ANEXO 1

ALGORITMOS DE AUTENTICACIÓN Y CIFRADO

1. CRIPTOLOGÍA

Su nombre deriva de las palabras griegas *krypto* (oculto) y logos (estudio). Esta ciencia aplicada se encarga del estudio de los sistemas que ofrecen seguridad en medios de comunicación en los que un emisor oculta o cifra un mensaje antes de transmitirlo para que sólo un receptor autorizado pueda descifrarlo. Sus áreas principales de estudio son la criptografía, que es el arte de esconder el significado de las palabras de un mensaje cifrándolas, y el criptoanálisis, que es la ciencia de obtener la información original en ausencia de las claves rompiendo la seguridad proporcionada; aunque también se incluye la esteganografía como parte de la criptología.

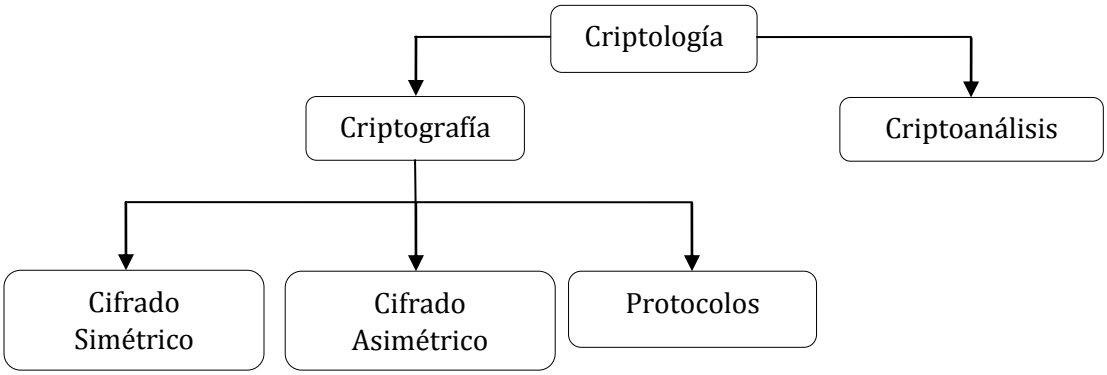

FIGURA A1. 1 CRIPTOLOGÍA

El cifrado es el proceso por el cual un mensaje en texto plano o legible se convierte en un mensaje ilegible o texto cifrado.

1.1. CIFRADO SIMÉTRICO

En los sistemas que utilizan cifrado simétrico se cuenta con una sola llave tanto para cifrar como para descifrar, esta debe ser conocida en el transmisor y en el receptor, por lo que generalmente se le llama *llave privada*. El esquema del proceso de cifrado simétrico se muestra en la figura. (Cisco , 2005)

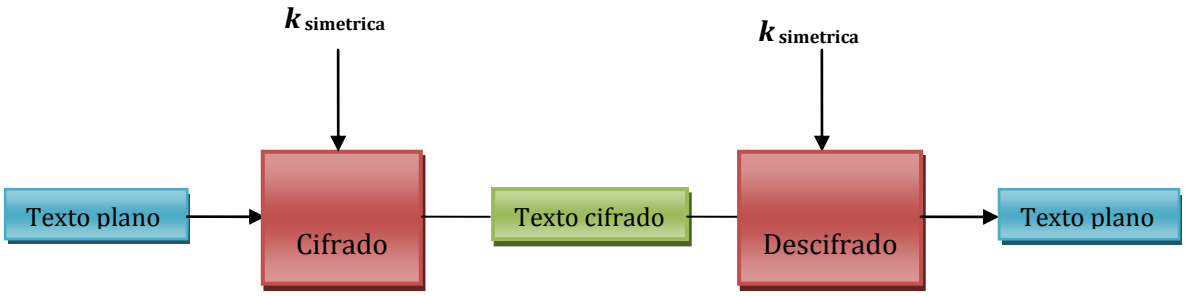

FIGURA A1. 2 CIFRADO SIMÉTRICO

Entre sus principales ventajas se pueden encontrar:

- Requiere considerablemente menos recursos computacionales.
- Presenta una longitud de clave menor que en los procesos asimétricos, con la excepción de los basados en curvas elípticas.
- Usa una clave única que sirve para cifrar y para descifrar.
- Consume menos ancho de banda.
- El cálculo de la clave no requiere que cada parte sepa quién inicio el intercambio.

Por otro lado presenta importantes desventajas como las siguientes:

- La simetría en el protocolo de administración de claves puede proporcionar vulnerabilidades.
- Al ser la clave generada en uno de los extremos, se debe de confiar en él, de lo contrario el método no sirve.
- Debido a que la clave debe estar tanto en el transmisor como en el receptor, se genera incertidumbre en como transmitir la clave si tenemos un canal inseguro.

En la criptografía simétrica se pueden encontrar dos técnicas fácilmente distinguibles dada su funcionalidad, como se aprecia en la gráfica:

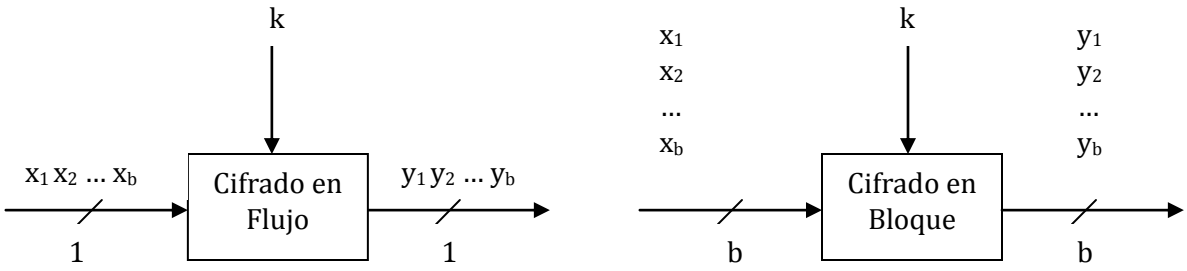

FIGURA A1. 3 TÉCNICAS DE CIFRADO SIMÉTRICO (CISCO, 2009), (CISCO , 2005)

1.1.1. CIFRADO EN BLOQUE:

Un algoritmo de cifrado por bloques transforma un bloque de texto plano de longitud n_b, a bloques en texto ilegible o cifrado de la misma longitud bajo la influencia de una clave de cifrado *k*, lo que significa que cualquier bit del bloque depende de cada uno de los otros bit en el mismo bloque y el proceso es independiente de anteriores entradas de bloques.

Mas precisamente, un bloque cifrado es un conjunto de permutaciones booleanas sobre un vector de n_b bits. Este conjunto contiene una permutación booleana para cada valor de la clave de cifrado *k*. Así, si la clave es de longitud n_k, entonces el bloque cifrado consta de 2^{nk} permutaciones booleanas.

Un algoritmo de cifrado por bloques debe satisfacer:

- Eficiencia: Dado el valor de la clave de cifrado y aplicando las correspondientes permutaciones booleanas, o su inverso, debe ser eficiente, preferiblemente sobre un amplio rango de plataformas.
- Seguridad: Debe ser imposible obtener conocimiento de la estructura interna del algoritmo de cifrado en ataques criptográficos.

Adicionalmente, antes de entrar en la descripción de los algoritmos de cifrado en bloque es importante recordar dos operaciones:

- Confusión: es una operación de cifrado donde la relación entre la clave y el texto cifrado se oscurece.
- Difusión: es una operación de cifrado donde la influencia de un símbolo del texto plano es dispersado sobre algunos símbolos del texto cifrado con el objetivo de esconder propiedades estadísticas del texto plano.

Dentro del cifrado en bloque existen tres diferentes modos de operación (Naganand & Harkins, 2003), (Cisco, 2009) DaemenRijmaen-The Design of Rijndael.pdf -- Kluwer - Fundamentals Of Cryptology.pdf]

- Modo ECB (Electronic Code Block): Este método subdivide el texto en bloques de longitud fija, por lo que el mensaje puede ser cifrado aplicando el algoritmo a cada uno de los bloques de forma independiente. Para obtener el texto original se aplica el algoritmo inverso a cada uno de los bloques.

  Entre sus ventajas se encuentra que permite codificar los bloques independientemente de su orden y que es resistente a errores, pues si uno de los bloques sufriera una alteración, el resto permanecería intacto; mientras que sus desventajas están en que si el mensaje presenta patrones repetitivos, es decir, si existen dos bloques repetidos el criptograma de ellos será igual, y la sustitución de bloques.

- Modo CBC (Cipher Block Chaining): En este modo, los bloques del mensaje son *aleatorizados* antes de aplicar el algoritmo de cifrado mediante una XOR con el bloque de texto que se desea cifrar y el último criptograma obtenido. Es decir se ha

implementado un sistema de retroalimentación, lo cual hace que la codificación del bloque actual este condicionado por la codificación de los bloques anteriores.

Presenta la desventaja de que si dos bloques son iguales, estos serán codificados de la misma manera, y peor aún, si los dos mensajes empiezan igual, estos serán codificados de la misma manera hasta llegar a la primera diferencia. Para evitar esto se recurre a un Vector de Inicialización (IV) que puede ser un bloque aleatorio, como bloque inicial de la transmisión.

- Modo CFB (Cipher Feed Back): Este modo permite codificar la información en unidades inferiores al tamaño del bloque, lo cual permite aprovechar totalmente la capacidad del canal de comunicaciones ofreciendo un buen nivel de seguridad.

- Modo de Salida Retroalimentada (OFB): Es utilizado para construir esquemas de algoritmos de cifrado en flujo.

- Modo Contador (CTR): Es otro modo en el cual se usa un algoritmo de cifrado en bloque como un algoritmo de cifrado en flujo.

- Modo Contador Galois (GCM): Es un modo en el cual se computa un Código de Autenticación de Mensaje (MAC).

1.1.2. CIFRADO EN FLUJO:

Un algoritmo de cifrado en flujo está diseñado para cifrar bit a bit, añadiendo un bit de un flujo de claves a un bit del texto, es decir se aplica el módulo 2 entre cada bit x_i del flujo de texto y un bit del flujo de clave secreta s_i donde la adición en módulo 2 es conocida como la función XOR. Hay algoritmos de cifrado de flujo sincrónico, donde el flujo de clave depende solo de la clave, y los asincrónicos donde el flujo de clave también depende del texto cifrado, como se puede apreciar en la siguiente figura:

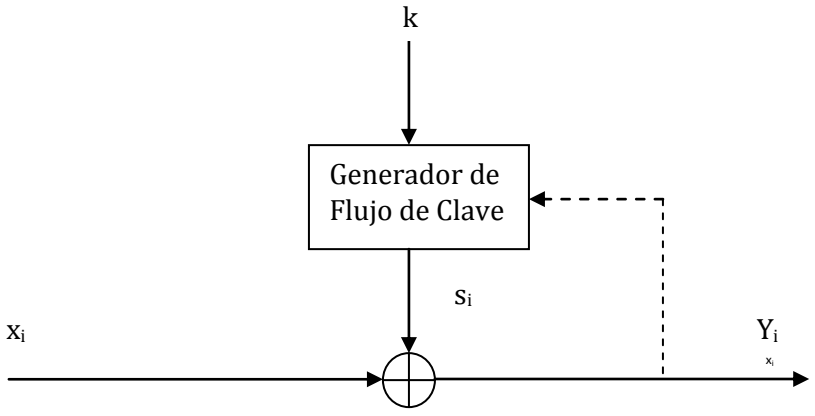

FIGURA A1. 4 CIFRADO EN FLUJO

Algunas de las principales diferencias entre las dos formas de cifrado se denotan a continuación:

- En las comunicaciones cifradas sobre Internet el cifrado en bloque es el de más amplio uso, razón por la cual en el presente proyecto se enfatiza en esta forma de cifrado.
- En terminales con recursos limitados como teléfonos celulares y dispositivos embebidos, entran a ser importantes algoritmos rápidos y de menor tamaño, por lo cual los algoritmos de cifrado en flujo son los más adecuados.
- Generalmente el cifrado en flujo es más eficiente que el cifrado por bloque, entendiéndose la eficiencia en software como menor necesidad de instrucciones al procesador para cifrar un texto plano y en hardware como menor necesidad de área de chip; sin embargo, en la actualidad la diferencia en la eficiencia entre ambos ya no es tan notoria. (Cisco, 2009) -- DaemenRijmaen-The Design of Rijndael.pdf -- Kluwer - Fundamentals Of Cryptology.pdf]

1.1.3. PRINCIPALES ALGORITMO DE CIFRADO EN BLOQUE

A) DES (DATA ENCRYPTION STANDARD)

Este algoritmo cifra bloques de 64 bits de longitud con una clave de 56 bits más 8 bits de paridad, lo cual hace que la clave sea de 64 bits, pudiendo generar 2^56 posibles claves.

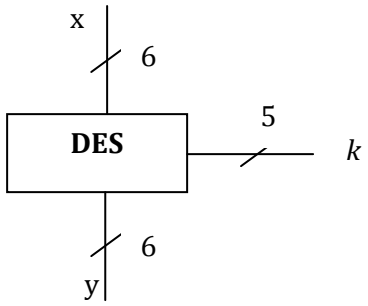

FIGURA A1. 5 ALGORITMO DE CIFRADO DES

El cuerpo del algoritmo radica en 16 iteraciones de una función ronda de llave, donde cada sub-llave utilizada es diferente y todas las sub llaves k_i son derivadas de la llave maestra *k.*

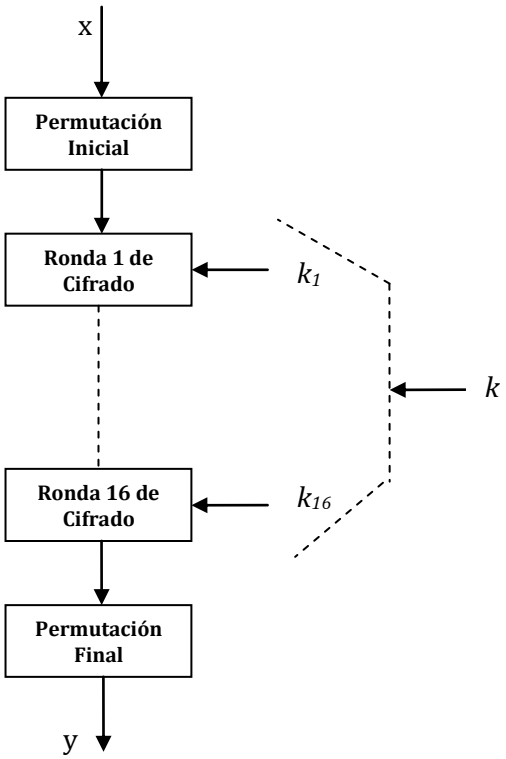

FIGURA A1. 6 ESQUEMA GENERAL DEL ALGORITMO DES

La función de ronda se puede entender mediante la siguiente gráfica, que explica los pasos se siguen en el proceso:

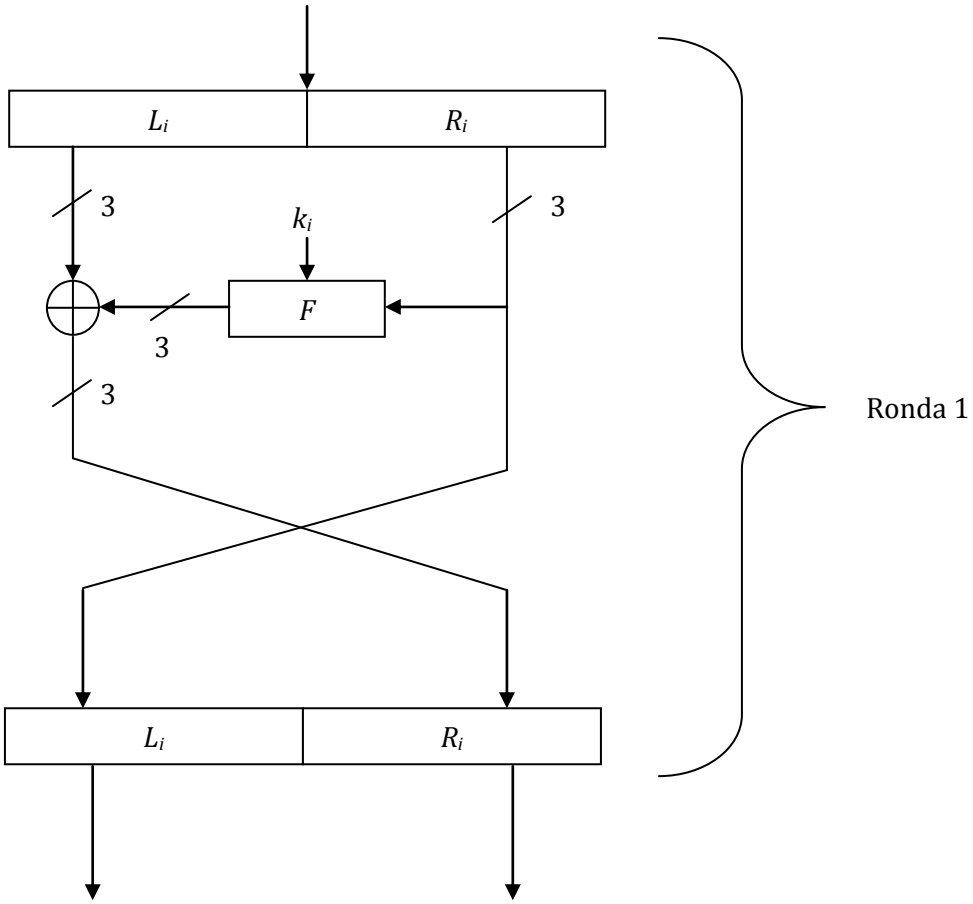

FIGURA A1. 7 PROCESO DE LA FUNCIÓN RONDA

El flujo es dividido en dos, una parte izquierda denominada L_i de 32 bits y un parte derecha R_i de 32 bits. L_i es modificado combinándolo con la salida de la función F por medio de una operación XOR; luego las partes izquierda y derecha son intercambiadas. Esta función de redondeo es también llamada *Estructura Feistel.* Un algoritmo que tiene redondeo con esta estructura es llamado algoritmo Feistel.

La función Feistel consiste de 4 pasos:

1. Expansión: Los 32 bit de entrada son expandidos a un vector de 48 bits.
2. Adición de Clave: el vector de 48 bits es modificado combinándolo con un clave de 48 bits usando la operación XOR.
3. S-Cajas (Cajas de Sustitución)[1]: EL vector resultante es mapeado dentro de un vector de 32 bits por S-cajas no lineales. El vector de 48 bits es divido entre ocho 6-tuplas que son convertidas en ocho 4 bits por ocho diferentes S-cajas no lineales, donde cada una convierte 6 bits de entrada en 4 bits de salida.
4. Permutación de Bit: Los bits del vector de 32 bits son transpuestos (Cisco Press, 2009), (Cisco, 2009), Ref: Applied Cryptography 2nd Ed – Bruce Schneier.

[1] Este tipo de cajas son utilizadas para reducir la relación existente entre el texto plano y el texto cifrado, para lo cual toma m bits de entrada y los convierte en n bits de salida.

B) TRIPLE DES

Surge como mejora de DES, dado que a mediados de 1988 se demostró que con un ataque por fuerza bruta se podía obtener la información. La debilidad radicaba en la debilidad de la clave, por lo cual con 3DES se aumenta la clave a 192 bits (64 x 3) implementando DES en fila 3 veces, aunque muchas veces la clave es de 128 bits dado que la primera y tercera clave son las mismas. A continuación se muestra de forma general su funcionamiento.

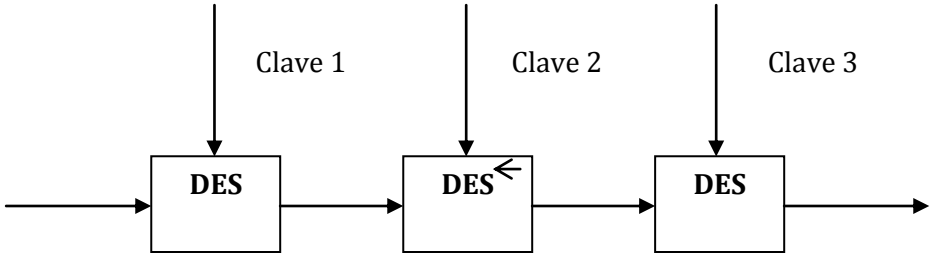

FIGURA A1. 8 ALGORITMO DE CIFRADO 3DES

C) AES (ADVANCED ENCRYPTION STANDARD)

Es el algoritmo más difundido y utilizado por los sistemas y personas en el mundo entero. Surgió como resultado de un proceso de selección por parte de Instituto Nacional de Estándares y Tecnología (NIST), el cual empezó en septiembre de 1997 y terminó el 2 de Octubre de 2000 teniendo como algoritmo ganador a Rijndael.

La única diferencia existente entre AES y Rijndael es el rango de valores soportado para la longitud del bloque y para la longitud de la clave de cifrado. Rijndael es un algoritmo con una longitud variable de bloque y de clave dentro del rango de 128 y 256 bits, mientras AES tiene una longitud fija de bloque de 128 bits, y soporta longitudes de clave únicamente de 128, 192 y 256 bits, como se muestra en la siguiente figura:

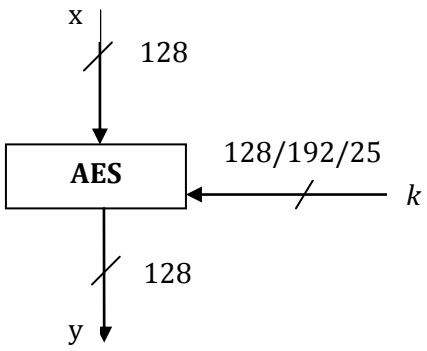

FIGURA A1. 9 ALGORITMO DE CIFRADO AES

Así, el número de interno de rondas depende de la longitud de clave utilizada, como se aprecia en la siguiente tabla:

Longitud de clave (bits)	Número de rondas n_r
128	10
192	12
256	14

TABLA A1. 1 NÚMERO DE RONDAS DE AES

Entre las principales aplicaciones de AES se puede encontrar:

- IPsec
- TLS
- Wi-Fi en el estándar IEEE 802.11i
- SSH
- VoIP (por ejemplo, Skype)

AES no hace uso de la estructura Feistel, su funcionamiento se basa en cifrar los 128 bits en una sola iteración, razón por la cual el numero de rondas es mucho menor comparado con algoritmos que hacen uso de la estructura Feistel como DES y 3DES. En su lugar, AES recurre al uso de capas; así, en cada ronda (excepto en la primera) existen tres capas donde cada una manipula todos los 128 bits de datos.

Las tres capas se describen a continuación:

1. Capa de Adición de Clave: Una ronda de clave de 128 bits, o sub claves, recordando que las sub claves son obtenidas de la clave maestra.
2. Capa de Sustitución de Byte (S-caja): Se recurre a una transformación no lineal para introducir confusión a los datos.
3. Capa de Difusión: Esta capa es la encargada de proveer difusión sobre todos los estados de los bits. Cosiste de dos subcapas:
    - *Cambio de filas:* Permuta los datos pero a nivel de bytes (octetos).
    - *Mezclado de Columnas:* Se realiza mediante una matriz que mezcla bloques de 4 bytes.

En la siguiente figura se puede observar un esquema que muestra lo anteriormente dicho:

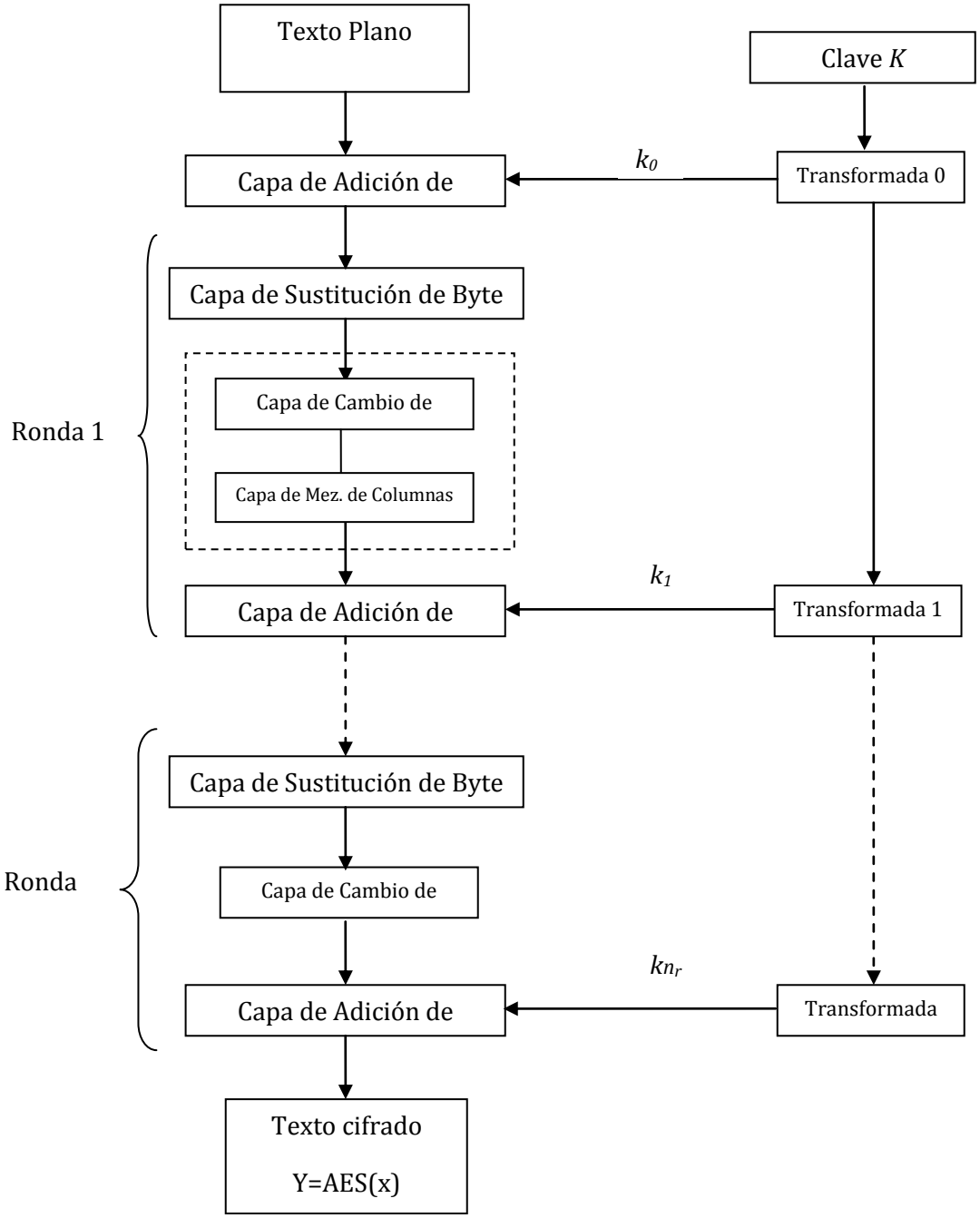

FIGURA A1. 10 FUNCIONAMIENTO DEL ALGORITMO AES (CISCO, 2009),

[Ref: Kluwer - Fundamentals Of Cryptology.pdf -- DaemenRijmaen-The Design of Rijndael.pdf]

1.2. CIFRADO ASIMÉTRICO

También conocido como cifrado de clave pública ya que se cuenta con un par de llaves, una llave pública y una llave privada. Aunque se puede cifrar con cualquier llave, por seguridad se recomienda utilizar la llave pública para cifrar y para el proceso de descifrado utilizar la llave privada; así, la llave pública es la que se reparte a los receptores con los que se desea establecer un canal seguro y aunque las llaves estén relacionadas matemáticamente no es factible derivar una a partir de la otra.

Para este tipo de cifrado se considera segura una clave si la longitud es mayor o igual a 1024 bits mientras que en los algoritmos de cifrado simétrico las claves mayores a 128 bits se consideran seguras; adicionalmente, los algoritmos asimétricos por su complejidad son sustancialmente más lentos que los algoritmos simétricos.

Dada su naturaleza de uso de dos claves, son ideales para establecer comunicaciones seguras a través de canales inseguros porque sólo es necesario transmitir la clave pública para cifrar, es por esto que en la práctica el uso se restringe al inicio de sesión.

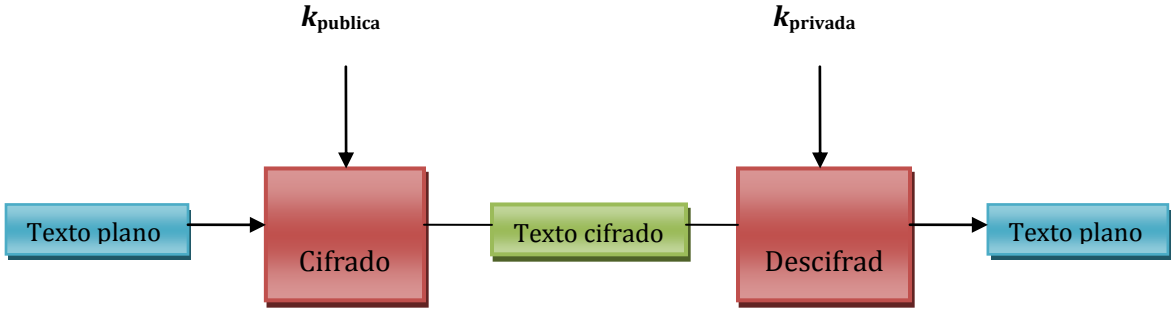

FIGURA A1. 11 CIFRADO/DESCIFRADO ASIMÉTRICO

Es de recordar que el sistema de cifrado asimétrico fue descrito en 1976 por Whitfield Diffie y Martin Hellman, aunque se cree que la Agencia de Seguridad Nacional (NSA) de los Estados Unidos ya tenía conocimiento del sistema desde 1966.

Así, entre las principales aplicaciones que se puede encontrar, están:

- Establecimiento de clave: Protocolos para el establecimiento de claves sobre canales inseguros, por ejemplo Intercambio de claves Diffie-Hellman o RSA.
- No repudio: A través de las firmas digitales se puede ofrecer no repudio e integridad del mensaje.
- Autenticación: Mediante firmas digitales y protocolos.
- Confidencialidad: se puede cifrar la información usando algoritmos como RSA o ElGamal.

De las anteriores aplicaciones, es importante profundizar en la prestación del servicio de confidencialidad dado que es una de las de mayor uso para poder establecer un canal seguro sobre un canal inseguro, y es en este entorno donde el cifrado asimétrico cobra importancia, ya que para establecer el canal seguro solo es necesario transmitir la clave para cifrar y no para

descifrar, contrario a lo que sucede en los algoritmos simétricos en donde la clave transmitida se utiliza para ambos propósitos. A continuación se describe una comunicación típica:
A desea enviar a B el mensaje m, pero el canal es inseguro, entonces A le pide a B que le envié la llave pública $K_{PUBLICA}$, B se la envía a través del canal inseguro sin ningún problema porque esta llave solo sirve para cifrar. Con la llave $K_{PUBLICA}$, A cifra el mensaje y se lo envía a B a través del canal, B lo recibe y lo descifra con su llave privada $K_{PRIVADA}$, y obtiene el mensaje m.

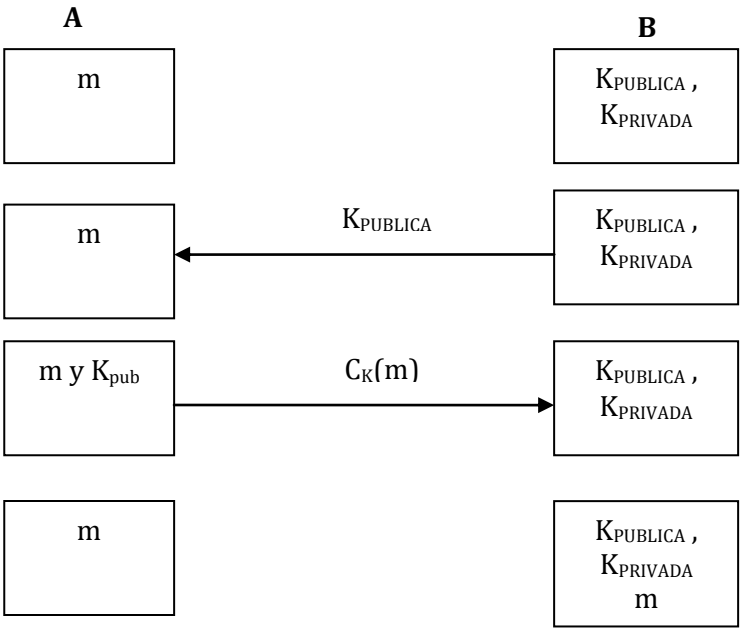

FIGURA A1. 12 PROCESO DE CIFRADO ASIMÉTRICO

Asimismo, es muy útil cuando se desea verificar la autenticidad del origen de los datos. Supóngase que A recibe un mensaje m de B, pero A no sabe si B fue quien realmente se lo envió, así que decide pedirle a B un resumen del mensaje r_M, mediante las funciones resumen o *hash* y codificado. B para codificarlo debe usar la llave privada $K_{PRIVADA}$ para así autenticarse, dado que esta llave solo la posee B. A recibe el mensaje codificado y lo descifra utilizando la llave publica $K_{PUBLICA}$, obteniendo el resumen r_M, ahora A con el mensaje m que había recibido previamente obtiene un resumen del mensaje r'_M, si los dos resúmenes son iguales entonces el mensaje m recibido es auténtico.

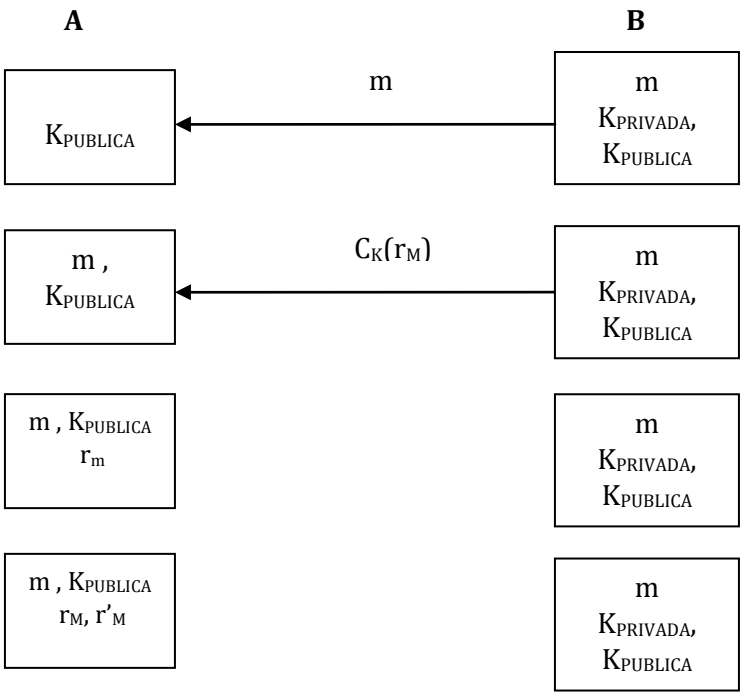

FIGURA A1. 13 PROCESO DE CIFRADO ASIMÉTRICO

[Ref: Kluwer - Fundamentals Of Cryptology.pdf --- Applied Cryptography 2nd ed. - B. Schneier (1996) WW.pdf --- CRIPTOGRAFÍA Y SEGURIDAD.pdf]

1.2.1. PRINCIPALES ALGORITMOS DE CIFRADO ASIMÉTRICO

A) INTERCAMBIO DE CLAVES DIFFIE–HELLMAN (ICDH)

Es un esquema que permite a las dos partes involucradas derivar un clave secreta común sin la necesidad de haber establecido previamente algún secreto entre ellas, todo a través de un canal inseguro. Es utilizado en protocolos muy comunes como *Secure Shell* (SSH), *Transport Layer Security* (TLS) e *Internet Protocol Security* (IPSec).

Supóngase que A y B desean usar ICDH, entonces se debe:

1. Escoger un numero primo grado *p.*
2. Escoger un entero α entre {2,3,…,*p - 2*}.
3. Publicar los valores *p* y α tanto para A como para B. A este paso se le como protocolo de inicio.
4. Luego, A escoge un numero aleatorio x entre {2,…, *p - 2*} y calcula $\alpha^x \equiv a \pmod p$ y se lo envía a B, posteriormente B hace lo mismo, calcula un numero aleatorio y entre {2,…, *p - 2*} y calcula $\alpha^y \equiv b \pmod p$ y se lo envía a A. Ahora A calcula $K_{AB} = b^x \equiv k_A \pmod p$ y B calcula $K_{AB} = a^y \equiv k_B \pmod p$ y de acuerdo a la siguiente expresión:

$$k = (\alpha^x)^y = \alpha^{yx} \bmod p$$

Se puede concluir que $K_A = K_B = K$.

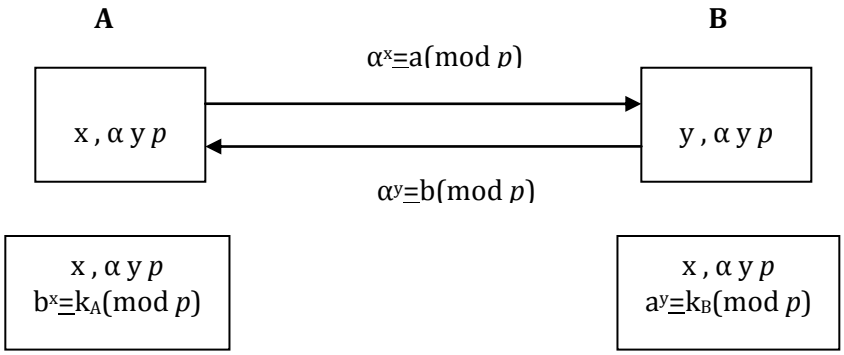

FIGURA A1. 14 INTERCAMBIO DE CLAVES DIFFIE-HELLMAN

(Cisco, 2009)[Ref: Applied Cryptography 2nd ed. B. Schneier (1996) WW.pdf --- CRIPTOGRAFÍA Y SEGURIDAD.pdf]

2. FUNCIONES HASH Ó RESUMEN

Las funciones resumen son muy importantes y útiles en el campo de la criptografía, dado que permiten obtener de un mensaje una cadena cuya longitud es más corta y fija, que permite verificar la integridad del mensaje. El resumen obtenido es una representación única del mensaje y es muy difícil de falsificar. Todas las funciones hash o resumen deben tratar de satisfacer tres pilares de la seguridad:

- Resistencia pre-imagen: No debe ser factible poder obtener el mensaje de entrada, a partir de la salida.
- Segunda resistencia pre-imagen: El resumen de dos mensajes diferentes no debe ser el mismo.
- Resistencia a colisiones y a ataques de cumpleaños.

La probabilidad de poder obtener de dos mensajes diferentes un resumen de igual valor es tan baja que se dice que computacionalmente es imposible, lo cual hace tan apetecible su uso; además de su reducido tamaño.

Las funciones hash se dividen en dos tipos:

- Funciones hash dedicadas: son todos los algoritmos específicamente diseñados para trabajar como funciones hash.
- Funciones hash basadas en Algoritmos de Bloques: son funciones hash obtenidas a partir del uso de algoritmos de bloques.

La idea básica de las funciones hash es subdividir el mensaje entrante en pequeños bloques de igual tamaño que son pasados secuencialmente a través de la función hash, la cual en su parte interna tiene una función de compresión. El proceso es iterativo originando que el resumen sea la última iteración de la función, como se muestra a continuación.

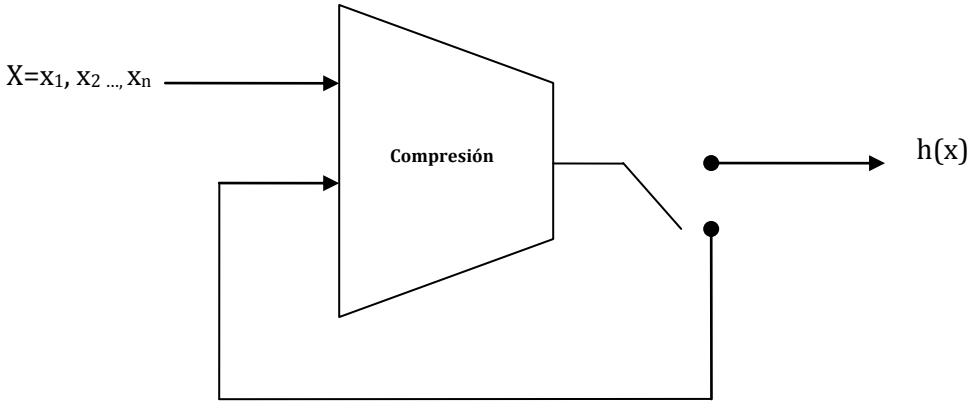

FIGURA A1. 15 FUNCIÓN RESUMEN

(Cisco Press, 2009), (Cisco, 2009) [Ref: CRIPTOGRAFÍA Y SEGURIDAD.pdf]

Los algoritmos Hash más utilizados son:

2.1. MESSAGE-DIGEST MD5

Este algoritmo pertenece a la familia MD, siendo la evolución del algoritmo MD4, con una serie de mejoras pero manteniendo la filosofía de MD4. Procesa el mensaje de entrada en bloques de 512 bits generando una salida de 128 bit y la resistencia a colisión es de 2^{64}. Los bloques de 512 bits son subdivididos en sub bloques de16 y 32 bits. Estos bloques son reorganizados en el bucle principal el cual consiste de 4 rondas y la salida de 128 bits es el resultado de concatenar 4 bloques de 32 bits.

Entre sus principales usos podemos encontrar:

- Protocolos de Seguridad para Internet.
- Sumas de verificación.
- Resumen de contraseñas las cuales han sido almacenadas.

A pesar de ser ampliamente utilizado ya presenta potenciales debilidades, siendo así que en abril de 2009 se ha presentado un ataque teórico.

2.2. SECURE HASH ALGORITHM SHA-1

Este algoritmo fue desarrollado por la NSA para ser incluido en el estándar DSS *(Digital Signature Standard)*. SHA-1 pertenece a la familia SHA y se diferencia de su antecesor en la forma de la programación de la función de compresión lo cual le permite mejorar su seguridad criptográfica. SHA-1 se recomienda como el reemplazo de MD5 debido a que presenta mejor respuesta frente a los ataques. Antes de ejecutar la función resumen se realiza un pre procesamiento, luego se procesa el mensaje de entrada en bloques de 512 bits generando una salida de 160 bits y su resistencia a colisión es de 2^{80}. La función de compresión consiste en 80

rondas las cuales son divididas en 4 estados de 20 rondas cada una. Tal es su aceptación que en 2001 el NIST publicó 3 variantes:

- SHA-224: produce un resumen es de 224 bits.
- SHA-256: produce un resumen es de 256 bits.
- SHA-384: produce un resumen de 384 bits.
- SHA-512: produce un resumen de 512 bits.

A continuación se presenta una tabla con el fin de poder comparar la familia SHA con MD5.

Algoritmos	Salida (bits)	Entrada (bits)	# de rondas	Colisiones encontradas
MD5	128	512	64	Si
SHA-1	160	512	80	No todavía
SHA-224	224	512	64	No
SHA-256	256	512	64	No
SHA-384	384	1024	80	No
SHA-512	512	1024	80	No

TABLA A1. 2 COMPARACIÓN MD5 VS SHA

[Ref: Cisco Press - Implementing Cisco IOS Network Security (IINS) (Self-Study)(2009).pdf ---- CRIPTOGRAFÍA Y SEGURIDAD.pdf --- Understanding Cryptography - A Textbook for Students and Practitioners Dec 2009.pdf]

2.3. HASHED MESSAGE AUTHENTICATION CODES (HMAC)

Los HMAC son el resultado de combinar el uso de las funciones resumen y cifrado simétrico, es decir se utiliza una clave secreta como entrada a la función hash, por lo que el resumen ya no solo depende de la entrada sino que depende además de la clave, aumentando el nivel de seguridad (Cisco Press, 2009). A continuación se muestra el funcionamiento:

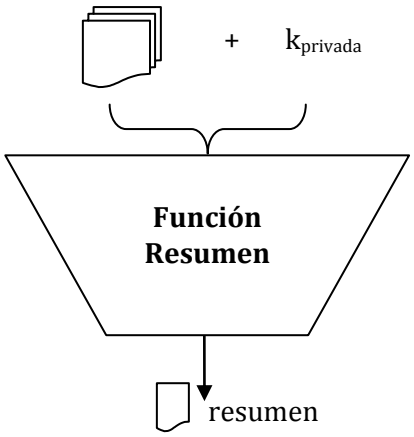

FIGURA A1. 16 ESQUEMA GENERAL DE FUNCIONAMIENTO DE LAS HMAC