

**UNIVERSIDAD TECNOLÓGICA DE LA SELVA**

**CARRERA:** Ingeniería en Desarrollo y Gestión de Software

**NOMBRE DE LA ASIGNATURA:** Seguridad Informática

**UNIDAD TEMÁTICA:** III. Criptografía

**ACTIVIDAD DE EVALUACIÓN:** Investigación documental

**INTEGRANTES DEL EQUIPO:**

|  |  |
| --- | --- |
| Matrícula | Nombre |
| 091910036 | Arroyo Ruiz Víctor Iván |
| 091910237 | Encino Gómez Edgardo Alexander |
| 091910039 | Guíllen Navarro Leonardo Antonio |
| 091910151 | Domínguez Santiz Luis Ángel |

**GRADO:** 7° **GRUPO:** “A”

**NOMBRE DEL PROFESOR:** Mtro. Jesús Domínguez Gutú

**FECHA DE ENTREGA:** 17/09/2021

**ÍNDICE**

[Algoritmos Simétricos 1](#_Toc82639021)

[DES 1](#_Toc82639022)

[3DES 2](#_Toc82639023)

[RC2 3](#_Toc82639024)

[RC4 4](#_Toc82639025)

[RC5 5](#_Toc82639026)

[IDEA 6](#_Toc82639027)

[Blowfish 7](#_Toc82639028)

[Twofish 7](#_Toc82639029)

[AES 9](#_Toc82639030)

[Aplicaciones de los algoritmos simétricos 10](#_Toc82639031)

[Algoritmos Asimétricos 11](#_Toc82639032)

[DSA 11](#_Toc82639033)

[ElGamal 11](#_Toc82639034)

[RSA 11](#_Toc82639035)

[Aplicaciones de los algoritmos asimétricos 12](#_Toc82639036)

[Algoritmos HASH 13](#_Toc82639037)

[MD5 13](#_Toc82639038)

[SHA 13](#_Toc82639039)

[SHA1 14](#_Toc82639040)

[RIPE-MD 15](#_Toc82639041)

[Aplicaciones de los algoritmos HASH 15](#_Toc82639042)

[Bibliografía 16](#_Toc82639043)

# Algoritmos Simétricos

## DES

Nació en los años 70 y fue un algoritmo de cifrado muy utilizado hasta no hace mucho, no caracterizándose precisamente por su seguridad. Desarrollado originalmente por IBM a requerimiento del NBS (National Bureau of Standards, Oficina Nacional de Estandarización, en la actualidad denominado NIST, National Institute of Standards and Technology, Instituto Nacional de Estandarización y Tecnología) de EE.UU. y posteriormente modificado y adoptado por el gobierno de EE.UU.

Utiliza cifrado por bloques con bloques de 64 bits, esto es, toma un texto plano de esa longitud y lo transforma, mediante una serie de operaciones, en texto cifrado de la misma longitud. Se utilizan claves de 64 bits, de los cuales solo se utilizan 56, para realizar el cifrado de los bloques.

El resto llevan información de paridad. Esta longitud tan corta se considera insuficiente para protegerse frente a ataques de fuerza bruta y es uno de los motivos por los que se considera inseguro, ya que estas claves se han llegado a romper en 24 horas. Aun así, se sigue utilizando en las transacciones realizadas en cajeros automáticos.

DES tiene 19 etapas diferentes. La primera etapa es una transposición, una permutación inicial del texto plano de 64 bits, independientemente de la clave. La última etapa es otra transposición , exactamente la inversa de la primera. La penúltima etapa intercambia los 32 bits de la izquierda y los 32 de la derecha. Las 16 etapas restantes son una Red de Feistel de 16 rondas. En cada una de las 16 iteraciones se emplea un valor, , obtenido a partir de la clave de 56 bits y distinto en cada iteración. (Arriazu, 1999)

## 3DES

Desarrollado por Tuchman en 1978, es una manera de mejorar la robustez del algoritmo DES que consiste en aplicarlo tres veces consecutivas. Se puede aplicar con la misma clave cada vez, o con claves distintas y combinando el algoritmo de cifrado con el de descifrado, lo cual da lugar a DES-EEE3, DES-EDE3, DES-EEE2 y DES-EDE2. El resultado es un algoritmo seguro y que se utiliza en la actualidad, aunque resulta muy lento comparado con otros algoritmos más modernos que también son seguros.

El 3DES usa tres claves y tres ejecuciones del algoritmo DES. La función sigue la secuencia cifrar-descifrar-­cifrar (EDE: encrypt-­decrypt­-encrypt): (Stallings, 2011)

Donde

= texto cifrado

= texto claro

= cifrado de usando la clave

= descifrado de usando la clave

## RC2

El RC2 es un algoritmo de cifrado por bloques de clave de tamaño variable diseñado por Ron Rivest de RSA Data Security (la RC quiere decir Ron's Code o Rivest's Cipher) en en 1987. El algoritmo trabaja con bloques de 64 bits y entre dos y tres veces más rápido que el DES en software. Se puede hacer más o menos seguro que el DES contra algoritmos de fuerza bruta eligiendo el tamaño de clave apropiadamente. (Bibing, 2021)

Sus características principales son:

* Emplea la función de Feistel, división en bloques y la aplicación en s-cajas.
* Este algoritmo utiliza una clave de 64 bits
* Este algoritmo ya no es usado debido a sido revelado y vulnerado en internet de forma anónima
* Este algoritmo trabaja con bloques de 64 bits y es entre dos otres veces más rápido que la DES en software. Se puede hacer más o menos seguro que el DES contra algoritmos de fuerza bruta eligiendo el tamaño de clave apropiadamente.

## RC4

El RC4 es un algoritmo de cifrado de flujo diseñado por Ron Rivest para RSA Data Security. Presentado en diciembre de 1994, fue diseñado para el cifrado en flujo y permite trabajar con claves de tamaño variable. (Bibing, 2021)

RC4 o ARC4 es parte de los protocolos de encriptación más comunes como WEP, WPA para dispositivos wireless y TLS. Utilizando 40 y 128 bits correspondientes a la clave secreta de encriptación y los 24 y 48 bits al vector de inicialización, por lo tanto, WEP utiliza los 64 bits y WPA los 128 bits.

El algoritmo RC4 utiliza un vector de inicialización (VI), este es generado dinámicamente y debería ser diferente para cada trama. El objetivo perseguido con el Vector de Inicialización es cifrar con claves diferentes para impedir que un posible atacante pueda capturar suficiente tráfico cifrado con la misma clave y terminar finalmente deduciendo la clave. Como es lógico, ambos extremos deben conocer tanto la clave secreta como el Vector de Inicialización. Lo primero sabemos ya que es conocido puesto que está almacenado en la configuración de cada elemento de red. El Vector de Inicialización, en cambio, se genera en un extremo y se envía en la propia trama al otro extremo, por lo que también será conocido. (Proaño)

## RC5

Aunque es muy seguro, se puede mejorar su efectividad modificando sus modos de operación: RC5-cifrador en bloque, RC5-CBC, RC5-CBC-relleno, RC5-CTS.

Diseñado por Ronald Rivest en 1994, el tamaño variable de bloques es de: 32, 64 o 128 bits. Palabra clave entre 0 y 2040 bits. vueltas entre 0 y 255. bloques de 64 bits (2 palabras de 32 bits), en 12 rondas o vueltas y con una clave de 128 bits (16 bytes). RC5 hace uso de rotaciones dependientes. En su estructura contiene algunas operaciones como sumas modulares y operaciones XOR.

Adecuado para ser implementado en hardware y software. Utiliza computación básica y operaciones que podemos encontrar en un microprocesador. Adaptable a procesadores de diferentes tamaños de palabra; puede utilizar tamaño de palabra de 16, 32 y 64 bits. Bajo consumo de memoria; uso en tarjetas inteligentes o dispositivos donde la disponibilidad de espacio sea un factor. Proporciona alta seguridad.

Utiliza la clave secreta del usuario para expandir un arreglo S de claves, que contiene 2(r + 1) palabras aleatorias determinada por la clave original. (Docplayer, 2021)

## IDEA

Fue creado por Xuejia Lai y James Massey en 1990. Es un cifrado de bloque, que opera sobre mensajes de 64 bits con una clave de 128 bits.

Es un algoritmo descrito por primera vez en 1991 y propuesto para reemplazar al DES. Este algoritmo trabaja con bloques de 64 bits y utiliza una clave de 128 bits, lo que hace que sea inmune al criptoanálisis diferencial. Además, el enorme número de posibles claves que hay que analizar hace que, con el estado actual de la computación, sea imposible de averiguar a través de ataques por fuerza bruta. Al igual que ocurría con DES, en IDEA se usa el mismo algoritmo para el cifrado y el descifrado.

Consiste en ocho transformaciones idénticas (cada una llamada una ronda) y una transformación de salida (llamada media ronda). Gran parte de la seguridad de IDEA deriva del intercalado de tres operaciones: Operación O-exclusiva (XOR) bit a bit, suma módulo 216 y multiplicación módulo 216+1. (Sanjuan, 2011)

## Blowfish

Es un algoritmo de cifrado por bloques de 64 bits diseñado por Bruce Schneier en 1993.

Utiliza claves de longitud variable entre 32 y 448 bits. A pesar de utilizar un tamaño de bloque pequeño, que podría facilitar su vulnerabilidad al procesar textos largos, se considera un algoritmo seguro y es más rápido que DES.

Este algoritmo se diseñó para máquinas de 32 bits y es considerablemente más rápido que el DES. El algoritmo es considerado seguro, aunque se han descubierto algunas claves débiles, un ataque contra una versión del algoritmo con tres rotaciones y un ataque diferencial contra una variante del algoritmo. (Chinta., 2015)

## Twofish

Es una variante de Blowfish que utiliza bloques de 128 bits y claves de 256 bits. También diseñado por Bruce Schneier, en colaboración con John Kelsey, Doug Whiting, David Wagner, Chris Hall, y Niels Ferguson, fue uno de los cinco finalistas en el proceso de selección de NIST para sustituir a DES como algoritmo estándar.

Su funcionamiento se detalla a continuación. Divide el texto plano en 4 grupos de 32 bytes. En cada una de ellas las 2 palabras de la izquierda son usadas como entradas de una función g, una de ellas es rotada 8 bits a la izquierda. Se les aplica whitening al inicio con cuatro llaves. Utiliza 40 subllaves de 32 bits. Las primeras ocho son usadas para whitening. Cuatro al principio y cuatro al final son utilizadas en un xor con el bloque entero, cada iteración usa dos de los 32 restantes subllaves, por lo que Twofish realiza 16 iteraciones. El resultado de las dos funciones g son mezcladas unas con otras a través de una Pseudo-Hadamard Transform (PHT), más dos llaves. A ese resultado se realiza una suma (XOR) con la parte derecha, una de las cuáles son rotadas un bit a la izquierda y la otra a la derecha. Las mitades izquierda y derecha son intercambiadas para la siguiente iteración. Cuando finaliza el proceso, el intercambio de la última iteración es revertido y se le suman las 4 llaves restantes para producir el texto cifrado. Twofish ha recibido críticas por su complejidad haciendo difícil su análisis durante el tiempo establecido en el proceso de desarrollo de AES. (Schneier, 2000)

## AES

Es un algoritmo de cifrado simétrico desarrollado por los estudiantes Vincent Rijmen y Joan Daemen de la Katholieke Universiteit Leuven en Bélgica, bajo el nombre "Rijndael" fue presentado en 1997 al concurso organizado por el Instituto Nacional de Normas y Tecnologías (NIST) para elegir el mejor algoritmo de cifrado; el algoritmo gano el concurso transformándose en un estándar en el año 2002, con algunos cambios fue posteriormente renombrado AES (Advanced Encryption Standard) y se convirtió en uno de los algoritmos más utilizados en la actualidad.

El estándar de encriptación avanzada es uno de los algoritmos más populares de clave simétrica y, de hecho, remplazará al DES utilizado habitualmente. Es rápido y eficiente y proporciona una encriptación segura utilizando un cifrado por bloques, con bloques de 128 bits y claves de 128, 192 o 256 bits. Se utiliza fundamentalmente en aplicaciones bancarias por Internet, comunicaciones inalámbricas, protección de datos en discos duros, etc.

AES toma como elemento básico al byte (8 bits) y ve a los bytes como elementos del campo finito de Galois o GF(), toda operación del algoritmo está basada en operaciones sobre este campo finito, rotaciones de bytes y operaciones de suma módulo 2. (Pousa, 2011)

## Aplicaciones de los algoritmos simétricos

* Cajeros automáticos
* Aplicaciones bancarias por Internet
* Comunicaciones inalámbricas
* Protección de datos en discos duros
* Correo electrónico
* Firmas digitales
* DNI electrónico
* Cifrado de la información
* SSL Y TTL

# Algoritmos Asimétricos

## DSA

Digital Signature Algorithm = Algoritmo de firma digital.

Se utiliza ampliamente como un algoritmo de firma digital, es actualmente un estándar, pero DSA no se utiliza para cifrar datos, solamente como firma digital. Este algoritmo se utiliza ampliamente en las conexiones SSH para comprobar la firma digital de los clientes, además, existe una variante de DSA basada en curvas elípticas (ECDSA), y está disponible en todas las librerías criptográficas actuales como OpenSSL, GnuTLS o LibreSSL. Otra característica de DSA es la longitud de clave, la mínima longitud de clave es de 512 bits, aunque lo más habitual es usar 1024 bits. (López, 2021)

## ElGamal

Es un algoritmo, procedimiento o esquema de cifrado basado en problemas matemáticos de logaritmos discretos. Usado en la criptografía asimétrica. ElGamal consta de tres componentes: el generador de claves, el algoritmo de cifrado, y el de descifrado. A continuación, se describe el algoritmo utilizando el grupo multiplicativo de enteros módulo p.

 Hasta el momento el algoritmo ElGamal de cifrado/descifrado puede ser considerado un algoritmo efectivo. Un adversario con la habilidad de calcular logaritmos discretos podría ser capaz de romper un cifrado ElGamal. (Ecured, 2011)

## RSA

**Este algoritmo se basa en la pareja de claves,** la pública y la privada. La seguridad de este algoritmo radica en el problema de la factorización de números enteros muy grandes, y en el problema RSA, porque descifrar por completo un texto cifrado con RSA no es posible actualmente, aunque sí un descifrado parcial. Algunas características muy importantes de RSA es la longitud de clave, actualmente como mínimo se debe utilizar una longitud de 2048 bits, aunque es recomendable que sea de 4096 bits o superior para tener una mayor seguridad. (López, 2021)

## Aplicaciones de los algoritmos asimétricos

Se utilizan los métodos asimétricos para iniciar una comunicación segura durante la cual se establece una clave secreta generada aleatoriamente y se acuerda un algoritmo de cifrado simétrico a partir de las preferencias de los dos interlocutores. A partir de ese momento la comunicación tiene lugar por medio de algoritmos simétricos que son más eficientes.

Una de las aplicaciones más extendidas de las técnicas de cifrado en la comunicación en modo seguro del navegador de Internet, es decir cuando el usuario pone https en lugar de http. Como se verá más adelante en este mismo artículo, al establecer una conexión https el navegador solicita la clave pública del servidor y luego se establece la comunicación mediante algoritmos simétricos. En el caso del correo electrónico cifrado, se sigue el estándar S/MIME que se basa en las normas PKCS#7. Análogamente se aplican cifrados simétricos con claves autogeneradas y luego dichas claves se cifran con algoritmos asimétricos (Delgado, 2006)

# Algoritmos HASH

## MD5

MD5 (Message Digest Algorithm 5) fue diseñado por Ron Rivest en 1991. MD5 procesa una longitud variable de mensaje en una salida de longitud fija de 128 bits. MD5 es una función hash popular. Funciona en bloques de 512 bits y procesa cada bloque a través de 4 rondas, donde cada ronda en turn procesa 16 sub-bloques (cada uno de 32 bits). En el 512 bit El mensaje se divide en 16 sub-bloques antes de su procesamiento.

Es altamente inseguro almacenar contraseñas en texto plano en la base de datos. Para aumentar la seguridad de las contraseñas, se pueden utilizar algoritmos MD5 para hacer un hash de las contraseñas originales y los valores hash, en lugar del texto plano, se almacenan en la base de datos. Durante la autenticación, la contraseña de entrada también es de forma similar, y el valor hash resultante se compara con el valor hash de la base de datos. (Kioon, 2013)

## SHA

SHA-0 es una función hash dedicada de 160 bits basada en el principio de diseño de MD4. Aplica el paradigma Merkle-Damg˚ard a una función de compresión dedicada de compresión. El mensaje de entrada se rellena y se divide en bloques de mensaje de 512 bits. En cada iteración de la función de compresión h, una variable de encadenamiento de 160 bits Ht se actualiza utilizando un bloque de mensajes Mt+1, es decir, Ht+1 = h(Ht, Mt+1). El valor inicial H0 (también llamado IV) está predefinido y Hk es la salida de la función hash. La función de compresión SHA-0 se basa en la construcción Davis-Meyer. (Stephane, 2021)

La familia SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el National Institute of Standards and Technology (NIST). El primer miembro de la familia fue publicado en 1993 es oficialmente llamado SHA. Sin embargo, hoy día, no oficialmente se le llama SHA-0 para evitar confusiones con sus sucesores. Dos años más tarde el primer sucesor de SHA fue publicado con el nombre de SHA-1. Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (todos ellos son referidos como SHA-2).

En 1998, un ataque a SHA-0 fue encontrado, pero no fue reconocido para SHA-1, se desconoce si fue la NSA quien lo descubrió, pero aumentó la seguridad del SHA-1.

## SHA1

El algoritmo SHA-1 (Secure Hash Algorithm) fue publicado por NIST (National Institute of Standards and Technology) en 1993, siendo una versión mejorada del SHA. SHA-1 es una función hash basada en el algoritmo MD4, por lo que tiene similitudes con el MD5.

SHA-1 es una función hash constituida por un búfer estado de 160 bits y trabaja con cuatro rondas conformadas por operaciones elementales de 32 bits. En lugar de procesar cada bloque del mensaje cuatro veces, SHA-1 utiliza una recurrencia lineal para utilizar 80 palabras de las 16 palabras de entrada del bloque que se está procesando. Esta recurrencia lineal asegura que cada bit del mensaje afectará las funciones internas al menos una docena de veces. La salida de SHA-1 es un resumen de 160 bits. (Algredo-Badillo, 2021)

## RIPE-MD

Es un algoritmo de primitivas de integridad de resumen del mensaje, es un algoritmo que reemplaza a MD4 y MD5 fue diseñado por Hans Dobbertin, Antoon Bosselaers y Bart Preneel publicados primeramente en 1996. Genera un hash de 20 bytes (160 bits). La función de este algoritmo es disminuir la posibilidad de colisiones hash accidentales. (Caza, 2015)

También existen versiones de 128, 256 y 320 bits de este algoritmo, llamadas RIPEMD-128, RIPEMD-256 y RIPEMD-320 respectivamente. La versión 128 bits fue pensada solamente como un reemplazo para el RIPEMD original, que eran también de 128 bits, y en la que habían sido encontradas razones para cuestionar su seguridad. Las versiones de 256 y 320 bits solamente disminuyen la posibilidad de colisiones *hash* accidentales, y no tienen niveles más altos de seguridad con respecto a RIPEMD-128 y RIPEMD-160.

## Aplicaciones de los algoritmos HASH

1. Las funciones criptográficas hash juegan un papel fundamental en la criptografía moderna. Hay muchas funciones hash, comúnmente usadas en aplicaciones no criptográficas. Nosotros nos referiremos siempre a las que admiten el adjetivo de criptográficas, y las llamaremos habitualmente simplemente funciones hash.
2. Las funciones hash se emplean en criptografía junto con los criptosistemas de firma digital para otorgar integridad a los datos. A la hora de firmar digitalmente un documento o mensaje, es práctica habitual hacer la firma sobre la huella digital del mensaje y no sobre la totalidad del mensaje a firmar.
3. Comprobación de integridad de ficheros.
4. Seguridad en procesos de identificación en sistemas.
5. Firma digital.
6. Protección de claves.
7. Huellas digitales.
8. Detección de virus.

# Bibliografía

Alfred J. Menezes, Paul C. Van Oorschot, Scout A. Vanstone. (1996). *Handbook of Applied Cryptography.*

Algredo-Badillo, I. (2021). *Implementación de un Módulo SHA-1 para una.* Obtenido de https://ccc.inaoep.mx/~cferegrino/Publicaciones/articulos/SHA1\_FPGA\_1Gbps\_TRC04.pdf

Arriazu, J. S. (1999). *Descripción del algoritmo DES.*

Bibing. (15 de 09 de 2021). *Capítulo 6. Algoritmos de cifrado*. Obtenido de Bibing: http://bibing.us.es/proyectos/abreproy/11314/fichero/MEMORIA\_FIRMA\_DIGITAL\_XML%252FCap%C3%ADtulo+6+Cifrado.pdf+

Caza, J. B. (04 de 2015). *Desarrollo de una aplicación para encriptar información en la transmisión de datos en un aplicativo de mensjeria web.* Obtenido de http://repositorio.puce.edu.ec/bitstream/handle/22000/8335/Disertacion\_MoyaCazaJohannaBeatriz\_EscobarErazoFranklinAndres.pdf

Chinta., S. K. (2015). *Blowfish.*

Delgado, V. (2006). *Aplicaciones prácticas de la criptografía.* Obtenido de https://www.icai.es/contenidos/publicaciones/anales/10\_16\_AplCriptog\_(II-2006)-1235.pdf

Docplayer. (15 de 09 de 2021). *LGORITMO RC5. Contreras Murillo Daniel. Flores Flores Armando. Pablo Erika Celina. Reséndiz Jiménez Omar.* Obtenido de Docplayer: https://docplayer.es/56526849-Algoritmo-rc5-contreras-murillo-daniel-flores-flores-armando-pablo-erika-celina-resendiz-jimenez-omar.html

Ecured. (2011). *Ecured*. Obtenido de ElGamal: https://www.ecured.cu/ElGamal

Kioon, M. C. (02 de 2013). *Security Analysis of MD5 Algorithm in Password Storage.* Obtenido de https://www.researchgate.net/publication/266650299\_Security\_Analysis\_of\_MD5\_Algorithm\_in\_Password\_Storage

López, A. (04 de Abril de 2021). *Todo sobre criptografía*. Obtenido de redeszone: https://www.redeszone.net/tutoriales/seguridad/criptografia-algoritmos-clave-simetrica-asimetrica/

Pousa, A. (2011). *ALGORITMO DE CIFRADO SIMÉTRICO AES.*

Proaño, C. A. (s.f.). *ALGORITMOS DE ENCRIPTACIÓN, ALGORITMOS DE ENCRIPTACIÓN,VERIFICACIÓN.*

Sanjuan, L. (2011). *Criptografía I.*

Schneier, B. (2000). *The Twofish Encryption Algorithm.*

Stallings, W. (2011). *Seguridad en redes.*

Stephane, M. (2021). *Collisions on SHA-0 in One Hour.* Obtenido de https://link.springer.com/content/pdf/10.1007%252F978-3-540-71039-4\_2.pdf

Vera Delgado, R. P. (2006). Introducción a la Criptografía: tipos de algoritmos. *Canales de mecánica y electricidad*, 46.

Vieites, Á. G. (2016). *Sistemas criptográficos Sistemas criptográficos simétricos.*