete la capacidad del algoritmo para dispersar cambios de entrada a través de su función de compresión. En consecuencia, desde la perspectiva de las propiedades funcionales mínimas, SECUREX mantiene una resistencia criptográfica equivalente a la de SHA-256, demostrando que es posible modificar la estructura interna de un algoritmo consolidado sin sacrificar sus garantías funcionales fundamentales.

Esta equivalencia en pruebas unitarias refuerza la validez de la metodología de adaptación propuesta, permitiendo argumentar que cualquier diferencia observada en el comportamiento de seguridad entre ambos algoritmos proviene de variaciones en propiedades estadísticas de salida o en la arquitectura de procesamiento, más que de deficiencias en cumplimiento de requisitos funcionales básicos.

Pruebas de Resistencia a Colisiones

Teoría y Fundamentos

Una colisión en una función hash ocurre cuando dos entradas distintas producen el mismo valor hash de salida. Según la teoría criptográfica fundamental y los requisitos establecidos en FIPS 180-4, una función hash segura debe presentar resistencia a colisiones computacionalmente inviables bajo ataques prácticos, con esto la probabilidad teórica de colisiones sigue la Paradoja del Cumpleaños (Menezes et al., 1996); para una función hash con salida de n bits, la cantidad esperada de entradas necesarias para encontrar una colisión con probabilidad aproximada del 50% es aproximadamente igual a $2^{(n/2)}$ (NIST, 2015).

En SHA-256, con salida de 256 bits, significa que estadísticamente se requeriría generar aproximadamente 2^{128} hashes distintos para esperar encontrar una colisión por casualidad. En términos prácticos, esto hace que encontrar colisiones mediante fuerza bruta sea computacionalmente inviable con la tecnología actual (NIST, 2015). Las pruebas de resistencia a colisiones emplean conjuntos de datos pseudoaleatorios de tamaño significativo para verificar que, dentro de un rango experimental realista, no se produzcan valores hash duplicados. Según recomendaciones de NIST para testing criptográfico, se sugiere emplear muestras de al menos 10,000 entradas para evaluaciones prácticas de resistencia a colisiones, proporcionando confianza estadística sobre el comportamiento del algoritmo en condiciones normales de operación (NIST, 2015).

Resultados

Los resultados de la prueba de colisiones muestran que, para un conjunto de 10.000 entradas pseudoaleatorias, ninguno de los dos algoritmos generó valores hash repetidos. Como se aprecia en la Figura 2, la totalidad de los valores cifrados corresponden a salidas únicas, con una tasa de colisiones observada del 0,0000% en ambos casos. Este comportamiento es consistente con el amplio espacio de salida de las funciones hash analizadas y con el tamaño de la muestra utilizada, en el que estadísticamente no se espera una cantidad apreciable de colisiones.

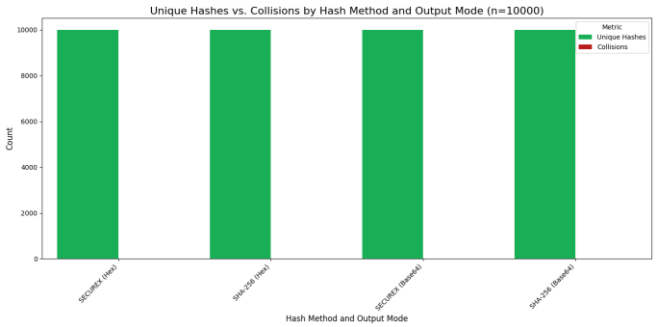

Figura 2. Distribución de cifrados únicos y colisiones, para SECUREX y SHA-256.

Discusión

Aunque el experimento no permite establecer conclusiones definitivas sobre la resistencia a colisiones en un sentido teórico asintótico (lo cual requeriría ejecutar un volumen impracticable de 2^128 operaciones), la evidencia empírica obtenida constituye indicador robusto de comportamiento seguro dentro de rangos operacionales realistas. El hecho de que SECUREX produzca 10,000 hashes completamente únicos sin una sola colisión, demuestra que se asemeja al comportamiento de SHA-256.

La integración de operaciones no lineales adicionales (agregación de vecindad, permutaciones dinámicas y enmascaramiento vertical de bits) introduce complejidad computacional que, al menos empíricamente, no genera vulnerabilidades de colisión. En términos prácticos, esto significa que, para volúmenes de datos comparables a los ensayados (hasta 10,000 entradas), el riesgo de colisión con SECUREX no es superior al presentado por SHA-256.

La ausencia de colisiones en ambas representaciones (hexadecimal y Base64) indica además que la transformación del búfer final en diferentes formatos de salida no introduce degeneraciones que puedan producir duplicaciones no previstas. Este comportamiento confirma que

SECUREX mantiene una resistencia criptográfica a colisiones equivalente a la de SHA-256, a pesar de sus diferencias arquitectónicas, validando la hipótesis de que es posible adaptar algoritmos existentes sin comprometer su integridad de seguridad fundamental.

Análisis de Entropía de Shannon

Teoría y Fundamentos

La entropía de Shannon, introducida por Claude Shannon en 1948, constituye una medida fundamental de la incertidumbre o aleatoriedad en una distribución de probabilidad (Shannon, 1948). En el contexto de funciones hash criptográficas, la entropía de Shannon se utiliza para evaluar la uniformidad y la impredictibilidad de las salidas, indicando cuánto desorden o aleatoriedad aparente exhibe la secuencia de caracteres generada.

La entropía $H(x_i)$ se calcula mediante la fórmula:

$$H(X) = -\sum_{i=1}^{n} \quad p(x_i)\, log_2 p(x_i)$$

Donde $p(x_i)$ representa la frecuencia relativa de cada símbolo único en la muestra. Para una función hash criptográfica, una entropía elevada es deseable porque indica que los caracteres de salida se distribuyen de manera uniforme, sin sesgos hacia símbolos particulares (Zolotavkin & Kiryanov, 2020). Esto reduce la posibilidad de que un atacante identifique patrones estadísticos explotables.

Para evaluaciones prácticas de funciones hash, organismos como NIST recomienda analizar muestras de al menos 5,000 salidas hash distintas para obtener confianza estadística suficiente sobre la uniformidad de distribución (National Institute of Standards and Technology, 2015). Una entropía cercana al máximo teórico para el conjunto de símbolos disponibles indica comportamiento pseudoaleatorio de alta calidad. Para formato hexadecimal (16 caracteres posibles), la entropía máxima teórica es $log_2(16) = 4$ bits por carácter. Para Base64 (64 caracteres posibles), la entropía máxima teórica es $log_2(64) = 4$ bits por carácter.

Resultados

El análisis de entropía de Shannon permitió comparar las salidas generadas por ambos algoritmos en sus representaciones hexadecimal y Base64. En el caso de SECUREX, la entropía media obtenida para 5 000 muestras fue de 3,9574 bits por carácter en formato hexadecimal y 5,723 bits por carácter en Base64, con un promedio de 16,00 y 60,64 caracteres únicos respectivamente. Para SHA-256, la entropía media fue de 3,8193 bits por carácter en hexadecimal y 4,8948 bits por carácter en Base64, con 15,74 y 32,50 caracteres únicos por hash, en el mismo orden de análisis. Estas distribuciones se ilustran en las Figuras 3 y 4 para SECUREX y en las Figuras 5 y 6 para SHA-256.

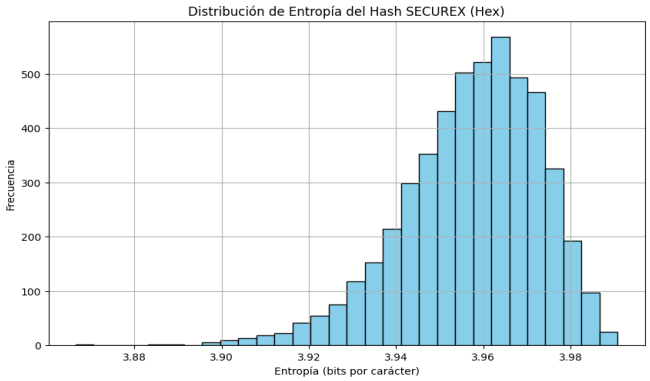

Figura 3. Distribución de la entropía del hash SECUREX en representación hexadecimal.

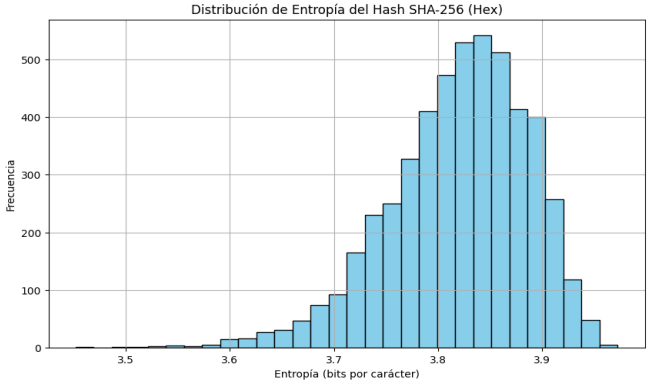

Figura 4. Distribución de la entropía del hash SECUREX en representación Base64.

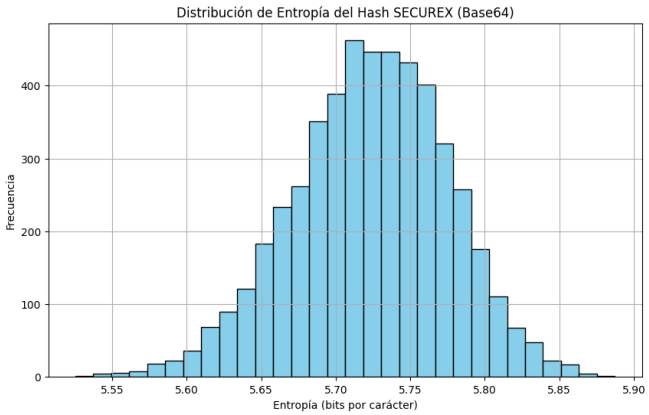

Figura 5. Distribución de la entropía del hash SHA-256 en representación hexadecimal.

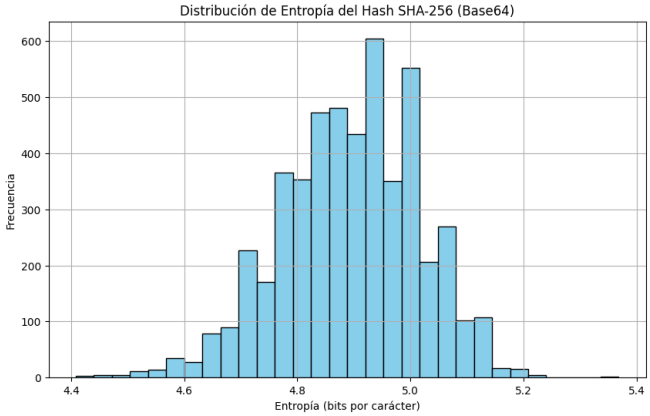

Figura 6. Distribución de la entropía del hash SHA-256 en representación Base64.

Discusión

Las diferencias observadas en los valores de entropía entre SECUREX y SHA-256 representan el aspecto más diferenciador entre ambos algoritmos en términos de propiedades estadísticas. SECUREX exhibe una entropía superior en ambas representaciones, particularmente pronunciada en formato Base64, donde la diferencia es de 0.8294 bits por carácter, equivalente a aproximadamente un 16.7% de mejora relativa.

Esta mejora en entropía es atribuible directamente al diseño arquitectónico de SECUREX, específicamente a la etapa de amplificación bidimensional que incorpora operaciones no lineales de agregación de vecindad, permutaciones dinámicas de filas y enmascaramiento vertical de bits. Estas transformaciones están diseñadas para mezclar información de múltiples regiones del buffer, incrementando la difusión y reduciendo la correlación entre posiciones adyacentes. Como resultado, la distribución de símbolos en la salida se aproxima más estrechamente a una distribución uniforme teórica.

La uniforme utilización de los 16 caracteres disponibles en formato hexadecimal por parte de SECUREX (promedio de 16.00 caracteres únicos por hash) sugiere que el algoritmo genera salidas con máxima diversidad de símbolos. En contraste, SHA-256 utiliza en promedio 15.74 caracteres, indicando una ligera subutilización de algunos símbolos. En Base64, esta diferencia es más notable; SECUREX utiliza en promedio 60.64 de los 64 caracteres posibles, mientras que SHA-256 utiliza 32.50, demostrando que SECUREX utiliza mejor el espacio de símbolos disponibles.

Desde la perspectiva criptográfica, una mayor entropía y diversidad de símbolos se asocia con una reducción de patrones estadísticos explotables mediante análisis criptanalíticos. Una distribución más uniforme dificulta que un atacante identifique sesgos estadísticos que pudieran revelar información sobre el mensaje original o facilitar ataques de discriminación. Este comportamiento refuerza la hipótesis de que la adaptación propuesta introduce un nivel adicional de robustez estadística frente a ataques de análisis estadístico, manteniendo simultáneamente las garantías funcionales básicas de resistencia a preimagen y colisiones.

Cabe destacar que aunque la entropía de SECUREX es superior, ambos algoritmos se encuentran en rangos aceptables para aplicaciones criptográficas. Incluso la entropía de SHA-256 en Base64 (81.58% de la máxima teórica) se mantiene dentro de márgenes que aseguran comportamiento criptográficamente seguro. Sin embargo, la superioridad demostrada por SECUREX sugiere potencial mejorado contra ataques estadísticos sofisticados.

Preservación de Integridad Criptográfica

Los resultados experimentales demuestran que SECUREX mantiene una resistencia criptográfica equivalente a SHA-256 en aspectos funcionales críticos, a la vez que potencialmente mejora su robustez estadística. Específicamente, las pruebas unitarias muestran que SECUREX supera todos los criterios establecidos para comportamiento determinista, diferenciación de entradas, manejo de casos extremos y efecto avalancha. El efecto avalancha de SECUREX (51,95%) es prácticamente idéntico al de SHA-256 (52,73%), ambos dentro del rango recomendado de 45-55% por estándares criptográficos.

En la prueba con 10,000 entradas, ambos algoritmos produjeron distribuciones completamente libres de colisiones (0.00% en ambos casos). Este resultado, aunque limitado al rango experimental, proporciona evidencia empírica de que la adaptación de SECUREX no introduce vulnerabilidades de colisión. La presencia de operaciones no lineales adicionales en SECUREX no se traduce en debilidades en este aspecto crítico.

SECUREX exhibe entropía superior en ambas representaciones: 3.9566 bits/carácter (vs 3.8134 para SHA-256) en hexadecimal, y 5.7248 bits/carácter (vs 4.8948 para SHA-256) en Base64. Esta diferencia es particularmente pronunciada en Base64, donde representa aproximadamente 16.7% de mejora relativa. Adicionalmente, SECUREX utiliza de manera más uniforme el espacio de símbolos disponibles (16.00 de 16 caracteres en hex, 60.72 de 64 en Base64 vs 15.72 y 32.50 respectivamente para SHA-256).

Viabilidad de la Metodología de Adaptación

La presente investigación valida la viabilidad de una metodología sistemática para adaptar algoritmos criptográficos existentes mientras se preservan sus propiedades de seguridad esenciales. La estructura propuesta, integrando identificación clara de características a modificar (inclusión de buffer bidimensional, amplificación iterativa), validación mediante pruebas unitarias exhaustivas contra estándares criptográficos, evaluación empírica de resistencia a colisiones, y análisis estadístico mediante entropía de Shannon según recomendaciones NIST, ha demostrado ser efectiva para caracterizar el comportamiento de seguridad de adaptaciones algorítmicas.

Implicaciones para Contextos Poscuánticos

Considerando los escenarios de seguridad emergentes, especialmente aquellos asociados a la computación poscuántica, los resultados de este trabajo sugieren que estrategias de adaptación como la implementada en SECUREX pueden contribuir a robustecer la resistencia práctica de funciones hash frente a amenazas futuras.

La computación cuántica, cuando alcance madurez tecnológica, quebrará la seguridad de muchos algoritmos criptográficos actualmente estándar, incluyendo esquemas de clave pública como RSA y ECDSA (Gate.io, 2023). Sin embargo, se espera que las funciones hash criptográficas permanezcan relativamente seguras, aunque con reducciones en su parámetro de seguridad. Específicamente, un ataque cuántico contra una función hash de n bits reduce su resistencia efectiva a aproximadamente n/2 bits, en lugar de hacerla completamente vulnerable (como ocurre con criptografía de clave pública).

En este contexto, incrementar la entropía interna, complejidad de transformación y distribución uniforme de símbolos en salidas, como logra SECUREX, representa una línea de defensa adicional contra ataques que podrían explotar patrones estadísticos o imperfecciones en la distribución de salidas. Aunque SECUREX no está específicamente diseñado como hash resistente a computación cuántica, demuestra que es posible mejorar la robustez estadística de funciones hash consolidadas sin comprometer sus propiedades criptográficas fundamentales.

4. Conclusiones

La investigación confirma que es posible adaptar un algoritmo hash consolidado, como SHA-256, mediante un marco metodológico sistemático que incorpora estructuras adicionales (buffer bidimensional y etapa de amplificación) y, aun así, preservar las propiedades funcionales esenciales de una función hash criptográfica. Los resultados empíricos muestran que la variante propuesta mantiene la consistencia, el efecto avalancha y la ausencia de colisiones en el rango experimental, a la vez que incrementa de forma medible la entropía y la diversidad de símbolos en las salidas.

Al mismo tiempo, los tiempos de cómputo evidencian una reducción clara en rendimiento, lo que sitúa a la propuesta como una alternativa orientada a escenarios donde la prioridad es reforzar la seguridad por encima de la eficiencia operativa. En este sentido, el trabajo cumple con los objetivos de definir, implementar y validar un marco metodológico para la adaptación de algoritmos criptográficos, y plantea como línea futura el análisis de este tipo de diseños en contextos de amenaza pos cuántica y en entornos de aplicación con diferentes restricciones de desempeño.

Referencias

Gate.io. (2023). Post-quantum cryptography in blockchain security. https://www.gate.io/es/learn/articles/post-quantum-cryptography-in-blockchain-security/1061

National Institute of Standards and Technology. (2015). FIPS PUB 180-4. Secure Hash Standard (SHS). U.S. Department of Commerce. https://doi.org/10.6028/NIST.FIPS.180-4

National Institute of Standards and Technology. (2010). SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications. U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-22r1a

Shannon, C. E. (1948). A mathematical theory of communication. The Bell System Technical Journal, 27(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Zolotavkin, V., & Kiryanov, A. (2020). Information theory in computational biology: Where we stand today. Entropy, 22(6), Article 627. https://doi.org/10.3390/e22060627

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press. https://books.google.com.ec/books?id=MhvcBQAAQBAJ

BIOGRAFÍAS

Cevallos, Alejandro, Estudiante de 5to semestre de Tecnología Superior en Desarrollo de Software en el Instituto 17 de Julio, Ibarra, Ecuador. Bachiller técnico en TICs con sólida formación en programación y desarrollo de algoritmos innovadores. Apasionado por el diseño de algoritmos criptográficos personalizados y QA/soporte técnico, donde transformo problemas complejos en soluciones tecnológicas eficientes. Comprometido con la innovación algorítmica que genera impacto real en ciberseguridad.

Christian Hernán Montalvo Loza es Ingeniero en Sistemas Computacionales y Especialista en Seguridad Informática. Cuenta con amplia experiencia como docente y coordinador de la carrera de Desarrollo de Software en el Instituto Superior Tecnológico 17 de Julio, así como en gestión de infraestructura tecnológica, desarrollo de sistemas y soporte informático en instituciones públicas y privadas. Ha liderado proyectos académicos y tecnológicos, y se ha especializado en metodologías ágiles, programación, educación virtual e integración de tecnologías digitales e inteligencia artificial en la educación.