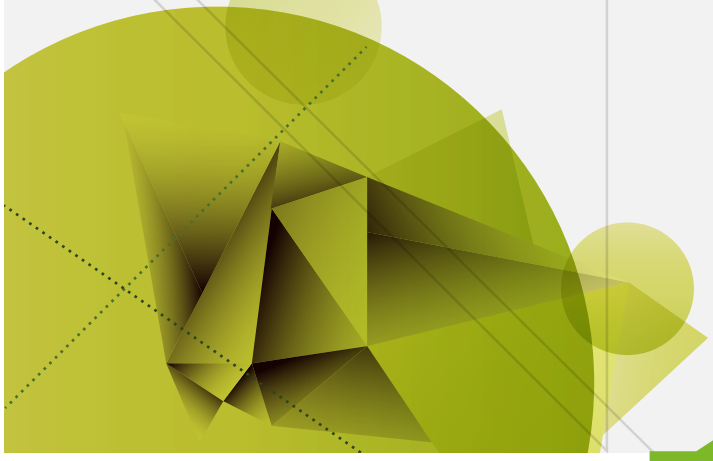
Criptografía y mecanismos de seguridad



Autor: Lucy Noemy Medina Velandia

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Criptografía y mecanismos de seguridad / Lucy Noemy Medina Velandia, / Bogotá D.C., Fundación Universitaria del Área Andina. 2017

978-958-5460-19-5

Catalogación en la fuente Fundación Universitaria del Área Andina (Bogotá).

Edición:
Fondo editorial Areandino
Fundación Universitaria del Área Andina
Calle 71 11-14, Bogotá D.C., Colombia
Tel.: (57-1) 7 42 19 64 ext. 1228
E-mail: publicaciones@areandina.edu.co
http://www.areandina.edu.co

Primera edición: noviembre de 2017

Corrección de estilo, diagramación y edición: Dirección Nacional de Operaciones virtuales
Diseño y compilación electrónica: Dirección Nacional de Investigación

Hecho en Colombia
Made in Colombia

Criptografía y mecanismos de seguridad

Autor: Lucy Noemy Medina Velandia

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

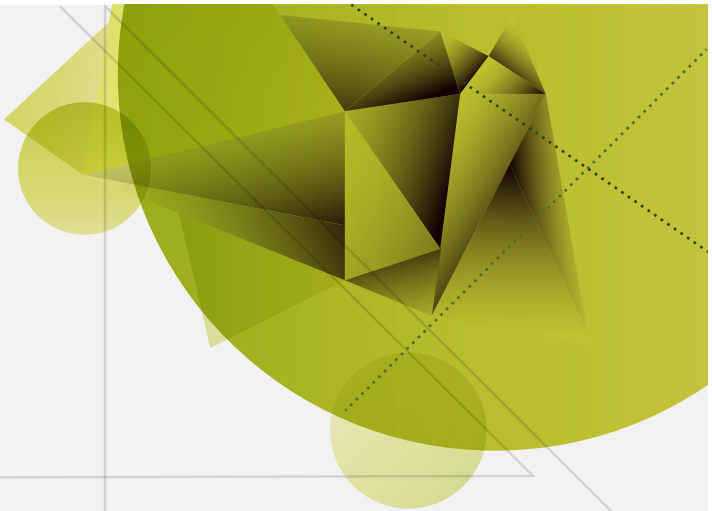
Índice

Índice

1

Unidad 1

Criptografía



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

Debido a la creciente incursión de los usuarios en la red y al gran volumen de información que se maneja, surge la evidente necesidad de ocultar cierta información que para algunas empresas o personas es relevante y de suma importancia. Internet es utilizado para compartir información y realizar transacciones de toda índole, sobre todo las entidades financieras que realizan la mayor parte de sus operaciones en línea; pero la información que viaja través de las redes puede ser interceptada y utilizada en contra del usuario o desfavoreciendo su bienestar. Por ello, asegurar la información se hace imperativo y qué mejor que a través de criptografía (que oculta mensajes), la esteganografía (que esconde la información de manera desapercibida) y el criptoanálisis (que reconstruye un mensaje cifrado) que dan la posibilidad de proteger los datos que viajan a través de la red.

La idea de las técnicas mencionadas es que la información se vuelva confiable, fiable, íntegra y auténtica, por ello, es importante aprender de dónde viene la criptografía, los métodos más utilizados y el futuro que se espera sobre esta materia.

U1 Metodología

En este primer capítulo del módulo, se estudiarán las bases para entender los siguientes capítulos y tener claras las ideas que serán utilizadas. Se trabajarán conceptos desde qué es la criptografía, su finalidad, los métodos utilizados en la criptografía, la seguridad de la información y se estudiarán conceptos sobre la técnica llamada criptosistema, que no es otra cosa que un conjunto de instrucciones para garantizar la seguridad de la información a través de técnicas criptográficas, las cuales utilizan las claves o llaves.

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

U1 Desarrollo temático

Historia de la criptografía

Para iniciar con la historia de la Criptografía, vale la pena entender lo que es. Básicamente, se trata de una herramienta de gran utilidad para tener segura la información y que ésta sea confiable, íntegra y esté disponible cuando se necesite. En síntesis, la Criptografía es una ciencia que apunta hacia la seguridad de los datos y los documentos, para ello, utiliza códigos o claves que permiten almacenar información secreta que pueda circular por medios públicos o privados y que para muchos pase desapercibida, solo la conocerá el emisor y el receptor siempre y cuando sepan cómo traducir la información para entenderla o cómo cifrarla para que otros no la comprendan.

Es importante que el educando aprenda que la Criptografía tiene dos divisiones, de una parte, la Criptografía y de otra el Criptoanálisis. La primera, estudia las técnicas del cifrado de la información y la segunda, se dedica a todo lo contrario, es decir, a descifrar o decodificar esa información.

En la figura 1, se muestra lo que sucede cuando se envía un mensaje, este se cifra y continúa su viaje como un Criptograma, o sea que hace referencia al mensaje cifrado, es decir que no es entendible por quien no conoce la clave para descifrarlo. En ese trayecto, el mensaje cifrado o Criptograma puede sufrir dos eventos, el uno que llegue al receptor y lo descifre o que el mensaje sea interceptado y los intrusos lo desencripten y puedan leer el mensaje.

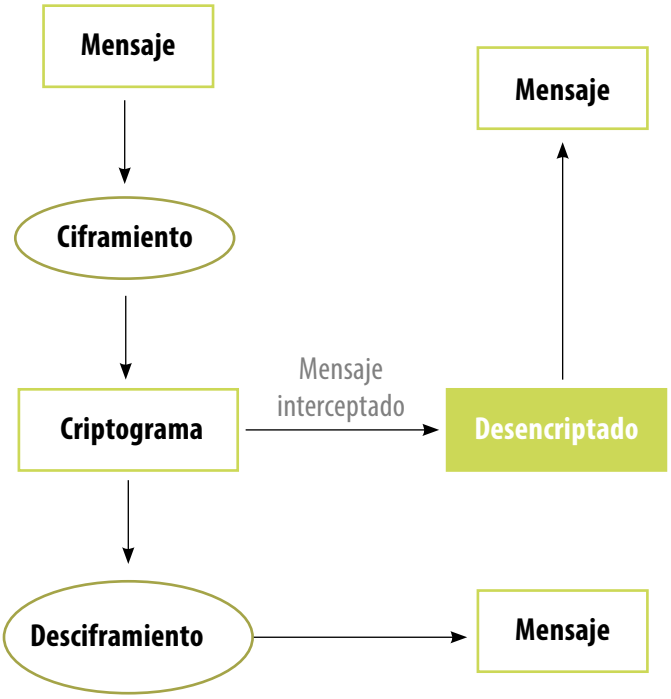


Figura 1. Ciclo de un mensaje secreto
Fuente: Propia.

Una vez entendidos los conceptos de criptografía y criptoanálisis, iniciemos con la historia de la criptografía.

La práctica del ciframiento, ha existido desde épocas remotas, es el caso de la Mesopotamia, la India y la China, utilizaban métodos

para que nadie supiera lo que ellos escondían. De la misma manera, Julio César, militar y político romano, utilizaba un método particular para cifrar mensajes, consistía en sustituir cada una de las letras de un mensaje por la tercera letra siguiente del alfabeto. Posteriormente, durante la Primera Guerra Mundial, la criptografía se convierte en un arma útil para la comunicación, pero los resultados no fueron los esperados, por ejemplo, en esta Primera Guerra Mundial se utilizó intensamente la criptografía; el código Ubchi fue desarrollado por los alemanes, el cual fue penetrado por los franceses, a la vez, los códigos de la fuerza naval de los alemanes fueron descifradas por Room 40, también conocido como 40 O.B., que era una sección del Almirantazgo británico que trabajaba con el criptoanálisis durante esta guerra mundial; esta intromisión dentro de los códigos navales alemanes, hizo que el Reino Unido tomara la delantera a Alemania y preparara batallas para combatirlos.

Criptografía clásica

Hace más de 4500 años en Egipto se inicia el uso de jeroglíficos tallados en monumentos.

Los hebreos utilizaron también los cifrados por sustitución monoalfabético.

El número "666" denominado el número de la bestia es considerado como un criptograma que oculta algo peligroso.

Los militares espartanos utilizaron el cifrado por transposición por medio de uno de los primero dispositivos de cifrado como era la Escitala.

Los romanos utilizaron el cifrado César y sus variaciones.

En la india consideran el Kama Sutra como un técnica encriptada para que los amantes se comunicaran sin ser descubiertos.

Criptografía medieval

Año 1000, se inventa la técnica del análisis de frecuencias con el objeto de romper o descifrar los cifrados por sustitución monoalfabética, todo esto se realizó a partir del Corán, porque le hicieron un análisis textual.

En 1465 León Battista Alberti inventó el cifrado polialfabético.

En 1510 se utiliza la criptografía en Japón, pero fue en 1860 que realmente se utilizaron las técnicas avanzadas de criptografía, cuando el país se abrió hacia occidente.

Criptografía de 1800 a la Segunda Guerra Mundial

En el siglo XIX se desarrollaron soluciones adecuadas para el cifrado y el criptoanálisis. Se distinguieron Charles Babbage, que estudió el cifrado polialfabético. Esta fue una época que se distinguió porque la criptografía constaba de reglas muy generales.

En el año 1840, otro personaje que se distinguió en esta época fue Edgar Alan Poe, quien desarrolló métodos sistemáticos que resolvían cifrados, Poe a través de un periódico invitada a las personas para que enviaran mensajes cifrados que él resolvía con facilidad, esto le trajo atracción por un tiempo.

Antes de la segunda guerra mundial William Friedman utilizó técnicas estadísticas para desarrollar el criptoanálisis.

Criptografía en la Segunda Guerra Mundial

Se empieza el uso de máquinas de cifrado mecánicas y electromecánicas.

Los alemanes utilizan bastante la máquina llamada Enigma del ejército alemán.

Una vez estalló la segunda guerra mundial, Allan Turing y Gordon Welchman, progresaron bastante en la tecnología de descifrado de la máquina Enigma.

En 1940 los criptógrafos británicos y holandeses rompieron los sistemas criptográficos de la Armada japonesa.

Militares del ejército americano rompió el sistema criptográfico de la diplomacia japonesa, que llevó a la victoria de Estados Unidos en la batalla de Midway.

Los alemanes en cabeza de Max Newman y su grupo de trabajo crearon el Colossus, que colaboró con el criptoanálisis.

Hubo tres máquinas utilizadas para la criptografía, como la Typex británica y la Sigaba norteamericana, Lacida de los polacos, la M-209 y la M-94.

Criptografía moderna

En 1949, Shannon Claude, llamado el padre de la criptografía moderna, realiza escritos y un libro sobre la teoría de la información y la comunicación, que sirvió como base para el criptoanálisis y criptografía moderna.

La criptografía es orientada al espionaje y contraespionaje, la dependencia de los Estados Unidos denominada la NSA, fue la abanderada de la criptografía entre los años 1950 a 1970.

A mediados de los años 70 se avanzó en dos aspectos, uno fue la propuesta que la IBM envió al Registro Federal una propuesta titulada "Data Encryption Standard - DES-", con el fin de desarrollar sistemas de comunicación seguras para empresas financieras-bancarias- y otras empresas.

En 1977, luego de realizar mejoras al escrito antes mencionado, la NSA de los Estados Unidos, publica el "Federal Information Processing Standard".

En 2001 el DES fue reemplazado por el AES -Advanced Encryption Standard- que es un algoritmo llamado Rijndael realizado por criptógrafos Bbelgas. De este algoritmo se han realizado variantes que hoy en día todavía se utilizan. Desafrotunadamente el DES (que utiliza 56 bits) es hoy en día inseguro para los nuevos criptosistemas.

En 1976 cambian radicalmente la forma de funcionamiento de los criptosistemas. Aparece el famoso "intercambio de claves Diffie-Hellman" desarrollado por Whitfield Diffie y Martin Hellman.

También en 1976 a partir de la creación del algoritmo Diffie-Hellman, aparece un nuevo algoritmo para realizar cifrado, llamados algoritmos de cifrado asimétrico.

También aparece en contraste a los algoritmos de cifrado simétrico, los algoritmos de clave asimétrica, que utilizan claves relacionadas matemáticamente, consiste en que una clave realiza el descifrado de la otra clave. A este par de claves se les llama hoy en día la clave privada y la clave pública. La clave privada, siempre será secreta, la clave pública la pueden conocer los usuarios en general. Para poder acceder a un mensaje cifrado de este tipo, el usuario debe conocer la clave privada y la pública.

Entre 1982 y 1992, Estados Unidos realiza grandes cambios legislativos a través de la NSA.

En 1991 Phil Zimmermann publicó el Pretty Good Privacy - PGP - que hacía una defensa del cifrado fuerte para uso público y fue distribuido como freeware, al pasar el tiempo se convirtió en el estándar RFC2440 u OpenPGP.

Criptoanálisis moderno

AES es considerado el cifrador irrompible.

Surgen los cifrados A5/1 y A5/2, dirigidos a los teléfonos móviles GSM.

Conceptos generales

La Criptografía se puede clasificar como se aprecia en la siguiente figura.

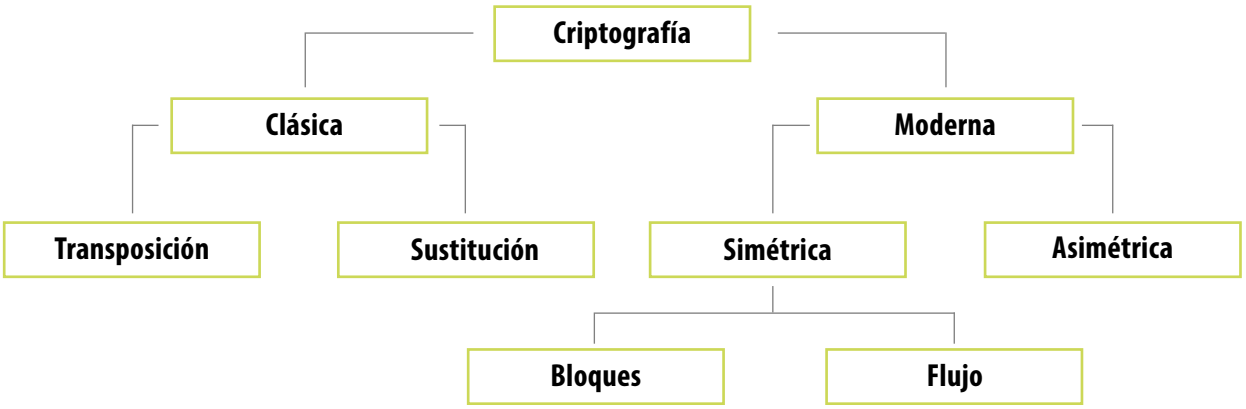


Figura 2. Clasificación de la Criptografía
Fuente: Propia.

Algunos términos comunes de la Criptografía

a. Criptografía: arte de cifrar y descifrar información, para ello utiliza técnicas que hacen posible el intercambio de mensajes de manera segura y éstos solo pueden ser leídos por la persona a quien se dirige el mensaje, sin importar si el mensaje está en un lugar inseguro o público. Son mil veces más rápidos que los algoritmos asimétricos.

b. Cifrado: proceso que transforma la información por medio de un algoritmo de cifrado y para poderlo descifrar se requiere de la clave secreta del algoritmo.

En la Figura 3, se aprecia gráficamente este proceso.

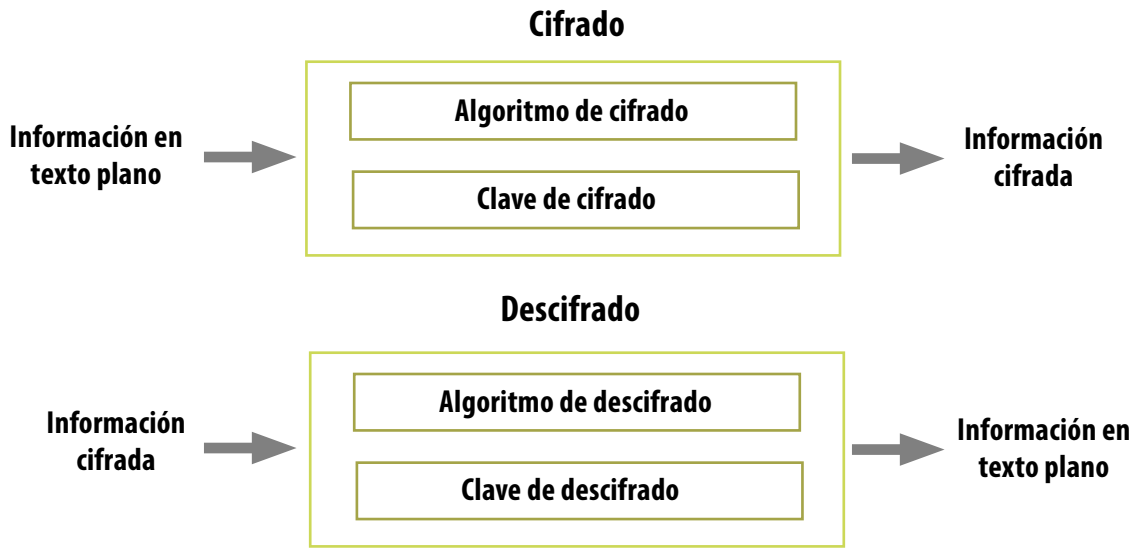


Figura 3. Cifrado y Descifrado de información
Fuente: Propia.

c. Algoritmo o criptografía simétrica: encriptan y desencriptan información con la misma clave o llave, estos algoritmos son seguros y se transmiten con velocidad. Vea la siguiente figura.

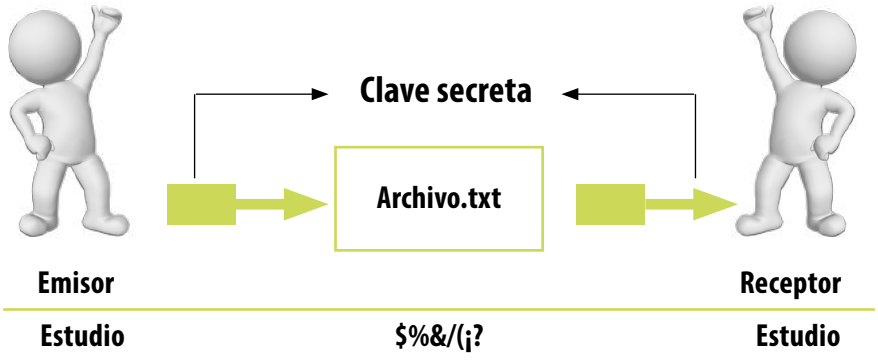


Figura 4. Cifrado y Descifrado de información con clave Simétrica
Fuente: Propia

d. Algoritmos o criptografía asimétrica: encriptan y desencriptan información con diferentes clave o llave. Los datos son encriptados con una llave pública y se desencriptarán con una clave privada. Los bits que utiliza la llave de 3000 bits para hacerla segura, son muy lentos. (Ver Figura 5)

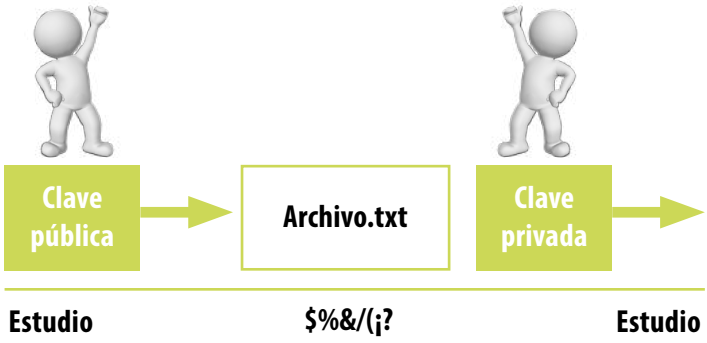


Figura 5. Cifrado con clave Asimétrica
Fuente: Propia

e. Criptosistema: procedimiento que a través de un algoritmo con una clave, transforma un mensaje en algo incomprensible para los usuarios, a menos que tengan la clave para desencriptar.

f. Esteganografía: procedimiento por medio del cual se esconde la información para que sea desapercibida, no necesariamente la información estará encriptada, puede estar a la vista de los usuarios y no percatarse de que existe un mensaje frente a sus ojos. Por ejemplo, en un paisaje que muestre el cielo, puede existir una nube (dentro de la cual estará el mensaje).

g. Criptoanálisis: es lo opuesto a encriptar, es decir es el estudio de métodos para obtener el sentido de una información antes cifrada (descifra mensajes), utilizan la clave secreta para realizar la desencripción.

h. Métodos de criptográficos: son métodos para encriptar información. Entre estos métodos se encuentran: la criptografía simétrica, la asimétrica, la híbrida, la mezcla de las anteriores y las funciones hash o de resumen

i. Métodos de seguridad: métodos de clave simétrica, métodos de clave pública, clave pública SSL, algoritmos de encripción hashing, métodos de autenticación (que pueden ser por contraseñas, tarjetas de acceso o firma digital)

Finalidad de la criptografía

El principio fundamental de la criptografía es mantener la comunicación entre dos entes con toda la seguridad y privacidad. Para ello, se modifica o altera el archivo original haciéndolo completamente incomprensible para otros individuos, excepto para el receptor de la información. El envío de información entre dos personas debe hacerse por medio de reglas o claves para poder enviar o recibir archivos confidenciales, sin temor a que caiga en manos extrañas y entiendan el mensaje.

Criptografía y criptosistema

Los criptosistemas, son un conjunto de procedimientos que salvaguardan la seguridad de la información entre dos entes y utilizan diferentes técnicas de criptografía. Existe un término fundamental en estos procedimientos, como es el uso de una llave o clave.

Los criptosistemas tienen la particularidad de tomar información legible y convertirla en información no legible o no entendible.

Los criptosistemas están formados matemáticamente por:

a. Un alfabeto, el cual facilita la construcción de un mensaje.
b. Espacios o claves, que pueden es conjunto finito de ellas y pueden ser para encriptar o desencriptar información.
c. Transformaciones de cifrado, diferentes aplicaciones del alfabeto, llamadas también transformaciones aritmético-lógicas de cifrado.
d. Transformaciones de descifrado, diferentes aplicaciones del alfabeto, llamadas también transformaciones aritmético-lógicas de descifrado.

La criptología es la ciencia compuesta por la criptografía y el criptoanálisis, recordando, la criptografía encripta la información y el criptoanálisis la desencripta. La criptografía utiliza técnicas y algoritmos para poder intercambiar información de forma segura.

Algunas características de la criptogrfía es la confidencialidad, que consiste en garantizar la seguridad de la información y que solo los interesados y autorizados puedan tener acceso a ella. Otra característica es la integridad, la cual garantiza que la información original no se modifique, sin importar si éste es público o privado. La autenticación, garantiza la identidad del autor.

Los tipos de criptografía pueden ser:

a. Criptografía de clave secreta: son algoritmos escritos específicamente para cifrar un mensaje utilizando una clave única y secreta que es conocida solo por el receptor y el emisor, es decir, que para descifrar el mensaje tanto receptor como emisor deben conocer perfectamente dicha clave.
b. Criptografía de clave pública: son algoritmos escritos específicamente para cifrar y descifrar los mensajes a través de claves o llaves distintas, una para el cifrado y otra para el descifrado. Una clave no permite conseguir la otra, pues están construidas por medio de relaciones matemáticas. Para poder enviar un mensaje, el usuario debe conocer la clave pública del usuario destino y cifrar el mensaje usando esa clave. Este mensaje será recuperado por el usuario receptor por medio de la clave privada.

   Este tipo de criptografía o algoritmos criptográficos trabajan con cualquiera de las claves, es decir, si un mensaje se ha cifrado con la clave pública, solo se descifra con la clave privada y el mensaje cifrado con la clave privada solo se descifrará con la clave pública.

Los usos que se le dan a la criptografía a diario pueden ser entre otros:

a. Para firmas digitales
b. Para certificados digitales
c. Para sistemas de autenticación
d. Para el correo electrónico seguro

Esteganografía

La esteganografía proviene de los griegos y significa "steganos"=secreto y "grafía"=escrito, lo que traduce "escritura secreta", la cual se define como el arte de ocultar un mensaje dentro de otros. Este tipo de ocultamiento de información se utiliza por ejemplo incrustando un mensaje dentro de un archivo digital utilizado como medio. Algunos de los archivos utilizados como: imágenes, videos, archivos de música.

Algunas herramientas de tipo free o gratuitas utilizadas para realizar esteganografía son: F5, desarrollada por Andreas Westfield. La herramienta MP3Stego, que esconde datos en archivos MP3, consiste en unir un archivo .wav con uno de texto y transformarlo en un archivo MP3. Otra herramienta es la Steganos, que encripta y esconde archivos, el usuario puede escoger un archivo o una carpeta para ser escondida y después el lugar donde desea integrarlo.

Criptoanálisis

Hace referencia a que por medio del Criptoanálisis se encuentran debilidades en los sistemas y descifrar o romper la seguridad. Una de las técnicas que utiliza el criptoanálisis es la de busca errores en el sistema y poderlo penetrar para hacer daños.

Los ataques criptoanalíticos tienen cierta clasificación:

1. Según la actitud del atacante y se subdividen en:
   - Ataques pasivos: en este tipo de ataque, el intruso no altera la comunicación, solo escucha o hace monitoreo para obtener información, utiliza la técnica de escucha de paquetes y de análisis de tráfico. No se detectan con facilidad, pues implican alteración de los datos.
   - Ataques activos: en este tipo de ataque se puede modificar el flujo de datos o se crean flujos falsos. Algunas técnicas de este tipo de ataque son: La suplantación, la modificación de mensajes (que consisten en capturar los paquetes para borrarlos, manipularlos, modificarlos o reordenarlos), la re actuación (que consiste en capturar un paquete o una retransmisión), la degradación (que consiste en degradar un servicio).
2. Según el conocimiento previo:
   - Ataque con sólo texto cifrado disponible: el atacante solo tiene acceso a unos textos cifrados o codificados
   - Ataque con texto plano conocido: el atacante tiene un conjunto de textos cifrados que conoce, pero también conoce el descifrado
   - Ataque con texto cifrado escogido: el atacante obtiene los textos cifrados planos que corresponden a algunos textos planos cifrados.
   - Ataque adaptativo de texto plano escogido: el atacante elige los textos planos teniendo como premisa la información que ha obtenido de descifrados anteriores.
   - Ataque de clave relacionada: similar al ataque con texto cifrado elegido, pero el que ataca puede elegir textos cifrados usando dos claves distintas.
3. Según el objetivo en criptoanálisis:
   - Ruptura total: el intruso deduce la clave secreta.

- Deducción global: el intruso descubre el algoritmo para cifrado y descifrado de mensajes, pero no tiene la clave.
- Deducción local: el intruso obtiene textos planos o cifrados adicionales a los conocidos.
- Deducción de información: el intruso descubre información que no era conocida anteriormente.

4. 4. Según el costo:
   - Tiempo: por el número de operaciones que pueden ser realizadas.
   - Memoria: por la cantidad de almacenamiento que se requiere para hacer el ataque.
   - Datos: por la cantidad de textos planos y cifrados.

Métodos criptográficos y seguridad

Los métodos criptográficos hacen referencia a las maneras como se cifra o descifra un mensaje.

Algunos de estos métodos de ciframientos son:

a. La criptografía simétrica: este tipo de criptografía es del tipo monoclave, es decir, una sola clave para cifrar y descifrar el mensaje, lo que hace vulnerable este tipo de ciframiento, pues si alguien escucha el canal por el cual se transmite el mensaje, se corre el riesgo de que intercepte también la clave, lo que hace inútil el ciframiento.

   Para tratar con este tipo de ciframiento simétrico, deberá utilizarse una clave muy segura y que el método de cifrado sea lo más adecuado posible, pues con los algoritmos de desciframiento que existen en la actualidad, es muy rápido que se puede contar con la clave y por supuesto con el mensaje.

   Dentro de este tipo de criptografía existen dos métodos importantes:

   - Data Encryption Standard -DES-: Este algoritmo para encriptar fue creado por la NSA cuyo objetivo era prestar servicios de protección de datos del gobierno de los Estados Unidos. En 1976, este algoritmo de encripción fue escogido por la Federal Information Processing Standar - FIPS- para ser promovido por el mundo entero. Pero al día de hoy, DES es muy inseguro, por cuanto su clave es de 56 bits, muy insuficiente para el poder de los algoritmos de desencripción y al poder computacional con que se cuenta.

   Algunas propiedades de este algoritmo:

   Cifrado por bloques, cada bloque con 64 bits, la clave tiene 64 bits y 8 de estos bits son empleados para comprobar la paridad, es decir que la longitud real de la clave es de 56 bits. DES trabaja así: toma un bloque de una longitud fija y lo transforma por medio de una serie de operaciones en otro bloque cifrado de la misma longitud.

   - Advanced Encryption Standard -AES-: Rijndael, es su otro nombre y en 1997 fue escogido como ganador en el concurso realizado por el Instituto Nacional de Normas y Tecnologías -NIST- como el mejor algoritmo de cifrado. Pasó a ser en 2001 como parte de FIPS y se transformó como estándar en 2002, a partir de 2006, es el algoritmo que más se utiliza n criptografía simétrica en el mundo.

El funcionamiento de este algoritmo es el siguiente: trabaja con una matriz de 4X4 bytes, que por medio de un algoritmo reordena los bytes de la matriz. La misma clave se utiliza para descifrar y cifrar, es decir, es un algoritmo simétrico. Este algoritmo funciona por medio de una serie de ciclos de repetición, de los cuales se utilizan diez ciclos para claves de 128 bits, 12 para claves de 192 bits y 14 para claves de 256 bits.

b. One-time PAD: este algoritmo de cifrado fue inventado en 1997, el algoritmo escoge un texto a cifrar y éste se combina con una clave aleatoria que tiene la misma longitud y solo se usa una vez, lo que lo hace irrompible.

c. Cifradores de flujo y de bloque: estos cifradores trabajan con claves simétricas, toman grupos o bloques de bits de X longitud fija.

   Los cifradores de bloque, utilizan un bloque de texto plano y como resultado obtiene un bloque de texto de igual longitud, pero cifrado; esa transformación de texto plano a cifrado, se hace por medio del uso de una clave secreta. El proceso contrario, de texto cifrado a plano, se hace de la misma manera.

   El inconveniente que presentan los cifrados por bloques es que hay determinadas aplicaciones que no pueden utilizar este tipo de ciframiento por estar formadas por un flujo constante de bits.

   Los cifradores de flujo, son algoritmos que convierten un texto en cifrado, pero lo hacen bit a bit. Trabajan muy parecido a los cifradores de bloque, pero en la entrada se tiene un flujo de datos, se genera una Flujo de Clave y como salida se produce un XOR bit a bit del flujo de datos y el de clave.

   - Algoritmo de Flujo -Wired Equivalent Privacy -WEP-: es un algoritmo de flujo. Este tipo de cifrado se incluye dentro del estándar IEEE 802.11. Se basa en el mecanismo de cifrado RC4. Tiene dos tipos o variantes, la una utiliza clave de 64 bits, de los cuales utiliza 24 bits para el vector de iniciación y los otros 40 para mensaje. La otra variante, utiliza clave de 128 bits, de los cuales 24 bits son para el vector de iniciación y 104 bits para el texto. Tiene dos estándares denominados WEP-40 y WEP-104. Estas versiones de algoritmo fueron declaradas inseguras en 2004, pero aún hoy se utiliza.
   - Algoritmo Electronic Codebook -ECB-: es un algoritmo de bloque. Aunque existe muchos algoritmos de esta familia de bloque, éste es uno de los más simples. La información la parte en bloques y cada uno es cifrado por separado, pero tienen una clave común. El principal problema de este algoritmo es que como los bloques son idénticos, los bloques cifrados, también lo son, lo que lleva a que se pueda conocer el patrón y de esta manera se facilitará obtener el mensaje original. Mejorando este algoritmo, se crea el Cipher Block Chaining -CBC-, cuya mejora es la de hacer XOR al bloque antes de cifrarse con el anterior cifrado y luego cifrarlo. También surgieron algoritmos como el CFB, el OFB que hacen cifrado pase a operar como un cifrador de flujo.

d. Cifrado asimétrico: se conoce también como de clave pública. Emplea dos claves para el mismo individuo, de las cua-

les, una es privada y la otra es pública, que cualquiera puede tenerla. Este algoritmo funciona así: el emisor utiliza la clave pública del receptor y solamente con la clave privada se puede descifrar el mensaje, así solo el receptor puede acceder a la información. Pero si el emisor utiliza la clave privada para cifrar el mensaje, este solo se puede descifrar con la llave pública.

Este tipo de algoritmos asimétricos son más seguros que los simétricos. Algunas desventajas de los algoritmos asimétricos, es que toman mayor tiempo de proceso que los simétricos, las claves son más grandes y el mensaje cifrado es mucho más grande que el original.

Algunos algoritmos asimétricos:

- Rivest, Shamir y Adlema –RSA-: este algoritmo asimétrico fue elaborado en 1977 por Rivest, Sharmir y Adleman, de ahí su nombre RSA. El funcionamiento del algoritmo es el siguiente, escoge dos números primos grandes que se escogen aleatoriamente y se mantienen en secreto. La factorización de esos números primos grandes, es la principal ventaja de este algoritmo en cuanto a seguridad. Es uno de los algoritmos más seguros hasta la fecha.
- Diffie-Hellman: este protocolo de cifrado asimétrico, recibe su nombre de los creadores Whitfield Diffie y Martin Hellman. Fue el primer algoritmo de llave pública. Se denomina que es un algoritmo que intercambia claves entre partes. Empleado para acordar claves simétricas, utilizadas para cifrar una sesión. La dificultad de este algoritmo se basa en la gran dificultad para calcular logaritmos discretos en un cuerpo finito.
- Funciones Hash criptográficas: las funciones hash son algoritmos matemáticos que con una entrada X, genera una salida Y. El algoritmo matemático deberá cumplir una serie de propiedades, como:
  - Cuando se tenga una función hash específica, sin importar la longitud de la entrada X, la salida Y debe ser de un tamaño fijo.
  - Para cada X, Y tiene que ser única.
  - La función hash deberá ser rápido y sencillo de calcular
  - No se devolverá de Y a X.
  - No se presentarán colisiones. es decir que si se tienen dos X distintas, no se darán un mismo Y.
- MD5: en 1995 fue creado este algoritmo por Ronald Rivest. Es un algoritmo de hash muy utilizado actualmente. Se basa en dos algoritmos, el MD2 y el MD4. Es llamado de reducción criptográfico de 128 bits. Usado para comprobar que a algún software que se instale en el PC bajado de internet no haya sido alterado por algún usuario que le haya agregado un virus o troyano al instalador; el MD5 indicará con seguridad si ese software que se descargue de la red es el oficial o ha sufrido cualquier cambio y podría ser dañino para el sistema.

Criptografía de uso cotidiano

Por medio de la criptografía la información sensible de las personas puede ampararse, es el caso de: Los datos personales, Datos Bancarios

La información puede protegerse en la creación, modificación, almacenamiento, procesamiento y transmisión.

Para preservar la identidad de las personas son indispensables los algoritmos criptográficos, de allí que deberán estar presentes en nuestro diario vivir como en:

- El comercio electrónico: como en las compras en línea, en los dispositivos móviles, pagos con algún tipo de tarjeta débito o crédito, pago de impuestos.
- Cifrar el almacenamiento de datos: bases de datos, dispositivos de almacenamiento, almacenamiento distribuido.
- Cifrar las comunicaciones: correo electrónico, redes inalámbricas, redes sociales, mensajería instantánea, intercambio de documentos, radio, televisión.
- Cifrar las investigaciones.
- Cifrar la información en el sector financiero.
- Cifrar diversos sectores: voto electrónico, pasaporte, planes estratégicos, sector financiero, cajeros automáticos, investigaciones.

Algunos ejemplos con ejercicios sobre distintos ciframientos

Ejemplo de cifrado de mensaje: Ana envía un mensaje a David

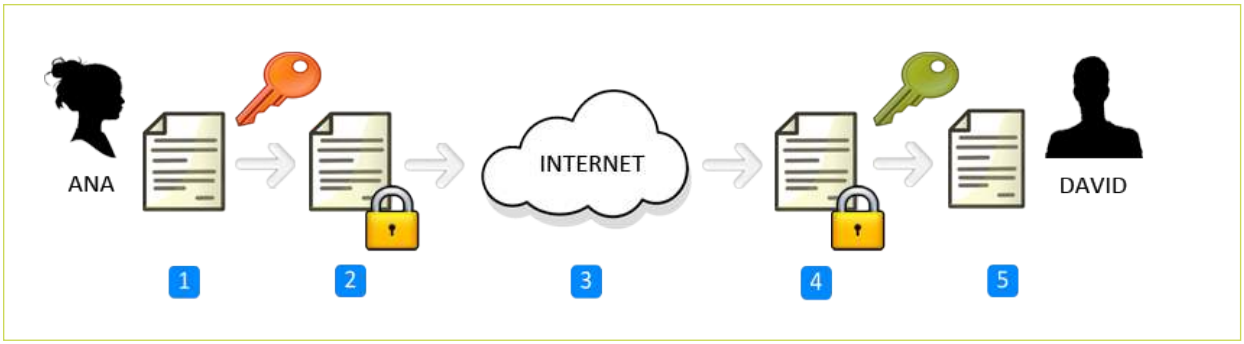


Imagen 1. Cifrado de un mensaje con clave pública.
Fuente: https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica#/media/File:CriptografiaAsimetrica.png

1. Ana redacta un mensaje.
2. Ana cifra el mensaje con la clave pública de David.
3. Ana envía el mensaje cifrado a David a través de internet, ya sea por correo electrónico, mensajería instantánea o cualquier otro medio.
4. David recibe el mensaje cifrado y lo descifra con su clave privada.
5. David ya puede leer el mensaje original que le mandó Ana.

Ejemplo de firma digital con clave asimétrica: David envía un mensaje a Ana

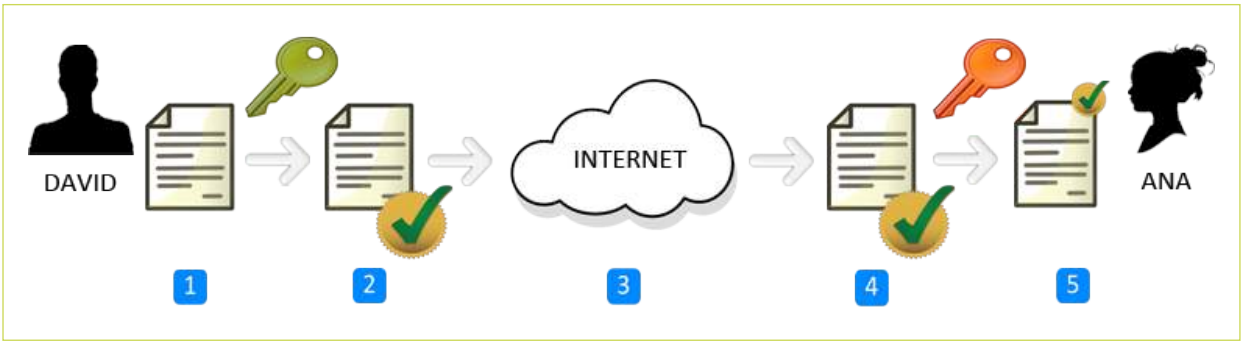


Imagen 2. Cifrado de un mensaje con clave privada
Fuente: https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica#/media/File:Firma_Digital_Asim%C3%A9trica.png

1. David redacta un mensaje.
2. David firma digitalmente el mensaje con su clave privada.
3. David envía el mensaje firmado digitalmente a Ana a través de internet, ya sea por correo electrónico, mensajería instantánea o cualquier otro medio.
4. Ana recibe el mensaje firmado digitalmente y comprueba su autenticidad usando la clave pública de David.
5. Ana ya puede leer el mensaje con total seguridad de que ha sido David el remitente.

Ejemplo de una clave simétrica

Imagen 3. Cifrado con clave Simétrica.
Fuente: http://www.rinconastur.com/php/php19.php

Ejemplo del cifrado César

Se tiene el texto Original: ABCDEFGHIJKLMNÑOPQRSTUVWXYZ, A este texto se le puede aplicar un desplazamiento de posiciones hacia la derecha de siete espacios hacia la derecha, entonces el texto cifrado será:

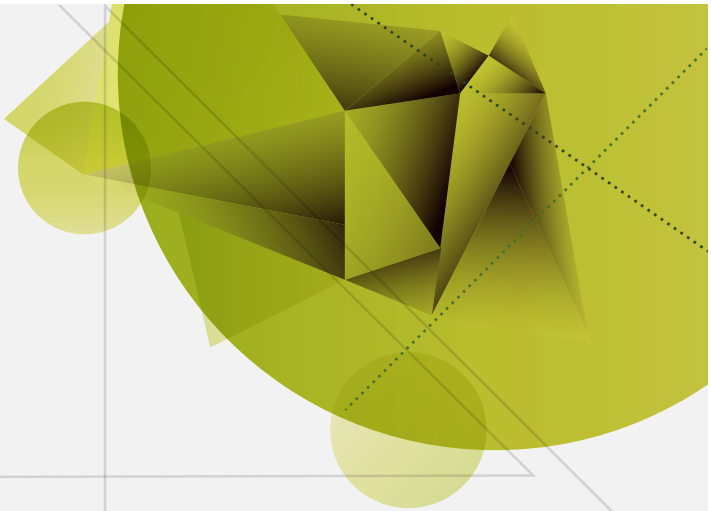
Texto cifrado: HIJKLMNÑOPQRSTUVWXYZABCDEFG.

Conociendo el texto original y su ciframiento, haga lo mismo con su nombre, de tal manera que su nombre quede cifrado.

1

Unidad 1

Usos de la criptografía



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

Desde tiempos remotos, los seres humanos han tratado de protegerse en todos los sentidos de los demás, en especial, el conocimiento, pues este lo utilizaban para crear mecanismos de defensa para salir victoriosos en las guerras. Surge a partir de allí la transformación del conocimiento en claves secretas representándolo a través de criptogramas. De ahí la importancia de garantizar la protección de la información por medio de protocolos de ciframiento que limitan el acceso de la información.

De otra parte, es importante que los estudiantes entiendan cómo establecer o confirmar la identificación de alguien a través de una red o de un sistema, para poder confirmar la procedencia de quien se identifica y poder verificar su identidad.

Es significativo que se entienda la importancia de mantener los datos libres de modificaciones no autorizadas y cómo controlar esta situación, pero también tener la información disponible para cuando se requiera, más importante aún, es la irrefutabilidad, que consiste en que cuando un usuario modifique o utilice la información éste no pueda negar la acción realizada.

En esta semana también se revisarán los protocolos de seguridad o criptográficos que permiten describir cómo debe utilizarse cada algoritmo para establecer claves, autenticar entidades, realizar cifrados, transportar datos de forma segura, entre otros.

U1 Metodología

Una vez estudiada la unidad 1, en donde se aprendieron los conceptos básicos sobre criptografía, en esta unidad se revisarán otros términos importantes para sustentar y tener como base para continuar con los siguientes conocimientos. Es importante que si no le ha quedado claro algún concepto de la unidad1, se contacte con su tutor para pedir explicación, o de lo contrario, acudir a la cartilla de la unidad 1 y volver a repasar los conceptos.

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

Usos de la criptografía

La confidencialidad

La información tiene una propiedad que la hace única y es la de garantizar que ésta sea accesible solamente por las personas interesadas para utilizar la información, pero en términos simples:

Confidencialidad de la información
=
Protección de datos entre un emisor y un receptor

Para garantizar la confidencialidad de la información, se utilizan los mecanismos de cifrados o protocolos con el fin de que la información permanezca segura, solo hasta que a quien se le envía el mensaje lo descifre.

Imagen 1. La confidencialidad
Fuente: http://definicion.de/confidencialidad/

La confidencialidad de los datos es justificada porque se requiere en el ejercicio profesional, por ejemplo, de los médicos, de los abogados, de los sacerdotes, etc.

Un concepto importante que debe quedar claro es el significado que tiene la seguridad de la información y la seguridad informática. Seguridad informática, hace referencia a la seguridad de los medios como el hardware, el software y por supuesto los datos. La seguridad de la información prevé y mantiene mecanismos para asegurar la integridad de la información.

Son tres características importantes que debe tener la información para que sea segura, estas son: la disponibilidad, la confidencialidad y la integridad, llamado el triángulo de la seguridad de la información.

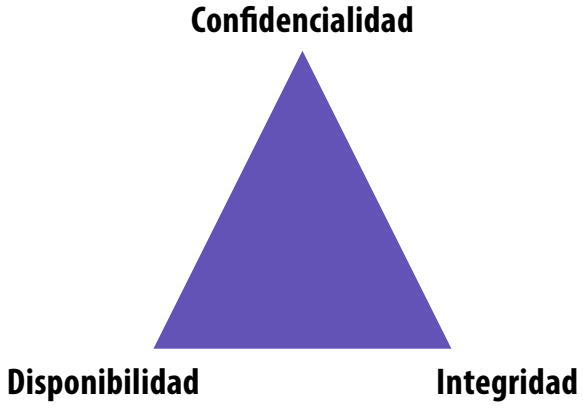


Figura 1. Triángulo de la seguridad de la información
Fuente: Propia.

La confidencialidad es la característica que se le da a la información para que esta permanezca en secreta tanto a personas como a sistemas que no tenga la autorización debida, es decir, que dicha información solo será conocida por las personas que tengan autorización. Por ejemplo, si usted hace una compra de un libro en el sitio Amazon.com, deberá pagar con su tarjeta de crédito, ¿qué sería si al transmitir los datos se produce una violación de la confidencialidad de sus datos? A esto es lo que se llama pérdida de la confidencialidad.

La integridad, hace referencia a que los datos no sean modificados por personal no autorizado, es decir, que la información se mantenga intacta todo el tiempo, que no sea manipulada o alterada por personas o sistemas que no estén autorizados.

Se presenta este fenómeno de violar la integridad de la información, por ejemplo, cuando con intención o sin ella, se modifican o borran los datos. Para que haya integridad de la información cuando se envía un mensaje, se utiliza por ejemplo la firma digital, así se hace más seguro el mensaje.

La disponibilidad, hace referencia a que la información se encuentre a disposición de quien la necesite, pero que cuente con la autorización.

Autenticación

El mecanismo de seguridad llamado autenticación, es aquel que verifica la identidad de un usuario cuando entra al sistema, la red o hace ingreso a una base de datos. Generalmente los usuarios acceden al sistema por medio de la contraseña y un nombre de usuario.

Hoy en día se utilizan otras maneras de autenticación como son:

a. Una contraseña, denominada, "por lo que el usuario sabe".
b. Una tarjeta magnética, denominada, "por las huellas digitales".
c. Las huellas digitales, denominadas, "por lo que uno es".

Entre más métodos se utilicen para la autenticación, más segura será ésta, pues todo dependerá del grado de importancia de lo que se quiera proteger.

Comúnmente se utiliza la contraseña para autenticarse en un sistema, pero esta deberá ser lo más segura posible, esto quiere decir, que dependiendo de las características de la contraseña será mejor o peor. Para que una contraseña sea segura, deberá ser larga y compleja, de esta forma no será violada ni descifrada tan fácilmente, por ejemplo, utilizar letras mayúsculas, minúsculas, números, caracteres especiales; el problema radica, en que no es fácil recordar una clave de esta magnitud, entonces se procede a tener como clave, el nombre del familiar, la mascota, el año de nacimiento de un allegado, y más aún, en las oficinas, escriben las contraseñas y las dejan debajo del teclado o la pegan en la pantalla del PC.

Verificaciones de integridad

Cuando se verifica la integridad de la información, es porque se han establecidos procedimientos para evitar o controlar que la información almacenada en archivos, sufran cambios que no han sido autorizados y que los paquetes o información enviada desde el emisor, llegue a su destino sin ser alterada.

La integridad de los datos hace referencia a la corrección y complemento de los datos en las bases de datos. Las instrucciones más utilizadas al ingresar a una base de datos como el "insert, update o delete", pueden modificar el almacenamiento que se tiene, es decir los datos pierden su integridad, por cuanto se le puede añadir datos que no son válidos a la base de datos.

Generalmente se utilizan algunas técnicas para controlar o mantener la integridad de los datos, por ejemplo, los antivirus, la encripción y las llamadas funciones hash.

Los antivirus proporcionan seguridad a la información siempre y cuando se actualicen con frecuencia. Cabe recordar que no todos los antivirus reconocen todos los virus.

La encripción hace referencia a la codificación de la información que se va a transmitir por una red, con el objeto de hacerla más segura y que en caso de ser interceptada no se pueda entender, solo será descifrada por la persona a quien va dirigido el mensaje. Estos procesos de cifrar y descifrar, se hace por medio de software especializado.

Las funciones hash, son utilizadas en la criptografía para dar mayor seguridad a la información que viaja por medio de la red. El hecho de venir cifrado un mensaje por medio de estas funciones hash, lo hace resistente a los posibles ataques.

Existen dos tipos de funciones hash. De una parte, los códigos de detección de modificaciones y por el otro, los códigos de autenticación de mensajes. Los códigos de detección de modificaciones, son aquellos que tienen como objetivo principal la verificación de la integridad de los datos, es decir, detectar que el mensaje no haya sido modificado, lo que quiere decir, que se verifica la integridad del mensaje enviado; mientras que los códigos de autenticación de mensajes, tienen por objetivo autenticar el origen del mensaje.

Mecanismos de no repudio

El repudio hace referencia a que ni el emisor ni el receptor nieguen que han participado en una comunicación.

En toda comunicación participan dos tipos de actores, de una parte el emisor (el que envía un mensaje) y de otra el receptor (el que recibe el mensaje). Entonces, el repudio puede clasificarse en dos tipos:

No repudio de origen: en donde se garantiza que el emisor no podrá negar que es quien envía el mensaje, pues el receptor, tendrá las pruebas.

Imagen 2. No repudio
Fuente: http://www.net753.com.ar/seguridad_it.html

No repudio en destino: en donde el receptor no podrá negar que ha recibido un mensaje, pues el emisor poseerá la prueba de que si recibió el mensaje.

Cuando se hacen negocios por Internet es importante el uso de este servicio, pues emisor y receptor incrementarán su confianza en las transacciones comerciales que realicen, pero para poder lograrlo, deben utilizar algunos mecanismos que les permite la confianza mutua:

- Autenticación: acción que reconoce al emisor de quien envía el mensaje, es decir a quien envía un documento o en su defecto la máquina que se comunica con otra a través de la red para solicitar algún servicio.

- Autorización: acción que ejecuta el control del acceso de los usuarios a sitios restringidos, a máquinas diferentes o a servicios, luego de haberse autenticado.
- Auditoría: acción que hace la verificación del correcto funcionamiento de las políticas, de la seguridad que se han tomado en la comunicación.
- Encriptación: acción que permite ocultar la información que se transmite por medio de la red o que se haya almacenado en los equipos, con el objeto de que terceros no autorizados, no puedan acceder a la información sin tener los algoritmos y clave desciframiento.
- Realización de copias de seguridad e imágenes de respaldo: acción que permite recuperar la información en caso de fallo.
- Antivirus: acción que permite proteger las máquinas y los datos de virus informáticos.
- Cortafuegos – firewall: se refiere a programas auditores para evitar conexiones no deseadas en los dos sentidos, desde la red a equipos o desde éstos hacia la red.
- Servidores proxy: son equipos de hardware con software especializado conectados posiblemente en la red interna de una empresa y una red externa, con el objeto de auditar y autorizar los accesos de usuarios a diferentes servicios como el FTP (transferencia de archivos), o el web (acceso a páginas en internet).
- Uso de la firma electrónica o certificado digital: elemento que permite garantizar la identidad de un usuario o máquina evitando el no repudio en la comunicación o en firma de documentos privados. Se usan actualmente también para realizar comunicaciones seguras entre una máquina de un usuario y los servidores internet, como por ejemplo, las páginas web de los bancos.
- Leyes dirigidas a la protección de datos personales: que permiten asegurar de forma obligatoria la confidencialidad de la información de las personas que albergan en sus bases de datos.

Estos servicios denominados de no repudio permite al emisor tener pruebas de que la información ha sido entregada al receptor; de la misma manera el receptor tendrá las pruebas del origen de los datos recibidos. En ambos casos, tanto receptor como emisor no pueden negar que son los autores del mensaje o de su recepción.

Un ejemplo concreto del servicio de no repudio es el correo electrónico, cuando se escriben las direcciones electrónicas del receptor del correo y por ende el correo cuando es recibido por el recepto sabe quién lo ha enviado.

Imagen 3. Envío de mensajes para el servicio de No repudio
Fuente: http://www.webmarketingid.com/tijuana/img/mail-marketing.jpg

Otros usos o aplicaciones

Para poder proteger la información se debe hacer desde que se crea, se modifique, se almacene, se procese o transmita, pues la información siempre deberá tener las tres características importantes para hacerla se-

gura como son: La confidencialidad, la integridad y la disponibilidad.

En nuestra vida diaria, los procesos criptográficos son propios de la identidad de los seres humanos que utilicen medios de comunicación, por ello, casi todos los utilizamos en nuestro diario vivir, por ejemplo:

Imagen 4. Comercio electrónico
Fuente: http://www.mordovmedia.ru/media/news/38276/53b27957148565a87c4669cb73120a79.jpg

- En el comercio electrónico: compra de libros u otros elementos por la red, pagos con tarjetas de crédito o débito, pago de impuestos, uso de dispositivos móviles.

Imagen 5. Comercio electrónico
Fuente: http://vantaivanchuyen.com/wp-content/uploads/2016/02/amazzon.jpg

- Cifrado de almacenamiento: bases de datos, dispositivos de almacenamiento, almacenamiento distribuido.
- Cifrado de comunicaciones: correo electrónico, redes inalámbricas, redes sociales, mensajería instantánea, intercambio de documentos, radio-tv.

Imagen 6. Cifrado de investigaciones
Fuente: http://mijobrands.com/ideas/2012/02/investigaciones-de-mercado-en-linea-consejos-y-herramientas/

- Cifrado de investigaciones: empresas dedicadas a la investigación cifran su información para tenerla más segura o investigadores que quieren tener sus secretos profesionales solo para ellos, mientras se les expide las patentes correspondientes.
- Cifrado en el sector financiero: pagos electrónicos con tarjetas crédito o debido, cualquier transacción bancaria: ver saldos, consignaciones, retiros, etc.
- Cifrado en sectores varios: votar electrónicamente, el pasaporte, los planes estratégicos.

Protocolos criptográficos

Los cifrados criptográficos o de cifrados, son protocolos de seguridad que permite a través de métodos criptográficos, prestar la seguridad necesaria para la información que se está enviando por la red.

La principal tarea de un protocolo criptográfico es la de indicar cómo se debe utilizar un algoritmo, describe sus estructuras de datos, representaciones e inclusive en dónde debe utilizarse para poder implementar versiones interopeables de un programa.

Los protocolos criptográficos se utilizan para enviar datos de forma segura. Cuando se utilizan estos protocolos deben incluir por lo menos, los siguientes elementos:

- Establecer claves.
- Autenticar entidades.
- Permitir cifrado simétrico y autenticaciones de mensajes.
- Transportar datos seguros en el nivel de aplicación.
- Soportar Métodos de no repudio.

Algunos protocolos criptográficos[1]:

- TransportLayer Security (TLS) es un protocolo criptográfico usado en conexiones web (HTTP) seguras.

4. Protocolo de establecimiento de claves:
    - Protocolo de Diffie-Hellman (establece de claves entre partes que no han tenido contacto previo, utilizando un canal inseguro, y de manera anónima (no autentificada).
    - ElGamal de negociación de claves provee negociación en un solo paso y autenticación unilateral (del receptor hacia el emisor) si la clave pública del receptor es conocida de antemano por el emisor.
    - El protocolo MTI/A0, 4 empleando dos mensajes y sin requerir firmas, proporciona claves de sesión variables en el tiempo con mutua autenticación implícita de claves contra ataques pasivos.
    - El protocolo STS5 es una variante de tres pasos de la versión básica que permite el establecimiento de una clave secreta compartida entre dos partes con mutua autenticación de entidades y mutua autenticación explícita de las claves. El protocolo también facilita el anonimato: las identidades de A y B pueden protegerse contra el adversario E. El método emplea firmas digitales.
5. Protocolos de autenticación:
    - Kerberos: es uno de los protocolos más utilizados en entornos internet. Identifica usuarios, los cuales están registrados en una gran base de datos encriptada, que no puede ser leída fuera del sistema. Cuando un sistema se comunica con un usuario, solo lo hace con el protocolo.
    - Radius: este protocolo autentica usuarios y es uno de los más antiguos. El protocolo ejecuta un software en el servidor, de tal manera que cuando un usuario se conecta a la red, un cliente Radius lleva los datos del usuario al servidor Radius de manera encriptada para que este los autentique, entonces, el servidor admite o rechaza al usuario entregando la respuesta al programa cliente de Radius.
    - Tacacs+: es otro protocolo de autenticación, muy similar a Radius, a diferencia de Radius, este protocolo en-

1 Protocolo de establecimiento de claves. Fuente: https://es.wikipedia.org/wiki/Protocolo_de_establecimiento_de_claves

cripta completamente los parámetros utilizados en todo el proceso de autenticación, este protocolo también encripta el nombre del usuario y algunos otros datos adscritos, a diferencia de Radius que solamente encripta la contraseña. Otra diferencia es que Tacacs+ es un protocolo de autenticación escalable, mientras que Radius es de autenticación independiente. Generalmente se utiliza Tacacs+ con Kerberos cuando las autenticaciones deberán ser muy fuertes.

Otra clasificación la hace, quien indica que los protocolos criptográficos se dividen en:

1. Protocolos de autentificación de usuario: permiten garantizar que el remitente de un mensaje o el usuario con el que establecemos comunicación es realmente quién pretende ser.
2. Protocolos de autentificación del mensaje: garantizan que el mensaje enviado no ha sido substituido por otro ni alterado (integridad del mensaje).
3. Distribución de claves: un problema importante en criptografía de clave privada es el de la creación y transporte de las claves a utilizar por cada par de usuarios. En cuanto a las claves de un sistema de clave pública, la problemática de su distribución es distinta (no es necesario el secreto), demandando protocolos específicos.
4. Protocolos para compartir secretos: su objetivo es distribuir un cierto secreto (por ejemplo la clave para abrir una caja fuerte), entre un conjunto P de participantes, de forma que ciertos subconjuntos prefijados de P puedan, uniendo sus participaciones, recuperar dicho secreto.
5. Pruebas de conocimiento cero: permiten a un individuo convencer a otro de que posee una cierta información, sin revelarle nada sobre el contenido de la misma.
6. Transacciones electrónicas seguras: permiten realizar de manera electrónica segura las operaciones bancarias habituales, firma electrónica de contratos, etc.
7. Compromiso de bit: permiten a una parte A comprometerse con una elección (un bit o más generalmente una serie de bits) sin revelar tal elección hasta un momento posterior. El protocolo garantiza a otra parte B que A no cambia su elección.
8. Transferencias transcordadas: permiten a una parte A enviar a otra B un mensaje o secreto entre dos posibles. Al no conocer cuál de los dos ha recibido realmente B.
9. Elecciones electrónicas: permiten realizar un proceso electoral electrónicamente, garantizando la deseable privacidad de cada votante y la imposibilidad de fraude.
10. Jugar al poker por internet: posibilita a dos personas, físicamente separadas, mantener una partida de poker (o similar: cara o cruz, chinos, etc.), comunicándose por correo electrónico, teléfono, etc., garantizando la imposibilidad de hacer trampa (Tena Ayuso, Juan. 2013).

Nociones preliminares al uso de algoritmos criptográficos

El cifrado es el acto por el cual un mensaje se encripta y desencripta, estos algoritmos se basan en el uso de llaves o claves, es así que los mensajes solo podrán ser desencriptados si la clave usada coincide con la que fue usada para encriptar el mensaje.

En la siguiente figura, se aprecia los principales algoritmos utilizados para encriptar información basados en llaves.

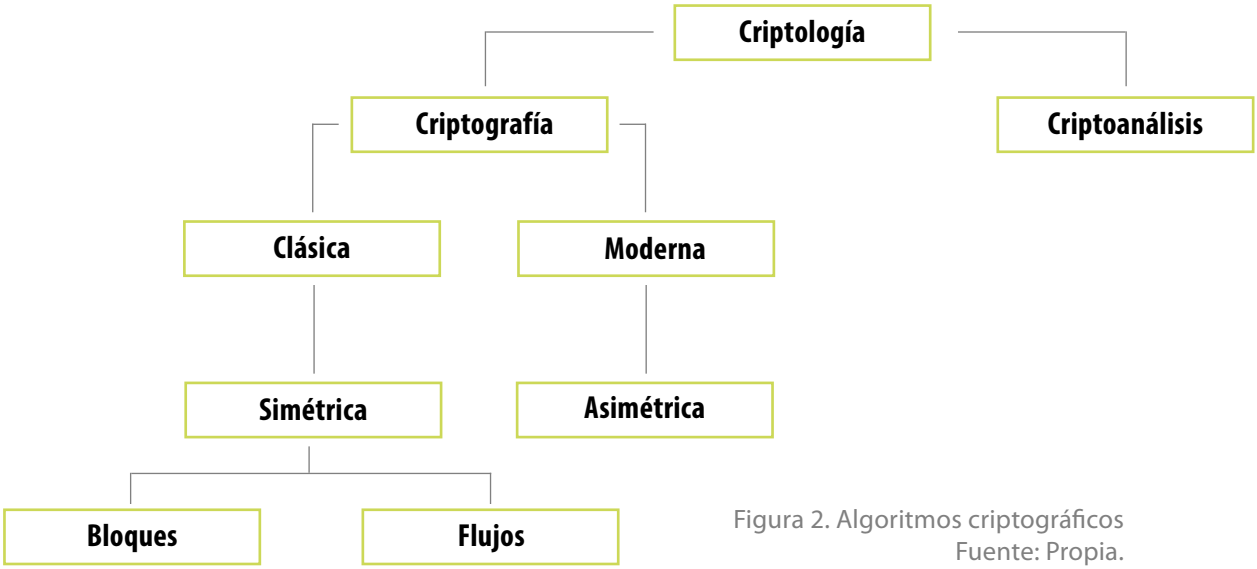


Figura 2. Algoritmos criptográficos
Fuente: Propia.

Los algoritmos para encriptar información basados en claves, son los algoritmos simétricos y los asimétricos, los primeros basados en llaves primarias y los segundos basados en llaves públicas. Los algoritmos simétricos se diferencian de los asimétricos porque utilizan la misma clave para encriptar y desencriptar información y los algoritmos asimétricos tienen llaves diferentes para encriptar y dencriptar y las llaves no podrán derivarse a partir de otra.

Los algoritmos simétricos se dividen en cifrado de flujo y de bloques. El primero encripta los mensajes bit a bit y el de bloques lo hace por medio de bloques de bits, generalmente lo hace con 64 bits, y lo encripta como una unidad simple. De otra parte, el cifrado asimétrico o de clave pública, tienen como llave para encriptar una llave pública, esto hace que cualquier usuario encripte datos con dicha llave, pero a quien se le dirige el mensaje deberá tener la llave privada para poderlo desencriptar.

Una característica importante de los algoritmos simétricos y asimétricos, es que son más rápidos los simétricos que los asimétricos para ser ejecutados por las computadoras.

También existe la encriptación híbrida, que consiste en que un algoritmo de llave pública utiliza para encriptar una llave randómica (generada al azar) y esta llave se utiliza para encriptar datos utilizando el algoritmo simétrico. Este algoritmo genera una llave pública y otra privada en el receptor, cifra el archivo de forma sincrónica, el receptor envía la clave pública al emisor, se cifra la clave que se utilizó para encriptar el archivo con la llave pública del receptor, luego se envía el archivo cifrado y la clave del archivo cifrada.

En síntesis, el cifrado híbrido utiliza cifrado simétrico y asimétrico, utiliza para ello una clave pública para compartir la clave para el cifrado simétrico. El mensaje será enviado con ciframiento que utiliza la clave. Como no es seguro utilizar la clave simétrica, en-

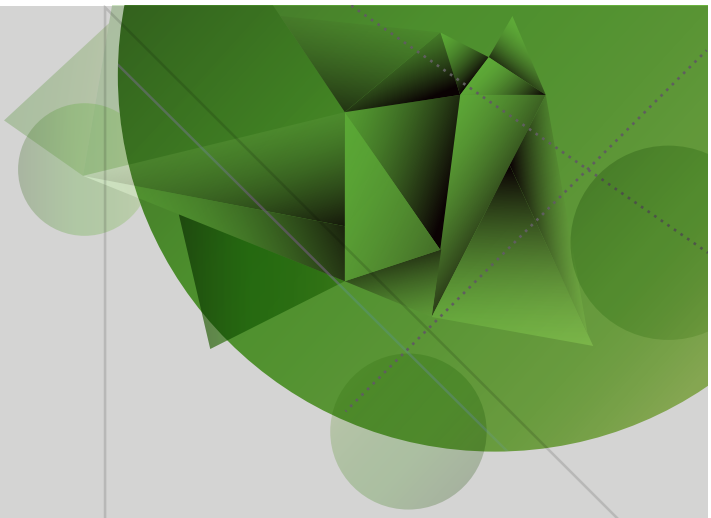
tonces, para cada envío se usa una clave distinta.

El algoritmo más popular que pertenece al algoritmo de cifrado simétrico es el Estándar de Encriptación de Datos-DES, pero debido a la capacidad de cómputo de las computadoras actuales, ya no se considera seguro, por lo que ha sido reemplazado desde el año 2000, por el Estándar Avanzado de Encriptación-AES, que hasta ahora es el algoritmo simétrico más usado.

2

Unidad 2

Herramientas de Desarrollo Disponibles



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

Luego de tener claro lo que significa proteger la información, se requiere aprender cuáles herramientas pueden utilizarse para desarrollar algoritmos criptográficos propios en los diferentes entornos de desarrollo.

Los framework son herramientas de gran utilidad, por cuanto ayudan al desarrollador de proyectos para agilizar su tarea, trabajar sobre una base estructural unificada, facilitan la escalabilidad y el mantenimiento del código.

Para lograr lo anterior, pueden revisarse entornos como los que ofrecen el software libre como es Netbeans, o el framework de Microsoft - NET que hace énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

También conviene revisar las herramientas de desarrollo que ofrece PHP para realizar desarrollos criptográficos como es el Laravel o el Yii, entre otros. Pero aún más importante, es conocer algunas herramientas que ofrecen proteger los datos en la base de datos, pues allí es en donde se almacena la información de vital importancia para una empresa o negocio.

Por todo lo anterior, se expone en esta cartilla algunos elementos que se deben tener en cuenta para que a la hora de realizar un desarrollo con alguna de las herramientas citas en párrafos anteriores, puedan escoger la más adecuada, dependiendo del objetivo del trabajo.

U2 Metodología

En este capítulo del módulo, se revisarán algunas de las herramientas para hacer desarrollos criptográficos en algunos de los entornos más utilizados.

Se trabajarán entornos de desarrollo como los que ofrece Java, .NET, PHP y por supuesto los más utilizados para las bases de datos.

Algunas de las recomendaciones a las que ha de poner más énfasis es a aprender algunas de las técnicas y consejos para elaborar código seguro que por ende llevará a que las empresas tengan más confianza en sus aplicaciones hechas a la medida.

Elabore el taller propuesto en el módulo y realice cada una de las actividades sugeridas en el orden indicado para comprender mejor la temática.

Recomendaciones puntuales:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Realizar el taller propuesto.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

U2

Desarrollo temático

Herramientas de desarrollo criptográficas para entornos Java

Las herramientas de desarrollo o también llamadas ambientes de desarrollo integrado o entornos de desarrollo interactivo - IDE- son aplicaciones informáticas que facilitan el trabajo al desarrollador de aplicaciones por medio de servicios integrales. Estos entornos tienen facilidades como los editores de código fuente, herramientas de construcción automática y depuradores; además, poseen facilidades para autocompletar código o compiladores e intérpretes.

Los entornos de desarrollo más comunes que existen para Java son el Netbeans y Eclipse.

Entorno de desarrollo Netbeans

El entorno de desarrollo Netbeans, está escrito en java y en él se puede compilar cualquier tipo de aplicación. Tiene soporte para JavaScript, mejora el desempeño en la programación, tiene soporte para MySQL para poder acceder a las bases de datos, también tiene soporte en Java Beans, al igual que para Ruby/JRuby.

El entorno de desarrollo Netbeans, se aprecia en la imagen 1, en él se encuentra hacia el lado izquierdo, el árbol de las de los proyectos que se desarrollan, se listan con cada uno de sus paquetes y clases. También se encuentran los Servicios que ofrece la herramienta como son: las bases de datos, los servicios Web y las tareas. De otra parte, se ve una pestaña para presentar cada uno de los archivos desarrolladores, estos, los presenta Netbeans como un árbol de carpetas y subcarpetas. En la parte izquierda inferior izquierda, se encuentra el Navegador, el cual mostrará elementos utilizados en cada uno de los proyectos.

En la parte superior derecha se pueden visualizar las aplicaciones que se estén desarrollando, tiene la opción de verlas todas, así como la que esté activa.

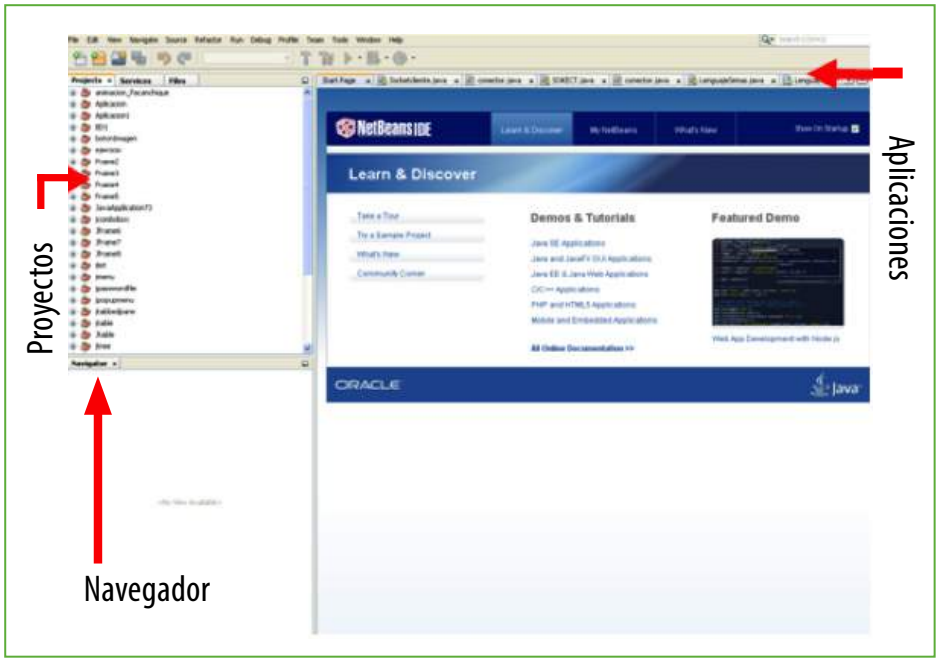


Imagen 1. Entorno de desarrollo Netbeans
Fuente: Propia.

En la siguiente imagen, se puede apreciar cómo se puede empezar a desarrollar un proyecto. En primer lugar, deberá escoger la opción de File- New Project. Se generará la ventana que aparece en la imagen 2, en donde deberá oprimir el botón Next para poder escribir el nombre del proyecto, que en este caso se llamará "cifrado".

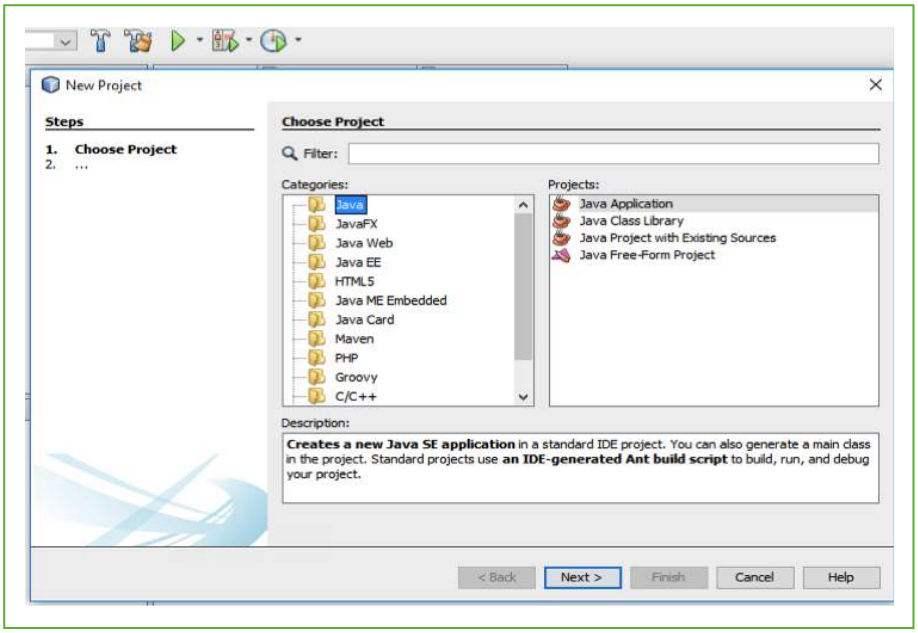


Imagen 2. Un nuevo proyecto en Netbeans
Fuente: Propia.

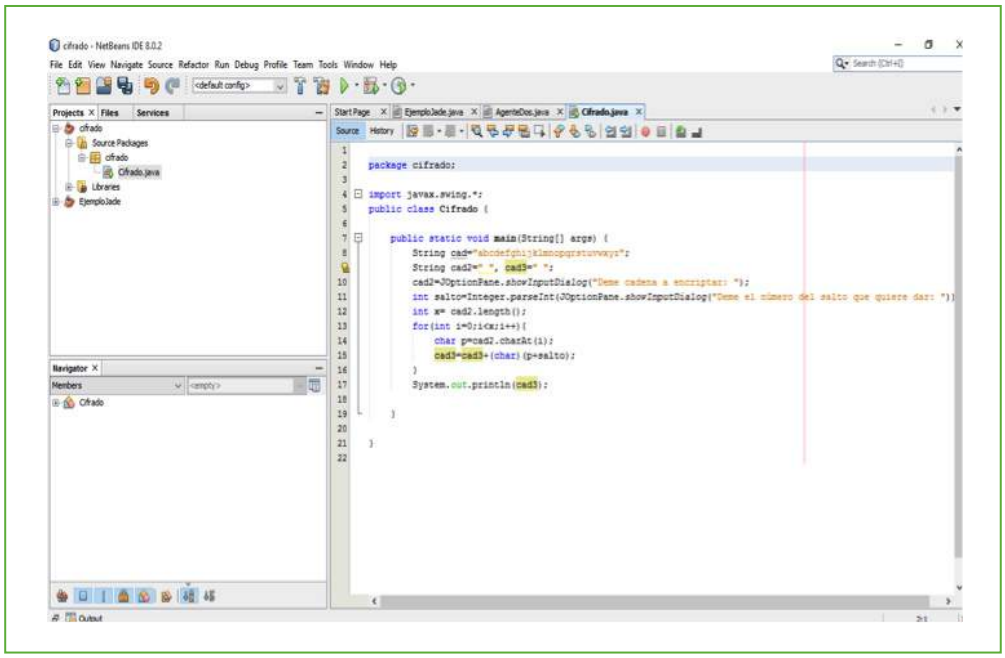


Imagen 3. Un programa escrito en lenguaje Java, sobre Netbeans, usando el cifrado César
Fuente: Propia.

Entorno de desarrollo Eclipse

De otra parte, el IDE Eclipse, permite trabajar múltiples lenguajes y ambos Eclipse y Netbeans, tienen plugins para C, C++, Ada, Perl, Python, Ruby y PHP.

El entorno de desarrollo Eclipse es un software de código abierto, es multiplataforma y permite trabajar aplicaciones de cliente llamas enriquecidas, a diferencia de las llamadas de cliente liviano, que se trabajan en los navegadores. Acepta entornos de desarrollo como son el Java Development Toolkit (JDT) y el compilador (ECJ).

En Eclipse se pueden compilar programas como C, C++, Python, LaTeX, JSP, Perl, Java, PHP, JavaScript, así como aplicaciones en red como Telnet y también sistemas de gestión de bases de datos.

Una característica importe de Eclipse es que puede utilizar GEF (Graphic Editing Framework), un Framework para la edición gráfica, como editores de diagramas como el UML, interfaces gráficas, entre otros.

Para iniciar un proyecto en Eclipse, basta con escoger la opción File (ver imagen 4.), luego New y escoger Java Project, como proyecto nuevo. En seguida se escribe el nombre del proyecto, que para este caso es "Proyecto 1" (ver imagen 5.) y como se observa en la imagen no se cambia ninguna opción de las que se encuentran marcadas por defecto, se oprime el botón Finish.

A continuación, en la imagen 6, puede observarse cómo se ve el "proyecto 1" acabado de crear. Como todo proyecto requiere de una clase para poder programar, se escoge también de la opción File-New-Class.

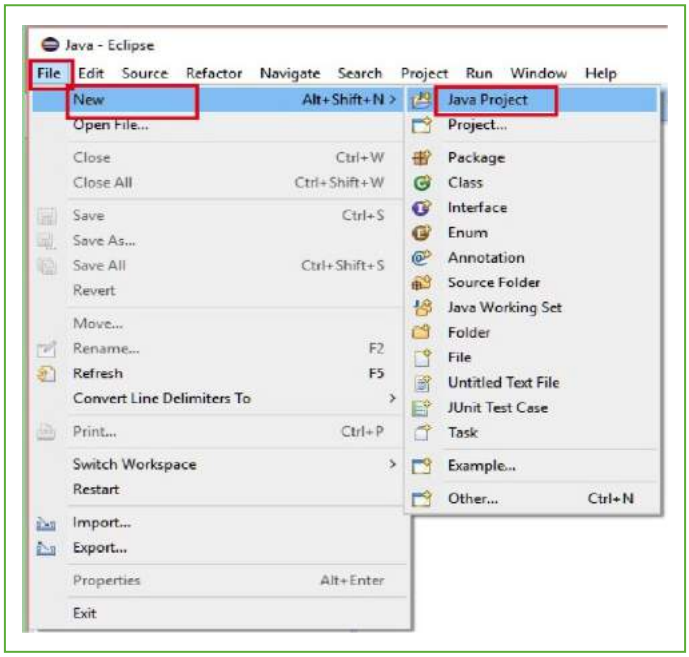


Imagen 4. Creando un proyecto en Eclipse
Fuente: Propia.

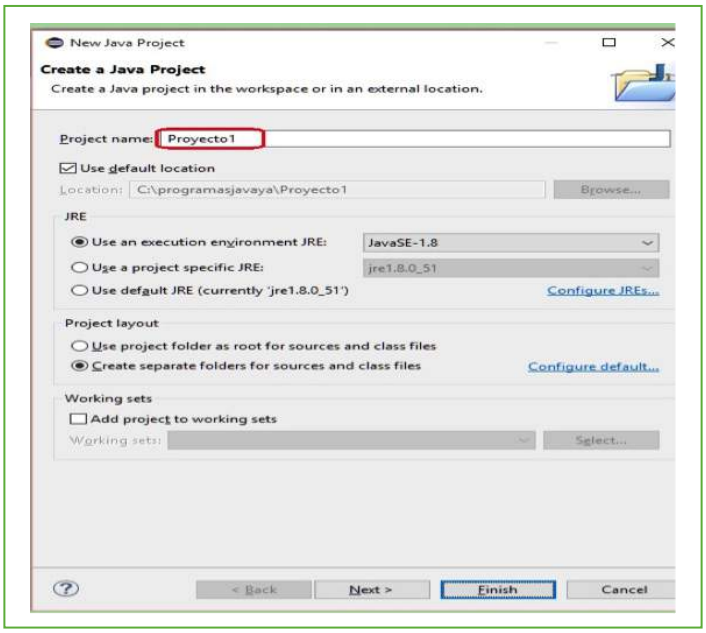


Imagen 5. Nombre de un proyecto en Eclipse
Fuente: Propia.

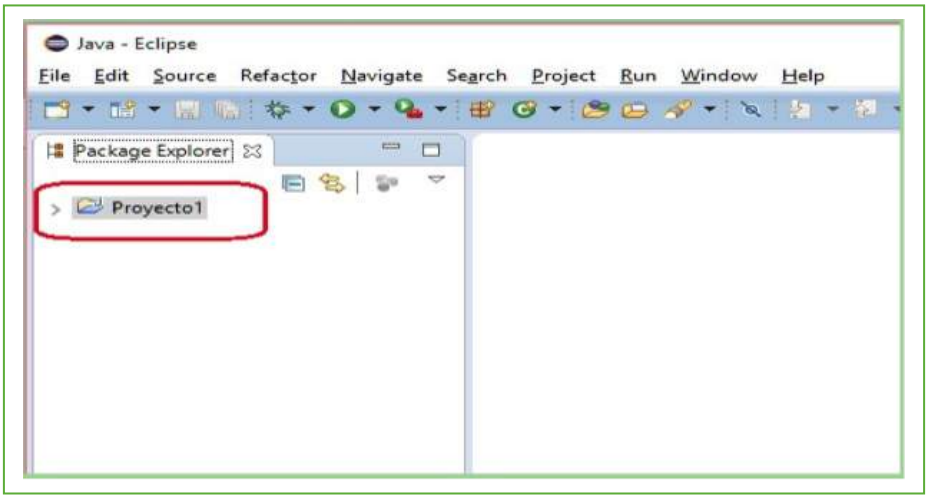


Imagen 6. Proyecto creado
Fuente: Propia.

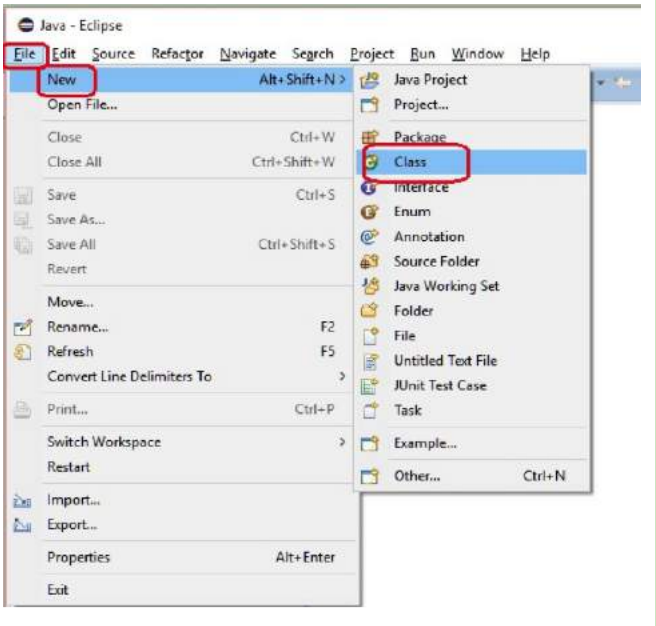


Imagen 7. Creando una clase en Eclipse
Fuente: Propia.

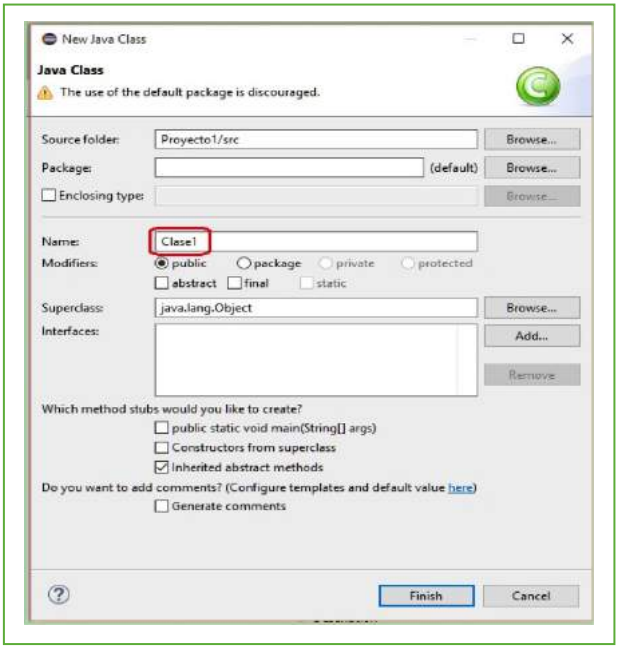


Imagen 8. Darle nombre a la clase creada
Fuente: Propia.

En la imagen 8, se muestra cómo se le da nombre a la clase, llamada "Clase 1", que pertenecerá al "Proyecto 1". Esta clase se crea, debido a que dentro de ella es en donde se escribe el código lenguaje, que en este caso es Java.

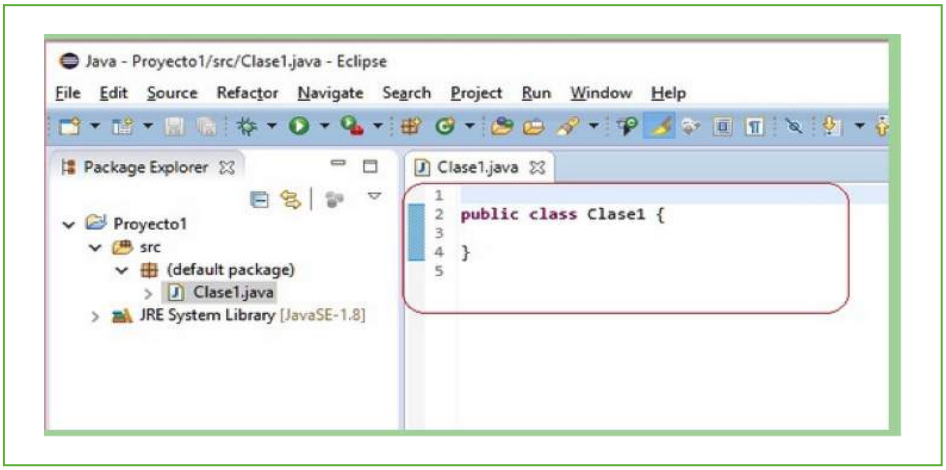


Imagen 9. Codificación de un programa en Java en el Framework Eclipse
Fuente: Propia.

En la imagen 9, se observa cómo puede empezarse a escribir una aplicación en java, estas aplicaciones pueden ser cifrados, descifrados, programas para establecer seguridad de algún sistema, programas a la medida, etc.

Herramientas de desarrollo criptográficas para entornos .NET

Visual Studio ofrece herramientas para desarrollos web, móviles y en la nube, también se pueden compilar aplicaciones para Windows, Android e IOs. Sobre estas herramientas pueden compilarse programas escritos en C·, C++, JavaScript, Python, TypeScript, Visual Basic, F#, entre otros. También permiten trabajar extensiones PHP y la elaboración de juegos.

Entre los lenguajes de .NET Framework que utilizan para desarrollar las aplicaciones, se encuentran: Visual Basic, C#, Visual C, Visual F#, JScript, Visual C++ en Microsoft, admite Servicios Web XML, al igual que AML un lenguaje de marcado para la programación declarativa de aplicaciones. También trabaja con ASP.NET que se fundamenta en las clases de programación de .NET Framework y entrega un modelo de aplicaciones web, al igual que un conjunto de controles, una infraestructura que hacen que la compilación de aplicaciones web resulte más sencilla.

También permite trabajar sobre Microsoft Ajax, que anexa una serie de bibliotecas de scripts de cliente que incorporan tecnologías entre exploradores ECMAScript (JavaScript) y HTML dinámico (DHTML). Las librerías de "Microsoft Ajax Library", se utilizan para compilar aplicaciones Ajax puras. También, también puede utilizar Ajax Library al compilar formularios Web Forms de ASP.NET o aplicaciones de ASP.NET MVC.

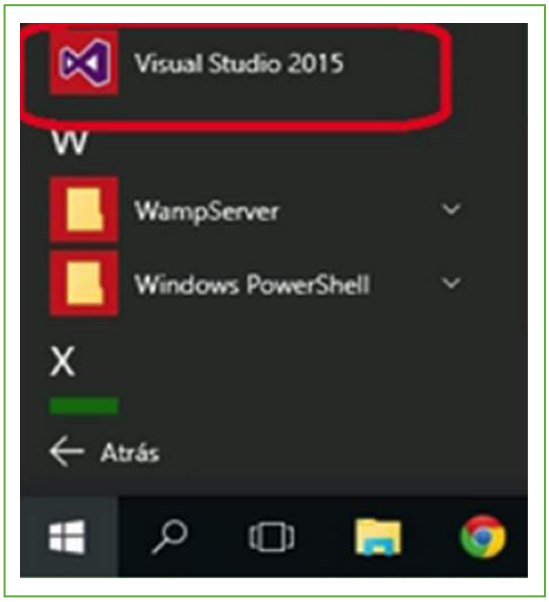


Imagen 10. Ingreso a Microsoft Visual Studio
Fuente: http://www.csharpya.com.ar/detalleconcepto.php?codigo=126&inicio=

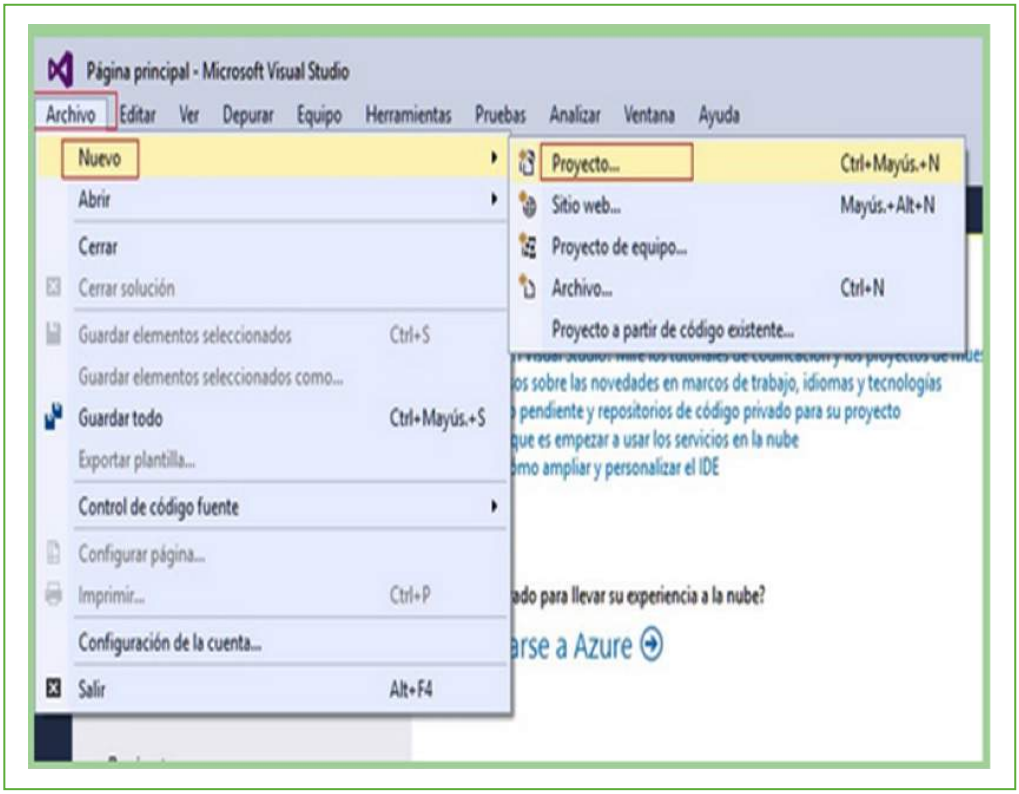


Imagen 11. Creación de un proyecto
Fuente: http://www.csharpya.com.ar/detalleconcepto.php?codigo=126&inicio=

En la imagen anterior, se observa cómo se crea un proyecto en Microsoft Visual Studio. Para ello, se selecciona el menú "Archivo", se escoge la opción "Nuevo" y escoge la opción "Proyecto".

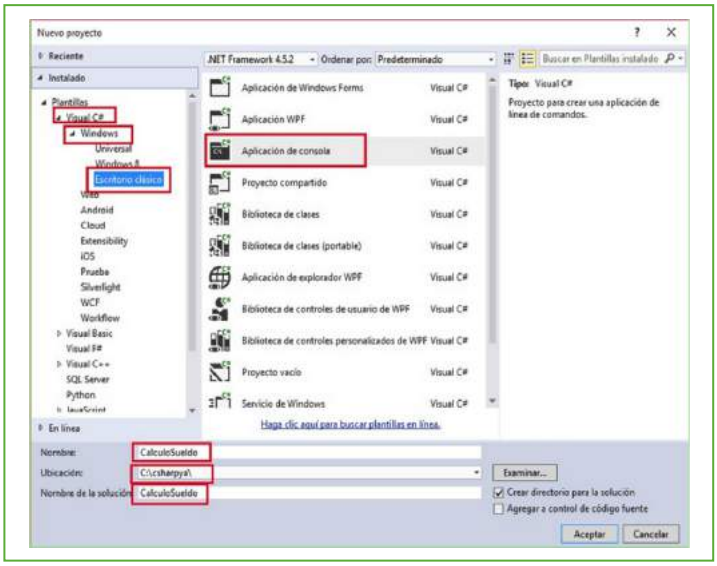


Imagen 12. Escoger tipo de aplicación a crear.
Fuente: http://www.csharpya.com.ar/detalleconcepto.php?codigo=126&inicio=

En la imagen anterior se visualiza cómo se escoge el lenguaje en que se va a desarrollar, en este caso es Visual C#, se escoge el sistema operativo en el cual se trabajará, como es el Windows y a continuación se escoge el tipo de aplicación que se va a realizar, como ejemplo, se toma una "Aplicación de consola", en la parte inferior de la gráfica, Nombre, se escribe el nombre que se le dará a la aplicación, la ubicación en la cual se almacenará el programa, así como el nombre de la solución, que generalmente se le da el mismo nombre de la aplicación.

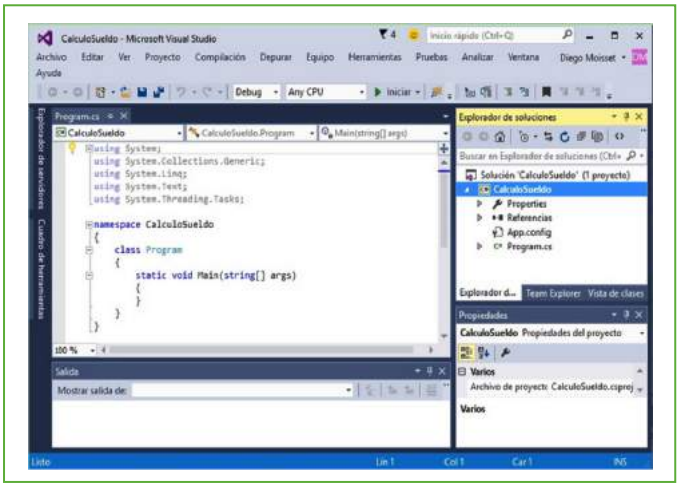


Imagen 13. Esqueleto automático para empezar a escribir el código
Fuente: http://www.csharpya.com.ar/detalleconcepto.php?codigo=126&inicio=

En la imagen anterior, se observa el entorno que se generó automáticamente, como es el esqueleto del programa.

Por último, luego de programar lo que se requiere y guardar la aplicación desarrollada, se cierra el entorno, como se observa en la imagen 14.

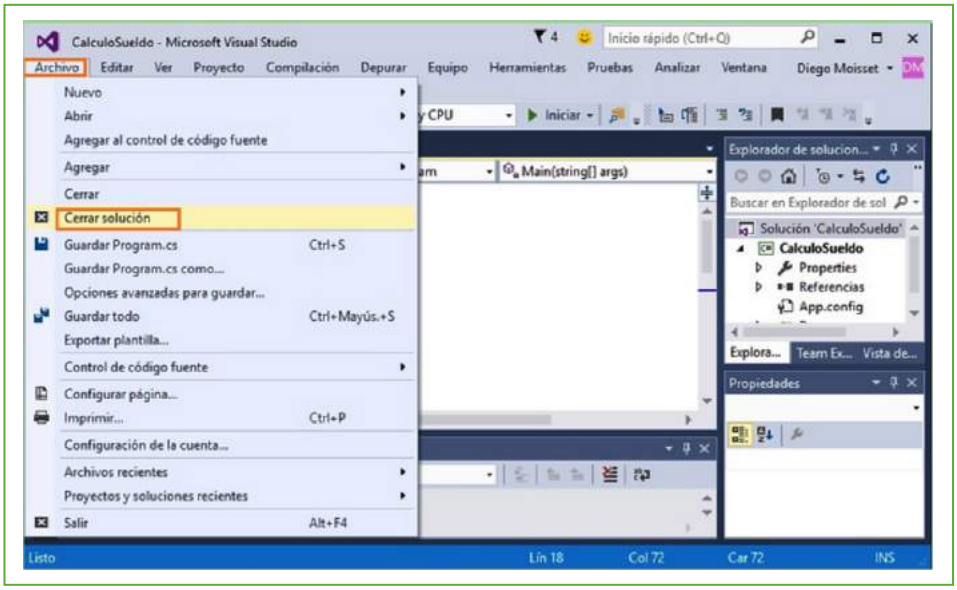


Imagen 14. Cerrando el proyecto
Fuente: http://www.csharpya.com.ar/detalleconcepto.php?codigo=126&inicio=

Para cerrar el proyecto se escoge el menú "Archivo" y se escoge la opción "Cerrar solución".

Herramientas de desarrollo criptográficas para entornos PHP

PHP es un lenguaje multiplataforma, que trabaja del lado del servidor, orientado al desarrollo de aplicaciones dinámicas y acceso a información en PHP, orientado a trabajos con objetos, en él se pueden desarrollar cualquier tipo de aplicaciones.

Como entornos de desarrollo o IDE para PHP, puede utilizarse los ya nombrados en numerales anteriores como son Netbeans o Eclipse, pero además, se puede utilizar alguno de los siguientes:

a. CodeIgniter. Utilizada para desarrolladores web.

b. CakePHP: en esta herramienta se elabora cualquier tipo de aplicaciones.

c. Symfony: usada por desarrolladores para hacer aplicaciones web complejas y escala cualquier aplicación realizada en una versión antigua de PHP.

d. Prado: herramienta usada para desarrollar aplicaciones web php5 y utiliza la programación orientada a objetos.

e. Qcodo: utilizada para trabajar con programación orientada a objetos completamente.

f. php.MVC: herramienta que utiliza el diseño modelo-vista-controlador-MVC- sirve para hacer aplicaciones web.

Zend Studio. Es un Integrated Development Environment -IDE - muy utilizado, es de código abierto, basado en eclipse e independiente de la plataforma para desarrollar proyectos. En la siguiente imagen 15 se aprecia el IDE Zend Studio.

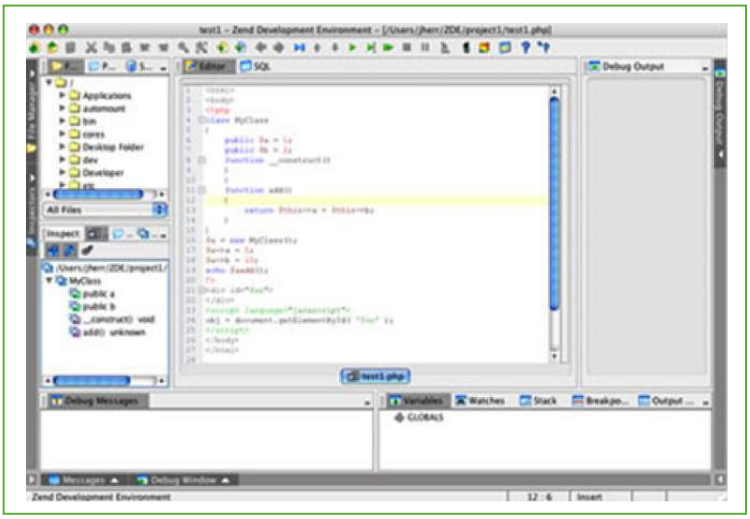


Imagen 15. Entorno de Zend Studio
Fuente: http://www.tufuncion.com/ide-php

De las plataformas más completas se encuentra Eclipse, es de código abierto e independiente, para PHP se utiliza la herramienta PHP Development Tools, en la imagen 16, puede verse las facilidades que permite dicho entorno.

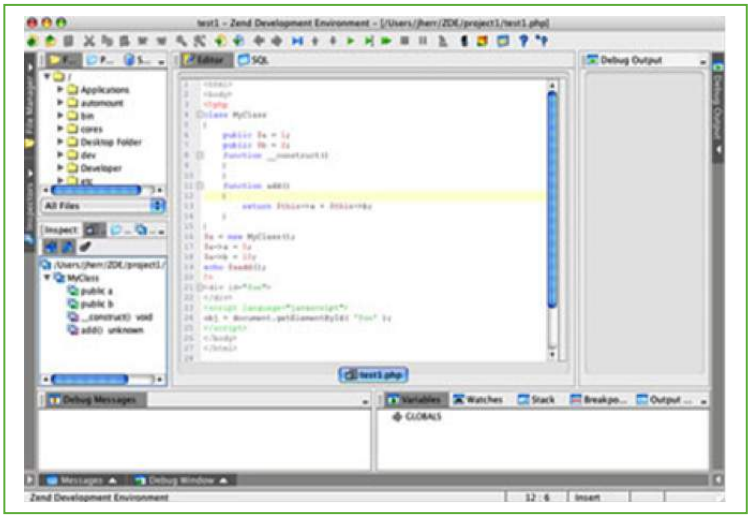


Imagen 16. Entorno PHP Development Tools
Fuente: http://www.tufuncion.com/ide-php

Otras herramientas que pueden servir para desarrollar en PHP: BlueShoes , Solar, Limb, Kohana, Phocoa, zajax, AjaxAC, Zoop, Seagull, Stratos, evoCore, Maintainable, Akelos y Zend Framework, entre otros.

Por último, un pequeño ejemplo del famoso saludo "Hola mundo, escrito en PHP.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title> Priemr ejemplo en PHP</title>
  </head>
  <body>
    <?php
      echo 'Hola mundo';
    ?>
  </body>
</html>
```

Herramientas de desarrollo criptográficas en bases de datos

Son varias las herramientas en las que se pueden desarrollar aplicaciones criptográficas y utilizar bases de datos. Algunas de ellas:

En PHP: se utiliza la base de datos MySQL y phpMyAdmin o también HeidiSQL, que es una herramienta sencilla, potente, no pesada y sobre todo libre. Otros entornos de desarrollo, Apache, Xampp, Wamp, Mamp, EasyPHP.

Como se nota, PHP tiene una gran capacidad de conexión con la mayoría de motores de base de datos actuales, pero es de destacarse la conectividad con MySQL y PostgreSQL.

Actualmente PHP soporte la conexión con servidores de bases de datos SQL como NoSQL, entre otros, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite o MongoDB.

De otra parte, existen muchos IDE de múltiples lenguajes tales como Eclipse, ActiveState Komodo, IntelliJ IDEA, MyEclipse, Oracle JDeveloper, NetBeans, Codenvy y Microsoft Visual Studio.

En cuanto al lenguaje de programación Java.

Un ejemplo para realizar la conexión entre java y MySQL:

```
Connection con = null;
    Statement stat = null;

    try {
      Class.forName("com.mysql.jdbc.Driver");
      con = DriverManager.getConnection("jdbc:mysql://localhost:3306/java1",
          "root", "");

      // Creamos un Statement para poder hacer peticiones a la bd
      stat = con.createStatement();
}
Catch(ClassNotFoundException | SQLException e){
 System.out.println("Error: " + e.getMessage());
}
```

Java y Netbeans permite también realizar conexiones con el Motor de Bases de Datos SQL. He aquí un ejemplo, donde se utiliza el driver correspondiente:

```
private Connection conn = null;
private Statement stm;
private ResultSet rs;
try {
Class.forName("org.gjt.mm.mysql.Driver");
conn = DriverManager.getConnection(url, user, password);
if (conn != null)
{
System.out.println("Conexión a base de datos "+url+" ... Ok");
stm = conn.createStatement();
}
}
catch(SQLException ex) {
System.out.println("Hubo un problema al intentar conectarse con la base de datos "+url);
}
catch(ClassNotFoundException ex) {
System.out.println(ex);
}
}
```

Si lo que se desea es conectar a una base de datos Oracle, se muestra en el siguiente código la forma de hacerlo, con el driver correspondiente.

```
try{
    Class.forName("oracle.jdbc.OracleDriver");
    String BaseDeDatos = "jdbc:oracle:thin:@localhost:1521:XE";
    Connection conexion= DriverManager.getConnection(BaseDeDatos,"HR","HR");
    if(conexion!=null)
    {
    System.out.println("Conexion exitosa a esquema HR");
    }
    else{System.out.println("Conexion fallida");}
    }
    catch(Exception e)
    {e.printStackTrace();}
  }
```

Cuando se quiera realizar una conexión con un motor de bases de datos PostgreSQL, se muestra el driver correspondiente:

```
String driver = "org.postgresql.Driver"; // el nombre de nuestro driver Postgres.
String connectString = "jdbc:postgresql://localhost:5432/ejemplo/"; // llamamos nuestra bd
String user = "postgres"; // usuario postgres
String password = " "; // no tiene password nuestra bd.
try {
Class.forName(driver);
//Hacemos la coneccion.
Connection conn = DriverManager.getConnection(connectString, user, password);
//Si la conexión fue realizada con éxito, muestra el siguiente mensaje.
System.out.println("Conexión a la base de datos Ejemplo realizada con éxito! ");
//Cerramos la conexión
conn.close();
}
//Si se produce una Excepción y no nos podemos conectar, muestra el sgte. mensaje.
catch(SQLException e) {
System.out.println("Se ha producido un error en la conexión a la base de datos Ejemplo! ");
}
```

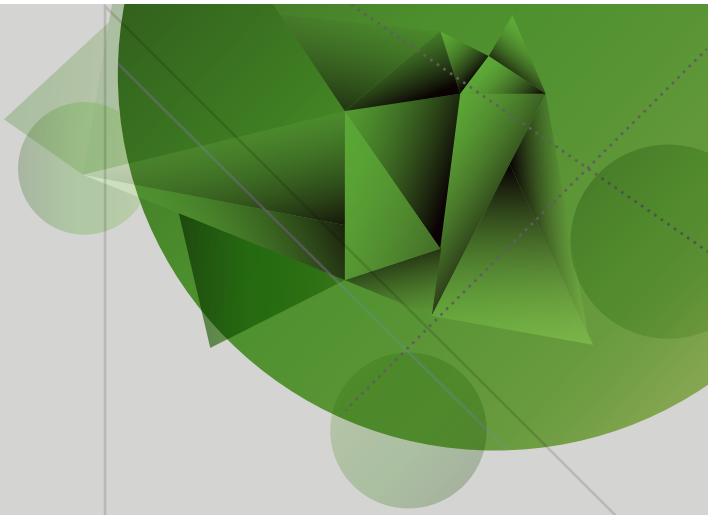
Ejercicio para realizar

Tome alguno de los IDE descritos, en el lenguaje que desee y realice la conexión a la base de datos que desee, para ello, utilice el driver adecuado, así como se señaló en el numeral 4.

2

Unidad 2

Protocolos criptográficos



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

Es esta unidad se trata el tema sobre los protocolos criptográficos, se realiza un recorrido sobre qué son y cuáles son dichos protocolos, así como los protocolos más conocidos como son los simétricos y asimétricos. También se revisan qué son y cuáles son los algoritmos criptográficos, las llaves criptográficas y cómo seleccionar un algoritmo en determinado caso.

Los temas vistos en esta unidad, son de vital importancia para la seguridad de la información cuando se transmite por medio de la red. Por lo anterior, los estudiantes deberán ejercitarse para aprender las temáticas aquí tratadas y ejercitarse en ellas a través de las aplicaciones que puedan reforzar sus conocimientos.

U2 Metodología

Para captar de mejor forma los tópicos aquí tratados, elabore cada una de las actividades aquí expuestas, lea esta cartilla con detenimiento y conciencia de su propio aprendizaje, ejecute las evaluaciones y hágase una retroalimentación para poder avanzar adecuadamente.

Recomendaciones puntuales:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Realizar las actividades de repaso y el parcial propuesto.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

U2

Desarrollo temático

Protocolos criptográficos

Introducción a los protocolos criptográficos

Los protocolos criptográficos tienen la tarea de velar por la seguridad de la información, para ello, utilizan métodos criptográficos. Estos protocolos indican cómo deben utilizarse los algoritmos de criptografía, e inclusive, algunos de ellos detallan las estructuras de datos, con el objeto de convertir en interoperables los algoritmos.

Generalmente, los protocolos criptográficos establecen las claves, autentican las entidades, incluyen cifrados simétricos, autentican claves, realizan el transporte de datos a nivel de capa de aplicación de forma segura y utilizan el método de no repudio.

Los protocolos criptográficos que trabajan a nivel de aplicación utilizan generalmente métodos distintos, de acuerdo a las claves, por eso se les llama protocolos criptográficos.

Los protocolos criptográficos tienen varias divisiones, dependiendo de la función que realicen, por ello, existen protocolos como los:

a. Protocolos de Establecimiento de claves o de intercambio de claves, los cuales establecen algunos pasos entre los participantes sobre la información que es secreta y quieren compartir y es lo que comúnmente se llama clave. Como ejemplo de estos protocolos, se encuentra el protocolo de Diffie-Hellman, el ElGamal de negociación de claves, MTI/A0 y el protocolo STS.

b. Protocolo de autenticación de usuario, estos protocolos garantizan que el emisor del mensaje o el receptor con quien se establece la comunicación es realmente quien pretende ser. Cuando se trabaja con servidores VNP, se establecen protocolos para intercambiar información e identificar el usuario, estos protocolos de acceso remoto son: PAP (Password Authentication Protocol), CHAP (Challenger Authentication Protocol), MS-CHAP (Microsoft CHAP), MS-CHAPv2 (Microsoft CHAP versión 2), EAP (Extensible Authentication Protocol), SPAP (Shiva Password Authentication Protocol). Se encuentra también el Kerberos, que tiene por objetivo identificar usuarios a través de bibliotecas de "claves" encriptadas que sólo asigna la plataforma Kerberos, así mismo se tiene también, el protocolo Radius-Remote Authentication Dial-In User Service, que autentica usuario y ejecuta un programa de software en un servidor, también combina la autenticación y autorización en el perfil de usuario.

Los protocolos de autentificación del mensaje tratan de garantizar la integridad del mensaje.

c. Protocolos de autentificación del mensaje, certifican que el mensaje enviado no ha sido substituido por otro ni alterado, es lo que comúnmente se llama integridad del mensaje.

d. Distribución de claves, la seguridad de las claves privadas del emisor y receptor es problemática y cuando se trata de claves públicas, se requiere de protocolos específicos, pues no se requiere el secreto de la clave.

e. Protocolos para compartir secretos, el fin de este protocolo es el de distribuir un secreto entre unos participantes de forma tal que un subconjuto de estos participantes puedan recuperar el secreto.

f. Protocolo de conocimiento cero o prueba de conocimiento cero, establece un método para interactuar concedido a una de las partes, con el fin de probar a otra parte que una declaración es verdadera y cierta, sin revelar nada más que la declaración, es decir, hace que una persona persuada a otro que posee una cierta información, sin revelarle nada sobre el contenido de la misma. Las propiedades que tiene este protocolo es la totalidad, cuando la afirmación es verdad, el que verifica es honesto y se convence al probar honestamente. La solvencia, cuando la declaración es falsa, no hay un probador engañoso que convenza al verificador honesto que es verdad. El conocimiento cero, cuando la afirmación es verdadera, un verificador engañoso no aprende otra cosa más que este hecho. Esto se formaliza mostrando que cada verificador engañoso tiene algún simulador que, teniendo en cuenta sólo la declaración a ser demostrada (y sin acceso al probador), puede producir una transcripción que "parece" una interacción entre el probador honesto y el verificador tramposo. (Wikipedia.org).

g. Protocolo de rellenado con paja y aventado, este protocolo utiliza una técnica denominada añadiendo paja y aventado, que en otras palabras no es más que a los paquetes con información real y verdadera se les agregan datos irrelevantes, con el objeto de que un intruso se le dificulte distinguir los paquetes irrelevantes de los que no los son.

h. Transacciones Electrónicas Seguras: las transacciones bancarias se realizan de manera segura electrónicamente, puede ser por medio de la firma electrónica.

i. Compromiso de bit: permiten a una parte A comprometerse con una elección (un Bit o más generalmente una serie de bits) sin revelar tal elección hasta un momento posterior. El protocolo garantiza a otra parte B que A no cambia su elección. (Tanenbaum, Andrew S. 2013).

j. Transferencias transcordadas: permiten a una parte A enviar a otra B un mensaje o secreto entre dos posibles. A no conoce cuál de los dos ha recibido realmente B. (Tanenbaum, Andrew S. 2013).

k. Elecciones Electrónicas: permiten realizar un proceso electoral electrónicamente, garantizando la deseable privacidad de cada votante y la imposibilidad de fraude. (Tanenbaum, Andrew S. 2013).

l. Jugar al póker por internet: posibilita a dos personas, físicamente separadas, mantener una partida de póker (o similar: cara o cruz, chinos, etc.), comunicándo-

se por correo electrónico, teléfono, etc., garantizando la imposibilidad de hacer trampa. (Tanenbaum, Andrew S. 2013).

Otros protocolos como el Transport Layer Security -TLS- es utilizado cuando se trabaja con conexiones web HTTP seguras, el mecanismo de autenticación se basa en el sistema X.509, posee una fase de configuración de claves, allí se trabaja con una clave de cifrado simétrico usando criptografía de llave pública y los datos son enviados a nivel de aplicación.

Protocolos de criptografía simétrica

Estos protocolos son también llamados Symmetric Key Ccryptography o criptografía de clave secreta.

Este tipo de protocolos, se basan en que utilizan la misma clave para cifrar y descifrar un mensaje en un canal inseguro. Tanto emisor como receptor, antes de comunicar el mensaje se ponen de acuerdo sobre la clave que van a utilizar, cuando el emisor cifra el mensaje con la clave acordada y el receptor lo recibe, éste lo descifrará con la misma clave que acordaron.

Recordando, el famoso código Enigma utilizado por los alemanes en la segunda guerra mundial, era un sistema simétrico, el cual a diario se entregaban las claves de transmisión en un libro de códigos, el cual los operadores de radio consultaban y retomaban el código del día para que emisor y receptor tuvieran la misma clave y pudieran comunicarse por medio de ondas de radio.

Un gran problema que tiene este tipo de criptografía es que si se desea tener un documento o mensaje secreto, privado entre un número grande de personas, se debe aprender o anotar en alguna parte, o intercambiar correos electrónicos con las claves secretas del mismo número de personas con quienes se va a comunicar, haciendo esta forma de comunicación muy frágil.

La fortaleza de los algoritmos criptográficos depende de la complejidad interna del mismo y de la longitud utilizada de la clave secreta, con el fin de evitar ataques denominados de fuerza bruta.

Los algoritmos simétricos tienen una ventaja y es su velocidad, pues se utilizan para cifrar grandes cantidades de datos. Es el caso de TrueCrypt que utiliza los algoritmos simétricos.

A continuación se relacionan algunos algoritmos de clave simétrica aunque algunos de ellos son considerados en esta época no seguros, vale la pena conocerlos por información general.

a. **Data Encryption Standard – DES**: es un algoritmo de cifrado monoalfabético, en el que se aplican permutaciones sucesivas y también sustituciones al texto original. La información de 64 bits se somete a una permutación inicial de y luego, a una de 8 bits y después a una sustitución de entrada de 5 bits, este proceso consta de diez y seis etapas de cifrado. Por lo anterior, esta algoritmo se denomina de clave simétrica de 64 bits, de los cuales, emplea los 56 primeros bits para el cifrado y los restantes ocho bits para la comprobación de errores mientras dura el proceso; lo que hace que solo 56 bits son utilizados como clave efectiva, esto quiere decir que existen 2 a la 56 posibles combinaciones posibles, que lo hace casi imposible de descifrar.

Ventajas	Desventajas
Es uno de los más utilizados y por tanto de los más probados.	Que la clave no puede ser de longitud variable, la cual se pueda incrementar para hacerla más segura.
Que tiene una implementación muy rápida y sencilla.	Si se conoce suficiente texto original o en claro y cifrado, es vulnerable al criptoanálisis llamado diferencia.
	La clave con una longitud de 56 bits es muy corta, lo que la hace vulnerable.
	Ya no es un estándar, porque en 1999 fue roto este algoritmo por una computadora.

Cuadro 1
Fuente: Propia.

b. **Triple Data Encryption Standard - 3DES:** su nombre lo recibe debido a que se aplica tres veces el algoritmo DES y la clave con longitud de 128 bits, la hace más segura que su antecesor, pero da la posibilidad de cifrar el mismo bloque de datos dos veces con claves distintas de 64 bits, esto hace que el tamaño de la clave se incremente, pero también se incrementan los recursos de la computadora. De este algoritmo se ha trabajado sobre una nueva versión llamada 3DES o mejor conocida como DES-EDE3, que trabaja con tres claves distintas y con una longitud de 192 bits, de esta forma el sistema será mucho más robusto.

c. **RC5:** en este algoritmo de cifrado simétrico, las longitudes de las claves cambian, igual que las iteracciones, esto hace que la seguridad del cifrado aumente, pero va de acuerdo con ese número de iteracciones, pues al aplicar las operaciones XOR sobre los datos, se alcanza a tener de 32, 64 o 128 bits. De otra parte, puede generarse números aleatorios, los cuales se les suma a los bloques de texto que se han rotado por medio de la operación XOR.

d. **International Data Encriptión Algorithm – IDEA:** este algoritmo utiliza claves de 128 bits sin paridad a bloques de datos de sesenta y cuatro bits, es utilizado para descifrar y cifrar mensajes. IDEA combina operaciones matemáticas como XOR, sumas con acarreo de módulo 2^{16} y multiplicaciones de módulo $2^{16}+1$, sobre bloques de 16 bits. (Wikipedia.org).

   IDEA es considerado como el mejor algoritmo de cifrado de datos, pues si se aplica fuerza bruta para descifrar el mensaje, se tendría que tiene 2 a las 128 claves privadas que se tendría que probar por fuerza bruta.

e. **Advanced Encryption Standard – AES:** es un método de cifrado utilizado por los usuarios de routers, por cuanto WPA opera con este método de cifrado. Es un método versátil, por cuanto se puede utilizar en hardware como en software y opera con bloques de claves de longitud variada que van desde 128, 192 y de 256 bits.

   El resultado intermedio del cifrado constituye una matriz de bytes de cuatro filas

por cuatro columnas. A esta matriz se le vuelve a aplicar una serie de bucles de cifrado basado en operaciones matemáticas (sustituciones no lineales de bytes, desplazamiento de filas de la matriz, combinaciones de las columnas mediante multiplicaciones lógicas y sumas XOR en base a claves intermedias). (Wikipedia.org)

De este algoritmo existen varias versiones como AES-CBC (Cipher-block chaining), AES-CFB (Cipher feedback) y AES-OFB (Output feedback).

f. **Blowfish:** es un algoritmo por bloques simétrico que utiliza 18 semiclaves y cuatro cajas, el proceso que utiliza es simple pero seguro pues aún no se ha podido aplicar criptoanálisis sobre él.

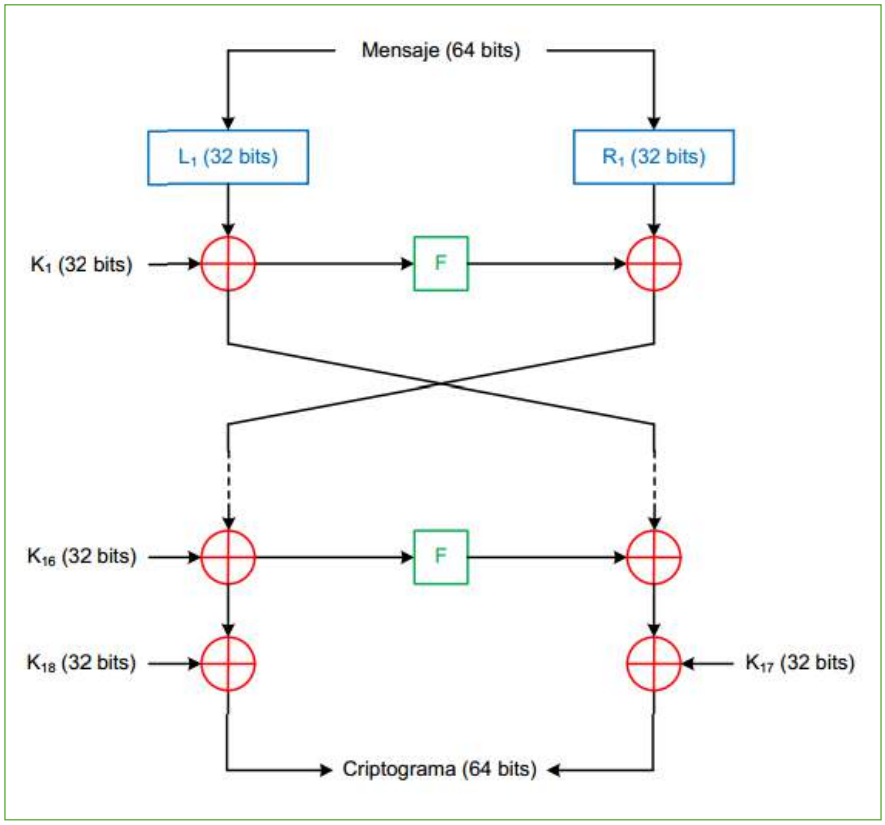


Imagen 1. Algoritmo de cifrado Blowfish
Fuente: naciek. (2013). http://lumbreras-criptografia.blogspot.com.co/2013/07/cifrados-por-bloque-blowfish.html

Funciones de una vía y Hash

Son funciones Hash que se trabajan por medio de algoritmos que cumplen que: H:U-> M x -> h(x), quiere decir, que existen unos elementos que generalmente son cadenas como entradas, se convierten en un rango de salida finito, denominado mapa, las cuales son también cadenas de longitud fija, esto quiere decir: la proyección del conjunto U sobre el conjunto M, donde M pueden ser un conjunto de enteros y se entiende que la longitud de la cadena es fija.

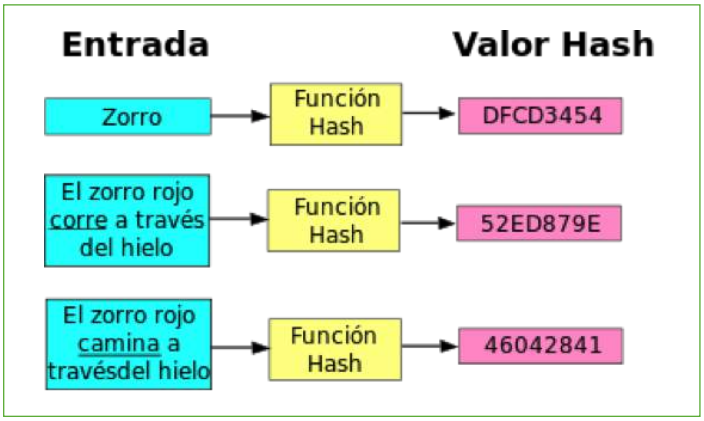


Imagen 2. Función Hash
Fuente: https://chuquispumasaunejesus.files.wordpress.com/2014/08/funciones.jpg?w=604

Protocolos de criptografía asimétrica

Los protocolos criptográficos asimétricos son la base de la criptografía actual, fueron inventados en 1976 por los matemáticos Whit Diffie y Martin Hellman.

Este tipo de criptografía simétrica utiliza dos claves, una clave primaria y una pública, estas son complementarias, funcionan así, cuando se codifica un mensaje con la clave privada, se requiere la correspondiente clave pública para su decodificación. De la misma manera, cuando se codifica con la clave pública se requiere de la clave privada para la decodificación.

Se denominan claves privadas porque solo son conocidas por el propietario y las claves públicas reciben su nombre porque se les puede conocer abiertamente.

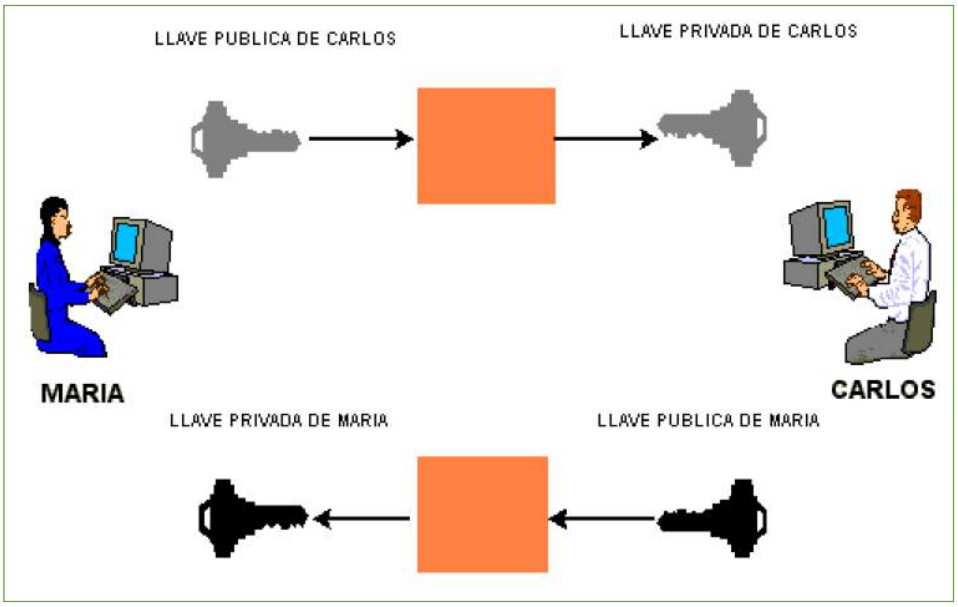


Imagen 3. Criptografía Asimétrica
Fuente: http://www.ongei.gob.pe/seguridad/seguridad2_archivos/lib5010/cap0506.htm

En la imagen 3, se presenta el funcionamiento del protocolo de criptografía asimétrica, María y Carlos poseen las dos claves, la secreta y la pública, pero si María codifica con la llave pública el mensaje para Carlos con la llave pública, Carlos deberá leerlo con la llave privada que solo él tiene y conoce; al contrario, si María envía Carlos un mensaje cifrado con la clave privada de María, entonces Carlos requiere la clave pública de María para descifrarlo.

Cuando se requiere saber la identidad del emisor y del receptor, se combinan las claves así, María puede enviar un mensaje a Carlos cifrado dos veces, con la clave privada de María y con la clave pública de Carlos.

El funcionamiento de la criptografía asimétrica tiene sus bases en el uso de números primos muy grandes, que si se multiplican, producen un número muy difícil de ser descompuesto, pues una computadora requeriría de miles de millones de años y tantos computadores como los átomos del universo. Por lo anterior, para que este cifrado surta un buen efecto sobre a información, se requiere que los números primos usados sean muy grandes.

Estos protocolos asimétricos tienen la desventaja que si el texto a cifrar es muy largo, la codificación se vuelve muy lenta.

Hoy se codifica el texto base con claves simétricas DES o IDEA, usan claves asimétricas para la comunicación de la clave simétrica usada.

Algunos algoritmos y tecnologías de clave asimétrica son:

a. **Protocolo criptográfico Diffie-Hellman:** en la imagen 4, se muestra el manejo del protocolo de Diffie-Hellman, el cual se basa en lo siguiente: (Wikipedia.org)

   La idea de que dos interlocutores pueden generar conjuntamente una clave compartida sin que un intruso que esté escuchando las comunicaciones pueda llegar a obtenerla.

   Para ello cada interlocutor elige un número público y un número secreto. Usando una fórmula matemática, que incluye la exponenciación, cada interlocutor hace una serie de operaciones con los dos números públicos y el secreto. A continuación los interlocutores se intercambian los resultados de forma pública. En teoría revertir esta función es tan difícil como calcular un logaritmo discreto (Un millón de millones de cuadrillones más costosa que la exponenciación usada para transformar los números). Por eso se dice que este número es el resultado de aplicar una función unidireccional al número secreto.

   A continuación ambos interlocutores utilizan por separado una fórmula matemática que combina los dos números transformados con su número secreto y al final los dos llegan al mismo número resultado que será la clave compartida.

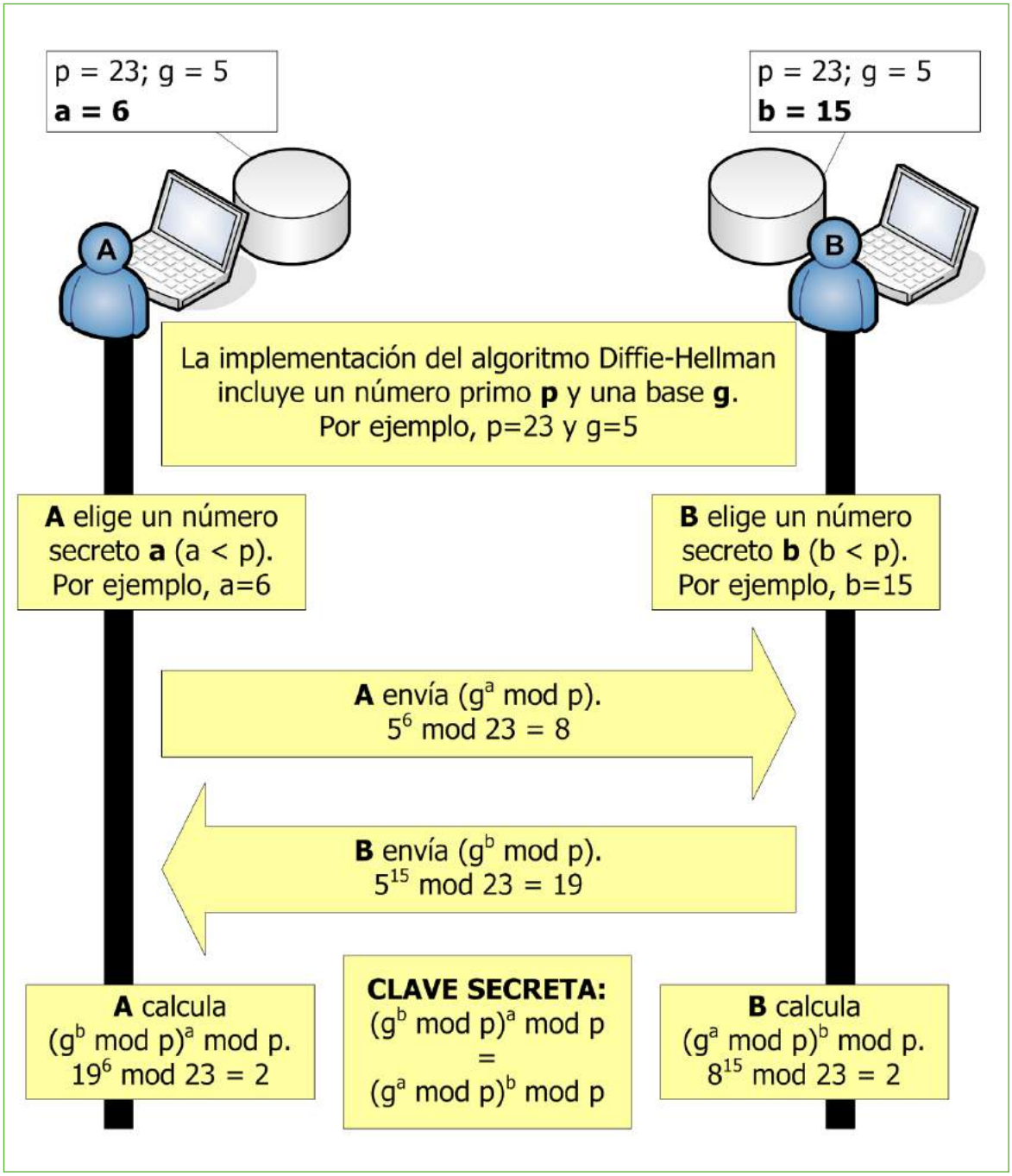


Imagen 4. Protocolo criptográfico de Diffie-Hellman
Fuente: http://www.javiercampos.es/blog/wp-content/uploads/2011/07/diffie-hellman.png

b. **Protocolo criptográfico RSA:** por sus siglas - Rivest, Shamir y Adleman-. Su uso es doble, para encriptar y para firmar digitalmente, utiliza clave pública en su funcionamiento, decir, tiene una clave pública y una privada, en donde quien envía el mensaje lo cifra con la clave pública de quien recibirá el mensaje, cuando se ha cifrado el mensaje y le ha llegado al destino, el recetor lo descifra con su propia clave privada.

   La seguridad de este algoritmo radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto, conocido, de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de diez a la 200, y se prevé que su tamaño crezca con el aumento de la capacidad de cálculo de los computadores. (Wikipedia.org)

c. **DSA Digital Signature Algorithm o también llamado algoritmo de firma digital:** sirve para firmar y no para cifrar información. Para mejorar la seguridad, se puede utilizar RSA y DSA simultáneamente, pero requiere más tiempo que el RSA.

d. **ElGamal:** usado para generar firmas digitales, para cifrar y descifrar. Este procedimiento se basa en cálculos cobre un grupo cíclico (G) que hace que la seguridad del mismo dependa de la dificultad para calcular los llamados logaritmos discretos en (G).

   Este esquema de firma, da la posibilidad que un verificador confirme la autenticidad de un mensaje enviado por un emisor sobre el canal de comunicación inseguro.

e. **Criptografía de curva elíptica ECC:** se llama así porque esta clave pública requiere de unas claves más pequeñas, esta característica hace que sea apta para sistemas que requieren más memoria.

f. **Criptosistema de Merkle-Hellman:** es un criptosistema de llave pública asmétrico, más simple que el RSA, no ofrece la opción para firmar y es un criptosistema que ya fue roto. Este criptosistema para la comunicación, requiere de una llave pública y una privada. Solo se utiliza la llave pública para cifrar, no para verificar firma y la llave privada solo descifra mensajes, no sirve para firmar, lo anterior indica que este criptosistema no se utiliza para autentificación por firma electrónica.

Protocolos de firma digital

La firma digital es una herramienta tecnológica para garantizar la autoría e integridad de documentos digitales, estas firmas asocian un mensaje digital para garantizar la identidad de la persona que firma, así como la integridad del mensaje. Una firma digital no asegura que el mensaje sea confiable.

Existen algunos algoritmos de firma que se basan en la criptografía de clave asimétrica para cifrar y descifrar, algunos de los más interesantes son:

Firma RSA: la firma digital RSA, se utiliza para el envío de mensajes, secretos o no con el objeto de que nadie los pueda modificar. Por ejemplo, Si Ana envía un mensaje a Raúl, Ana posee una clave pública de dos valores (a y b) y una clave privada (c).

Firma DSA: este estándar es utilizado para firmas digitales y no para cifrar la información. Su funcionamiento se compone de tres fases o etapas como son, la generación

de claves, la firma y la verificación. Tanto la generación de las claves como de la firma, las ejecuta el emisor y la verificación la ejecuta el receptor.

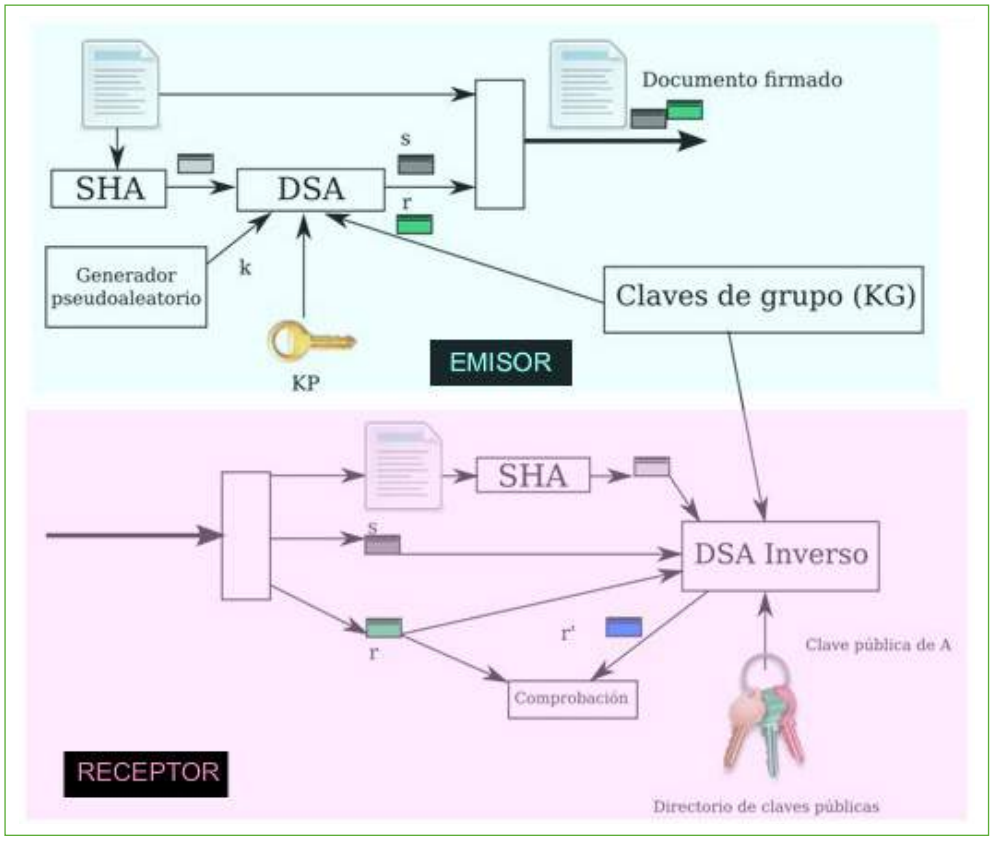


Imagen 5. Diagrama DSA.
Fuente: DSA (Digital Signature Algorithm). Tomado de http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/5-criptografia-asimetrica-o-de-clave-publica/56-firmas-digitales/562-dsa-digital-signature-algorithm

Otros protocolos de firma que se basan en la criptografía de clave asimétrica son:

- Firma ESING.
- Firma de clave asimétrica de Rabin.
- Firma ElGamal.
- Firma con curvas elípticas.
- Firma de Guillou-Quisquater.
- Firma de Ohta-Okamoto.
- Firma de Schnorr.
- Firma de Okamoto.
- Firma de Feige-Fiat-Shamir.

De otra parte, algunos protocolos de firma que se basan en la criptografía de clave simétrica son:

- Firma de Desmedt.
- Firma de Lamport-Diffie.
- Firma de clave simétrica de Rabin.
- Firma de Matyas-Meyer.

Generación de números aleatorios

Muchos de los algoritmos criptográficos requieren generar números aleatorios para hacer las claves, es decir, que en casi todos los sistemas criptográficos se generan números aleatorios para que los mensajes no se puedan adivinar por intrusos. Ejemplos en donde se utilizan los números aleatorios está la telefonía digital GSM, que se usa para asignar claves aleatorias para autenticar usuarios o dar seguridad a la asignación inicial de números secretos a las tarjetas de crédito.

La generación de números aleatorios es utilizada para generar claves de sesiones y fortaleza o calidad, para que la calidad de los sistemas sea fuerte, pues de ellos depende la fortaleza de las claves.

Si el generador de números aleatorios es roto, se volverá el punto más frágil del sistema criptográfico. Existen muchos generadores de números pseudo-aleatorios como el llamado Yarrow, la cual es usado ampliamente por el gobierno americano.

Otro cifrador de flujo muy usado es el RC4, trabaja con bytes y genera bytes aleatorios, opera también en modo sincrónico, es muy eficiente en implementaciones de software.

Introducción a los algoritmos criptográficos

Los algoritmos criptográficos son métodos matemáticos empleados para cifrar y descifrar mensajes, que operan con una o más claves, las cuales pueden ser numéricas o de cadenas de caracteres, utilizados como parámetros de los algoritmos, con el objeto de recuperar los mensajes a partir de la versión que se utilice para cifrar el mensaje.

Antes de ser cifrado un mensaje se le llama texto en claro, pero una vez ese texto se ha cifrado, se le llama texto cifrado.

Los algoritmos criptográficos modifican los datos de un documento o mensaje con el fin de lograr algunas de las características de seguridad como son, la autenticación, la integridad y la confidencialidad.

Son tres grupos en los que se clasifican los algoritmos criptográficos:

1. Criptografía simétrica o también llamada de clave secreta.
2. Criptografía asimétrica o también llamada de clave pública.
3. Criptografía Hash o de resumen.

En cuanto a la criptografía simétrica, estos encriptan y desencriptan el mensaje con la misma clave. Estos algoritmos son seguros y veloces, pues son mil veces más rápidos que los algoritmos asimétricos.

Algunos de los algoritmos simétricos ya tratados en esta cartilla son: DES, 3DES, RC2, RC4, RC5, IDEA, AES y Blowfish.

En cuanto a la criptografía asimétrica o de clave pública, se diferencia de la simétrica, porque utiliza una clave para encriptar y otra para desencriptar; ninguna de las dos claves depende la una de la otra. Las longitudes de las claves son muy superiores a las de las claves simétricas y su complejidad en el cálculo, hace más lentos los algoritmos asimétricos que los simétricos.

Algunos de los algoritmos asimétricos son: Diffie-Hellman, RSA, ElGamal y Criptografía de curva elíptica.

Hash: es una función que genera claves para representar un documento, registro,

archivo, mensaje u otro, que resume o identifica un dato por medio de la probabilidad usando una función Hash o también llamado algoritmo Hash y lo que resulta se denomina un hash.

La criptografía utiliza mucho este tipo de funciones, por ejemplo en las firmas digitales, en el procesamiento de datos y para encriptar mensajes o documentos. Algunos algoritmos criptográficos que utilizan funciones hash: MD5, SHA-1, DSA.

Matemáticas involucradas

La criptografía aprovecha las técnicas matemáticas para su uso, pues les sirven como herramienta para lograr sus objetivos. Tanto la matemática como la informática, le han dado a la criptografía los avances que ha logrado hasta el momento.

La relación entre la complejidad y la longitud de las claves se ha dado gracias a la matemática, por cuanto utilizan números randómicos y el tiempo necesario para encriptar y desencriptar los mensajes; de esta manera, la matemática ha sido clave para encontrar y definir con claridad sistemas criptográficos estables y seguros.

Otro uso que la criptografía hace de las matemáticas es en los algoritmos de encriptación simétricos, pues utilizan los métodos de transposición y permutación. Además los de algoritmos de clave pública se basan en complejas operaciones matemáticas.

Otro ejemplo claro, es el procedimiento de cifrado/descifrado ElGamal, que se refiere a un esquema de cifrado basado en el problema matemático del logaritmo discreto. Es un algoritmo de criptografía asimétrica basado en la idea de Diffie-Hellman y que funciona de una forma parecida a este algoritmo discreto.

A partir de los años 70 se han dado los aportes más notorios de la matemática en la criptografía de clave pública, pues se creó la criptografía simétrica AES, la cual es una de las áreas que trabaja el álgebra.

Uno de los usos de la matemática en la criptografía ha sido los números primos, por ejemplo en la RSA, generando las claves, también en DSA generando el dominio de parámetros para la firma digital, en el DH, se utiliza el esquema de intercambio de claves, al igual que las versiones elípticas EDSA y EDH.

Otro uso de las matemáticas ha sido a través de la factorización, es así que RSA basa su seguridad en la imposibilidad de factorizar un número de determinada longitud, producto de dos números primos. Últimamente la criptografía estudia en gran medida este tema, pues existen distintos métodos para factorizar números enteros y en muchas de ellas, es imposible factorizar en ciertas condiciones, un ejemplo concreto es el famoso número de Fermat.

En tercer término, Diffie y Helmman toman como suyo el Problema del Logaritmo Discreto PLD, el cual dio inicio a la criptografía de clave pública, la idea es encontrar un grupo particular donde PLD se defina y no pueda ser resuelto en tiempo polinomial, lo que trae consigo el diseño de sistemas criptográficos de clave pública muy seguros. También los campos finitos generales, han sido utilizados para implementar esquemas criptográficos y los conocidos como campos finitos de característica 3, han sido considerados en aplicaciones eficientes de firmas cortas dentro de la criptografía bilineal.

Otros usos de la matemática en la criptografía son las curvas elípticas, que hoy en día son muy utilizadas en la mayoría de estándares criptográficos. Al igual que las curvas hiperelípticas, que se utilizan en varios países comercialmente.

Otra área de la criptografía son los lattices o retículas, aquí se encuentra un vector con Longitud Mínima dada una lattice. Este tipo de criptografía se utiliza para el cifrado y firma digital, al igual, se usa en ataque a esquemas RSA.

Los mapeos bilineales, utilizados por los esquemas de firma digital, de cifrado, intercambio de claves, entre otras.

Un campo de investigación actual es el llamado grupo de trenzas o braid group, este grupo de trenzas puede definir PLD y de esta manera define un esquema criptográfico.

El nuevo tipo de criptografía denominado ecuaciones multivariables, las cuales tiene claves de longitud equivalente a la criptografía simétrica y además que su complejidad de ataque es exponencial.

Por último, la criptografía simétrica utiliza un estándar comercial llamado Rijndael, que basa su representación en 8 bits y las operaciones básicas se basan en campos finitos y anillos polinómicos.

Tipos block y stream de algoritmos criptográficos

Los algoritmos criptográficos, se encargan de modificar los datos de un documento para alcanzar características de seguridad como son la autenticación, integridad y confidencialidad. De estos algoritmos se tienen, los Cipher Blocks y los Cipher Stream.

Cipher Blocks - Cifrado por bloques, es un método encargado de encriptar texto, en el que la clave criptográfica y el algoritmo se aplican en bloque de datos, esto hace que se divida el texto original en bloque de bits con tamaño fijo y se cifran de manera independiente dando así como resultado cifrados que no son idénticos del texto cifrado, el vector de inicialización lo que hace es derivar el generador de números aleatorios que se combina con el texto en el primer bloque y la clave, esto asegura que los bloques posteriores no posean el mismo texto de la primera encripción.

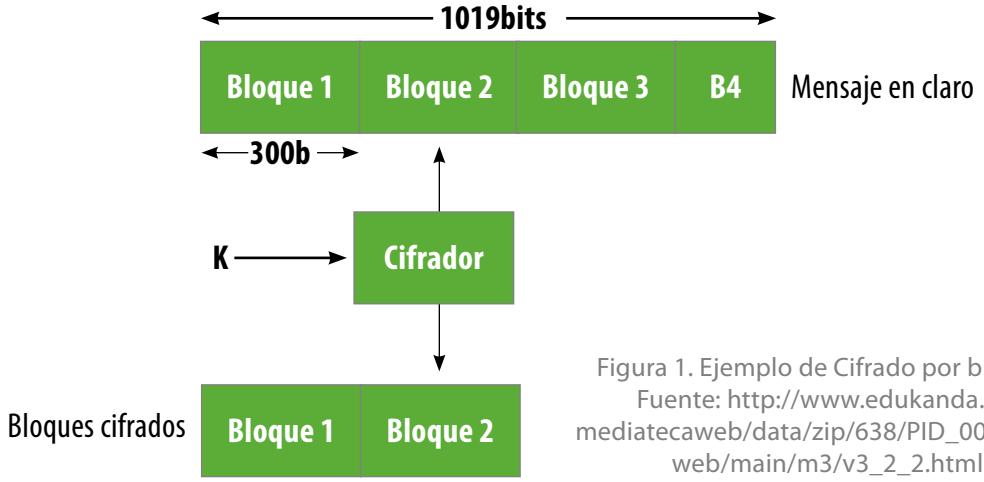


Figura 1. Ejemplo de Cifrado por bloques
Fuente: http://www.edukanda.es/mediatecaweb/data/zip/638/PID_00150197/web/main/m3/v3_2_2.html

En la figura anterior se presenta un mensaje inicial de 1019 bits, el cual se divide en tres bloques de 300 bits y el último de 119 bits.

Este tipo de cifrado es una unidad de clave simétrica, que trabaja con grupos de bits de longitud fija, al cual se les denomina bloques y se les aplica una transformación denominada invariante. Cuando se descifra el texto, se ingresan bloques de texto cifrado y se generan bloques de texto plano.

Como ejemplo de cifrado por bloques se tienen los siguientes algoritmos de cifrado: DES-Data Encription Standard, AES-Advanced Encryption Standard, IDEA- International Data Encryption Algorithm, RC2, RC5, Khufu, KHAFRE, GOST, SAFER, Blowfish, Akelarre, FEAL, Skipjack, 3DES.

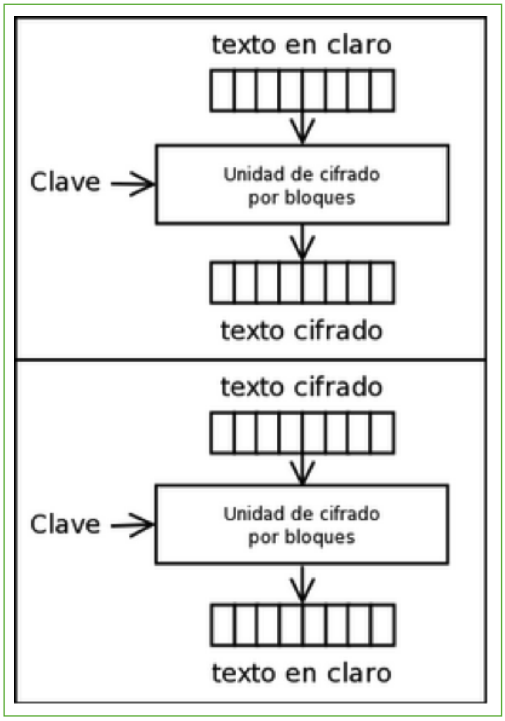


Imagen 6. Cifrado por Bloques
Fuente: https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Cifrado_por_bloques.png/250px-Cifrado_por_bloques.png

Cipher Stream - Cifrado de Flujo: este método de cifrado utiliza bit a bit o cada dígito binario, dado esto, no es muy útil en la criptografía moderna pero aun así es un buen método.

Este tipo de cifrado es usado en las telecomunicaciones, porque cuando se establece una comunicación vía teléfono móvil, la voz se digitaliza (se convierte a un flujo de bits) y se envía cifrada por la red de comunicaciones, pero es tan rápido el cifrado que los hablantes no se dan cuenta.

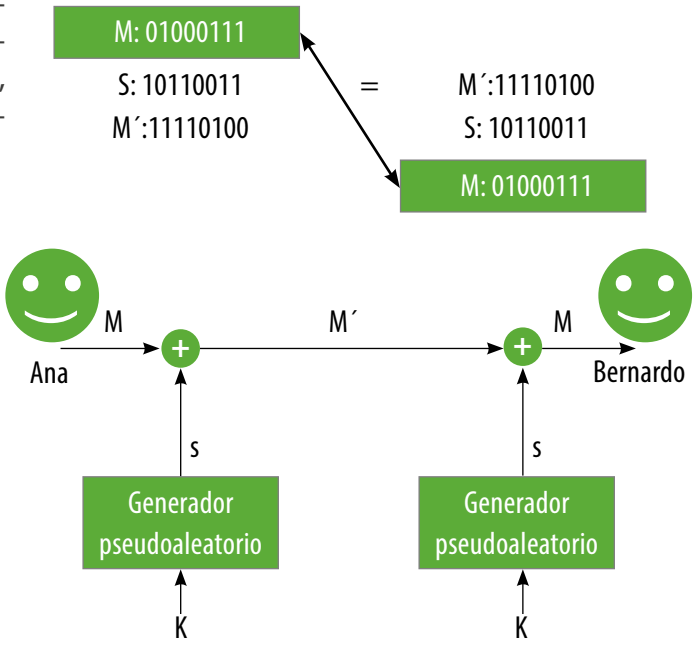


Figura 2. Cifrado de flujo
Fuente: http://www.edukanda.es/mediatecaweb/data/zip/638/PID_00150197/web/main/m3/v3_2_1.html

En la figura 2, se muestra que cada bit que entra al sistema, se combina utilizando a función lógica XOR con el bit que corresponda del flujo clave S y da lugar al bit correspondiente al flujo de salida. El receptor hace lo mismo, combina XOR y así obtiene el flujo.

La clave utilizada por este tipo de ciframiento, es la fortaleza del cifrado de flujo, pues se utiliza una clave aleatoria muy larga, que utiliza un generador pseudoaleatorio, cuyo algoritmo a partir de un mismo valor de entrada, se genere el mismo flujo de bits de salida, que se asemeja a una secuencia aleatoria.

Los métodos más conocidos de este tipo de cifrado es el de sustitución y transposición, como es el método de Julio César.

Modos ECB, CBC, CFB Y OFB de algoritmos criptográficos

ECB – Electronic codebook- Libro de código electrónico: al enviar un mensaje, este se divide por bloques, cada uno de ellos utiliza la misma clave (key), (ver imágenes 7 y 8), es decir que los bloques de manera individual son cifrados y una vez se unen estos bloques ya cifrados, se obtiene el mensaje completo cifrado. Cuando en este método se tienen bloques de "texto claro" iguales, los bloques cifrados son iguales, esto quiere decir que el patrón de los datos no es escondido por el método, esta es la gran desventaja de este método, pues lo hace sensible para reconocer los patrones guía y de esta forma descubrir el mensaje original a partir del ciframiento hecho. Este método es útil para datos aleatorios.

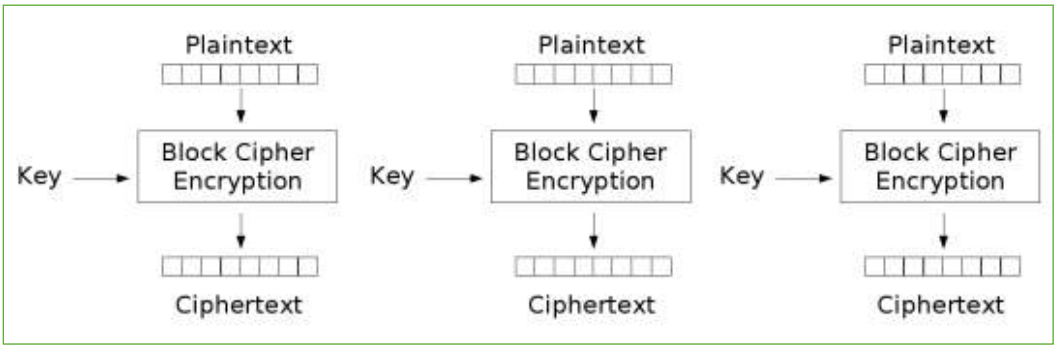


Imagen 7. Modo de encripción ECB – Electronic CodeBook
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques#/media/File:PCBC_encryption.svg

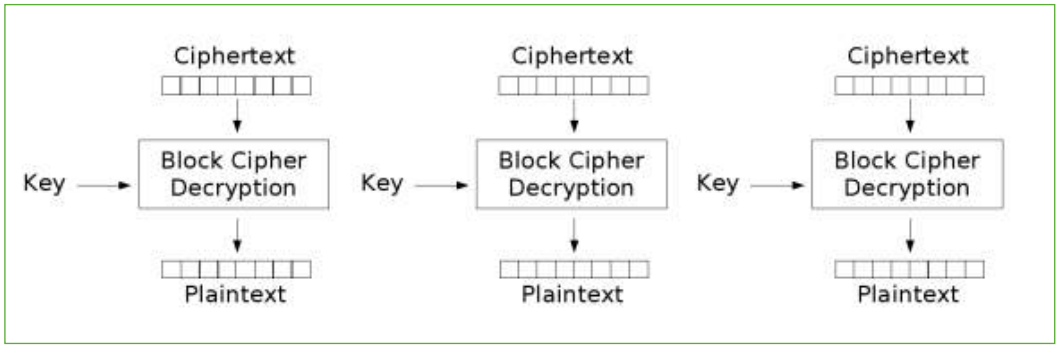


Imagen 8. Modo de desencripción ECB – Electronic Codebook
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques#/media/File:PCBC_decryption.svg

CBC – Cipher Block Chaining. Cifrado en Bloque Encadenado: se inicia con el "texto claro", el cual es dividido por bloques, a cada uno de ellos se le aplica la operación XOR del bloque obtenido en la etapa anterior, esto es que cada bloque encontrado dependerá de todos los bloques de texto anteriores. Aunque es muy utilizado en el ciframiento, tiene una desventaja que consiste en que es secuencial, pues el resultado de un bloque depende de otro y esa forma de trabajo dificultaría el procesamiento por bloques paralelos. Es muy útil para encriptar archivos, en donde la seguridad se incrementa respecto a ECB ver (imágenes 9 y 10).

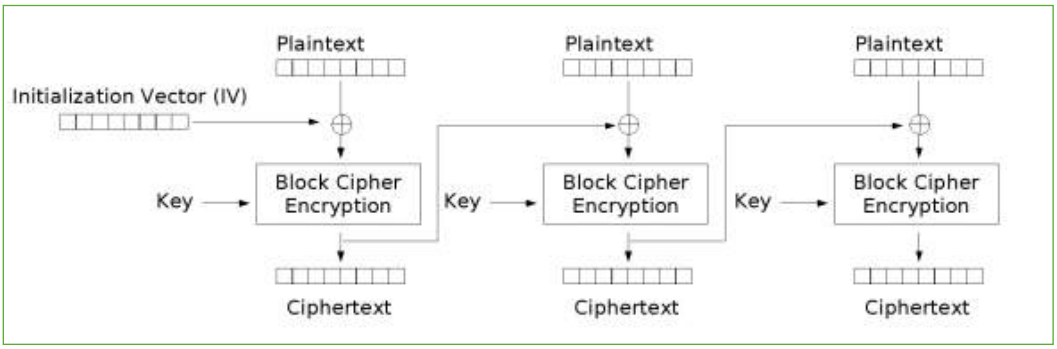


Imagen 9. Modo de encripción CBC – Cipher Block Chaining.
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques#/media/File:Cbc_encryption.png

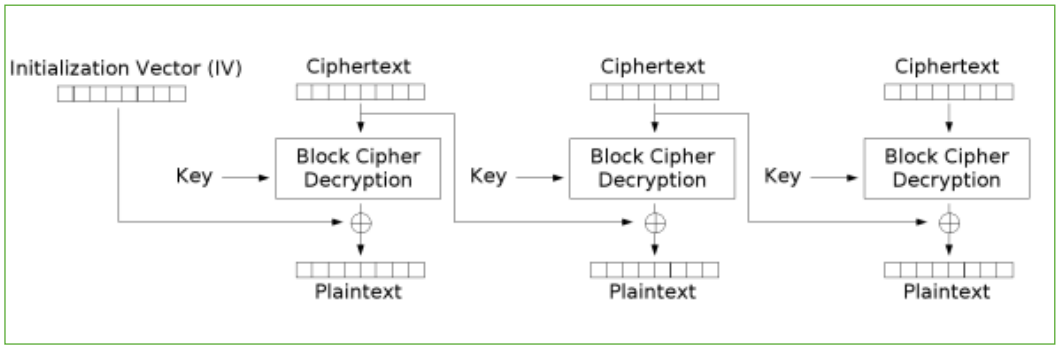


Imagen 10. Modo de desencripción CBC – Cipher Block Chaning
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques#/media/File:Cbc_decryption.png

CFB - Cipher FeedBack - Cifrado Realimentado y OFB - Output FeedBack: Estos métodos toman los bloques y los transforma en flujo, debido a que crean bloques de flujo de llave, una vez hace el cambio, se cifra el texto claro a partir de la operación XOR y así se obtiene el texto cifrado, para ello, cuando se intercambia un bit en el texto que se cifra, se produce un intercambio estrictamente en la misma posición del texto claro.

En el método CFB cada bloque de flujo de llave, se calcula cifrándose el bloque de texto cifrado anterior. Este método se usa principalmente para flujos continuos y es considerado la mejor manera de encriptar flujos de bytes, en donde cada byte se encripta.

OFB – Output FeedBack: este modo es muy útil para transmisiones de flujos continuos en ambientes ruidosos. Al igual que CFB utiliza el operador XOR para realizar las operaciones correspondientes una vez se utiliza la clave para crear un bloque pseudoaleatorio con el texto claro con el fin de crear un texto cifrado; para ello necesita un vector de inicialización que es exclusivo para cada construcción que realiza. Este modo se utiliza en aplicaciones donde la propagación de errores no puede soportarse (ver imágenes 11 y 12).

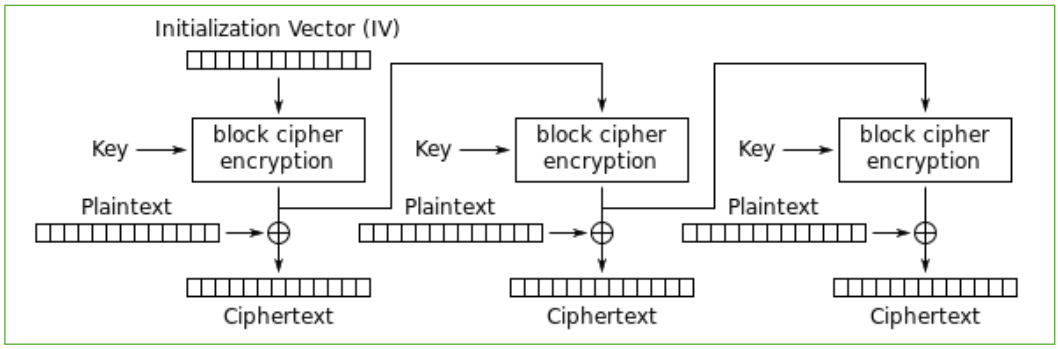


Imagen 11. Modo de encripción OFB – Output FeedBack
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques#/media/File:OFB_encryption.svg

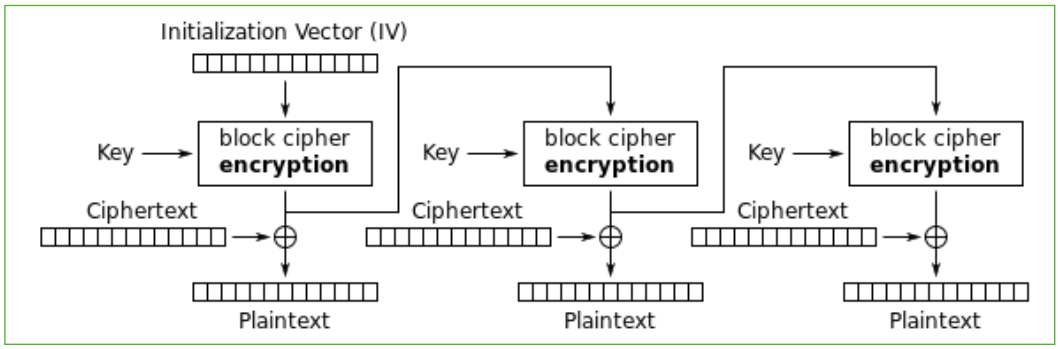


Imagen 12. Modo de desencripción
Fuente: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_una_unidad_de_cifrado_por_bloques

Relativo a las llaves criptográficas

Las claves o llaves criptográficas corresponden a información que controla los procesos de los algoritmos criptográficos, los cuales toman un texto plano o llamado texto claro y lo convierten en un texto cifrado o descifrado, para ello utilizan secuencias de letras o de números.

Estas claves las utilizan los algoritmos criptográficos como por ejemplo, las firmas digitales y los códigos de autentificación de mensajes, también llamadas funciones Hash.

Para seleccionar adecuadamente una clave segura se deben tener en cuenta varios factores a saber: la longitud, la cual, a mayor número de bits que la componen, el intruso deberá trabajar con más número de combinaciones con lo que usualmente se llama "fuerza bruta". Otra característica que debe tener la clave de seguridad es la aleatoriedad en lo posible con números, letras y caracteres especiales, si se utiliza este método surte un mejor éxito, pues se genera menor probabilidad de desciframiento de la clave. Un último elemento que se debe tener en cuenta para tener una clave segura es el periodo de uso, pues será más insegura esta clave cuando más dure su uso, por ello, es aconsejable cambiar las claves con la mayor regularidad.

La fuerza en la seguridad de un sistema cifrado se basa en que la clave permanezca secreta. Lo aconsejable cuando se tiene un sistema que almacena las claves para que un usuario se autentique, será más seguro si las claves están cifradas, inclusive, suele utilizarse como clave de cifrado la misma contraseña que se desea cifrar y cuando el usuario escriba la contraseña para realizar la autenticación, podrá descifrarse la contraseña almacenada; en caso de que la contraseña digitada por el usuario y la descifrada coincidan, el sistema permitirá el ingreso del usuario, pues la contraseña es correcta, en otro caso, el resultado del descifrado serán caracteres aleatorios, inentendibles para aquel que pretende atacar el sistema.

Las claves son el elemento más débil del sistema, pues si el intruso se hace a la clave por cualquier medio, podrá acceder a los datos cifrados del sistema.

Una frase puede utilizarse como clave para un sistema, pero éste debe por algún algoritmo de derivación de claves, expandiéndola o reduciéndola de acuerdo a la longitud que se desee, por ejemplo en un algoritmo de cifrado por bloques se utilizan claves de 128 bits, la cual es suficiente para que ocupe menos espacio y no se reduzca la seguridad. De otra parte, las funciones Hash utilizan un sistema de cifrado de contraseñas que producen resultados con la misma longitud, sin importar la longitud de la frase original, esto con el propósito de no inferir la longitud de la frase a partir de la longitud de la clave.

Una buena seguridad con algoritmos de clave simétrica se obtienen a partir de 80 bits, aunque si son de 128 bits, se considerará una clave muy segura.

Los sistemas de clave pública utilizan cierta estructura matemática para las claves de sus sistemas, pues se considera que no deben ser completamente aleatorias y que deben estar relacionadas entre sí, es el caso de los sistemas RSA, los cuales utilizan claves públicas generadas a partir de dos números primos. Lo anterior quiere decir que, los sistemas de clave pública necesitan longitudes mayores en sus claves que los sistemas simétricos, con el fin de dar una seguridad similar.

Una característica de los sistemas que se basan en factorización y logaritmos discretos es de 3072 bits, con los cuales se obtienen seguridades similares a un cifrado simétrico de 128 bits.

Los conocedores de la seguridad en los sistemas indican que el método de "libreta de

un solo uso" (es un algoritmo de cifrado en el que el "texto claro" se combina con una clave aleatoria de igual tamaño que el texto en claro y solo se usa una vez) es el único que ha sido probado por la matemática y ha sido seguro.

Selección de un algoritmo

Seleccionar un buen algoritmo de cifrado no es fácil, todo dependerá para que se desea utilizar, pues no existe ningún algoritmo que cubra todas las necesidades de un sistema. Suele tenerse en cuenta que para obtener un buen algoritmo seguro se han de tener buenos recursos de CPU, que las frases largas producen mejores resultados que las claves cortas; que los cifrados asimétricos son menos seguros que los simétricos con la misma longitud de clave, pero es mucho más lento; además que el cifrado por bloques con claves largas y complejas, son mucho más seguros que los cifrados de flujo. Un consejo es que cuando se cifren bastantes datos, se haga con una clave simétrica y a la vez cifrar la clave simétrica con una clave asimétrica.

Un buen algoritmo de cifrado puede ser, utilizar el cifrado AES de 128, 192 o 256 bits en el caso de querer realizar un backup online, pues con los procesadores actuales, una máquina se tardaría doscientos millones de años en dar con la clave AES de 128 bits.

La banca mundial utiliza el cifrado AES de 128 bits y la Agencia de Seguridad de los Estados Unidos también lo usa.

Los algoritmos más utilizados y seguros son los siguientes: DES-Data Encription Standard, algoritmo de 64 bits, emplea una clave de 56 bits en la ejecución, fue un algoritmo de cifrado simétrico muy empleado en el mundo.

El algoritmo Triple-DES encripta tres veces una clave DES, su grado de seguridad varía dependiendo del método que se elija para el ciframiento, es muy lento y por ello ha sido descartado.

RC2, es un algoritmo que reemplazó a DES, maneja ciframiento por bloques, la clave es de tamaño variable y trabaja en bloques de 64 bits, es mucho más rápido que el DES, si se elige el tamaño de la clave adecuada, se vuelve mucho más seguro que DES en ataques de fuerza bruta.

AES - Advanced Encription Standard o Estándar Criptográfico Avanzado, también es un algoritmo de cifrado que trabaja por bloques simétrico, reemplazó a DES, se puede utilizar con información clasificada. El tamaño de bloques que maneja AES es de 128 bits y sus llaves son de 128, 192 y 256 bits.

Ejercicio para realizar

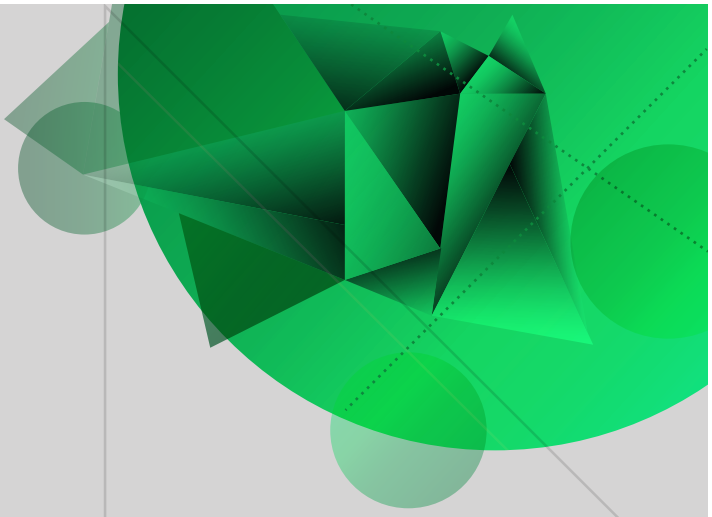
En el lenguaje de programación que usted domine, elabore un programa que resuelva la siguiente situación:

María quiere tener una clave muy segura a la cual Usted deberá realizarle un cifrado combinando dos números aleatorios que se encuentren entre 10000 y 600.000.000. Cuando genere los dos números aleatorios, intercale un dígito del primer número con el segundo, así sucesivamente hasta terminar. Esta sería la clave que María utilizaría para proteger su información, muestre por pantalla la nueva clave que usted le generó a María.

3

Unidad 3

Algoritmos de criptografía simétrica



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

El ciframiento de datos se ha dado desde tiempos remotos, con el fin de lograr transmitir cierta información de forma confidencial. Por lo anterior, han surgido métodos para que la información transite segura por medio de redes inseguras sin que le afecte algún ataque que intrusos quisieran realizar a la información.

Por lo expresado se han inventado métodos como los simétricos y asimétricos que ayuden a los usuarios a comunicarse de manera fiable y segura, para ello, han utilizado las matemáticas como medio para inventar claves o llaves seguras que aseguren la fidelidad de la información transmitida y que entre más segura sea la clave generada por los algoritmos basados en matemáticas, mayor confianza existirá entre un receptor y un transmisor.

En esta semana se revisarán con más detalle que en anteriores semanas, los algoritmos criptográficos simétricos, con el objeto que los estudiantes aprendan a diferenciar entre los algoritmos simétricos y asimétricos, los usos que se les da en la comunicación, la importancia de proteger la información, porque se trata del elemento más preciado por las personas, también se revisarán las características que cada uno de los algoritmos que pertenece a este tipo de criptografía tiene.

U3 Metodología

En este momento, los estudiantes tienen la fundamentación de lo que es la criptografía, sus tipos, el uso que ha tenido las matemáticas en este campo, han aprendido la terminología esperada para comprender el tema del ciframiento y lo que éste hace sobre la seguridad. Por ello, es pertinente que en este momento se conozca más a fondo los algoritmos simétricos, sus usos y características. Para lograr este cometido, se hacen las siguientes recomendaciones:

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Participar en el foro propuesto.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

Algoritmos de criptografía simétrica

Algoritmo DES y triple DES

Algoritmo DES

Inicios: Data Encryption Standard. Creado por la IBM en 1975 y el 1977 fue adoptado como un Estándar Federal de Procesamiento de la Información, recibiendo el nombre de FIPS PUB-46, por el hoy conocido Instituto Nacional de Estándares y Tecnología - NIST. A partir de 1981 la ANSI lo estandarizó como el ANSI X.3.92, duró como un estándar federal hasta 1998, año en el cual la Fundación de las Fronteras Eléctricas, con un ataque de Fuerza bruta, duró 56 horas para desencriptarlo.

El algoritmo de cifrado DES, trabaja por bloques de datos de 64 bits y usa una clave de 56 bits.

El algoritmo realiza combinaciones, sustituciones y permutaciones entre el texto que será cifrado y la clave; de igual forma se procederá para el descifrado.

Al codificar la clave de 64 bits, se reparte en 16 bloques de 4 bits. De los 64 bits, solo se usan 56 para el ciframiento y 8 bits para control de paridad, es decir para verificar la integridad de la clave; con estos 56 bits, se pueden generar 2 a la 56 claves distintas.

Como ejemplo, ver figura 1: Si se tiene el Texto Plano para el mensaje, y se encuentra en Hexadecimal **M= 0123456789ABCDEF**

M: *0000 0001 0010 0011 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111*

L: *0000 0001 0010 0100 0110 0111*

R: *1000 1001 1010 1011 1100 1101 1110 1111*

The first bit of M is "0". The last bit is "1". We read from left to right.

Let K be the hexadecimal key K 133457799BBCDFF1

K: *00010011 00110100 01010111 01111001 10011011 10111100 1101111 11110001*

Figura 1. Ejemplo del Algoritmo criptográfico DES
Fuente: Propia.

En la imagen 1, se muestra el funcionamiento del Algoritmo DES, con sus tres fases, en donde en la fase 1, se realizan las permutaciones del texto plano. En la fase 2, se realizan los desplazamientos, la sustitución, nuevamente se permuta y se realizan los intercambios de bits necesarios. Por último, en la fase 3, se ejecuta la permutación contraria o inversa a la inicial y se genera el criptograma.

Algoritmo Triple DES.

Este algoritmo también se le denomina TDES - Triple Estándar de Cifrado de Datos. En 1999 fue liberado como una mejora del algoritmo simétrico DES, por el NIST, este algoritmos recibe su nombre, porque ejecuta el cifrado DES tres veces y utiliza para ello, tres claves.

El algoritmo TDES fue inspirado en el DES, pero con la idea de hacerlo más seguro, ya que el DES solo utiliza 56 bits para la clave, mientras que el TDES lo elaboraron para trabajar con 168 bits como longitud de clave. Esta decisión fue tomada para evitar cualquier ataque de fuerza bruta, de tal manera que al agrandar la clave no se cambiara el algoritmo de cifrado, pero a los intrusos les costara más establecer un ataque. El TDES utiliza tres claves diferentes (K1, K2, K3), aunque puede en momentos usar dos claves, para ello hace que el K1 = K2, como se muestra en la imagen 2, 3 y 4, esta igualdad hace que la clave efectiva sea de 112 bits.

La seguridad del algoritmos TDES, es superior al DES, pero requiere más recursos tanto para cifrar como para descifrar un mensaje o documento.

De este algoritmo hay varios tipos a saber:

- DES-EEE3: este es un tipo de cifrado de 3 claves diferentes, también es un algoritmo TDES.
- DES-EDE3: este tipo de cifrado utiliza una clave distinta para cada una de las operaciones de triple DES, como son, el cifrado, el descifrado y el cifrado
- DES-EEE2 y DES-EDE2: este tipo de cifrado, utiliza una clave distinta para la segunda operación, es decir, para el descifrado.

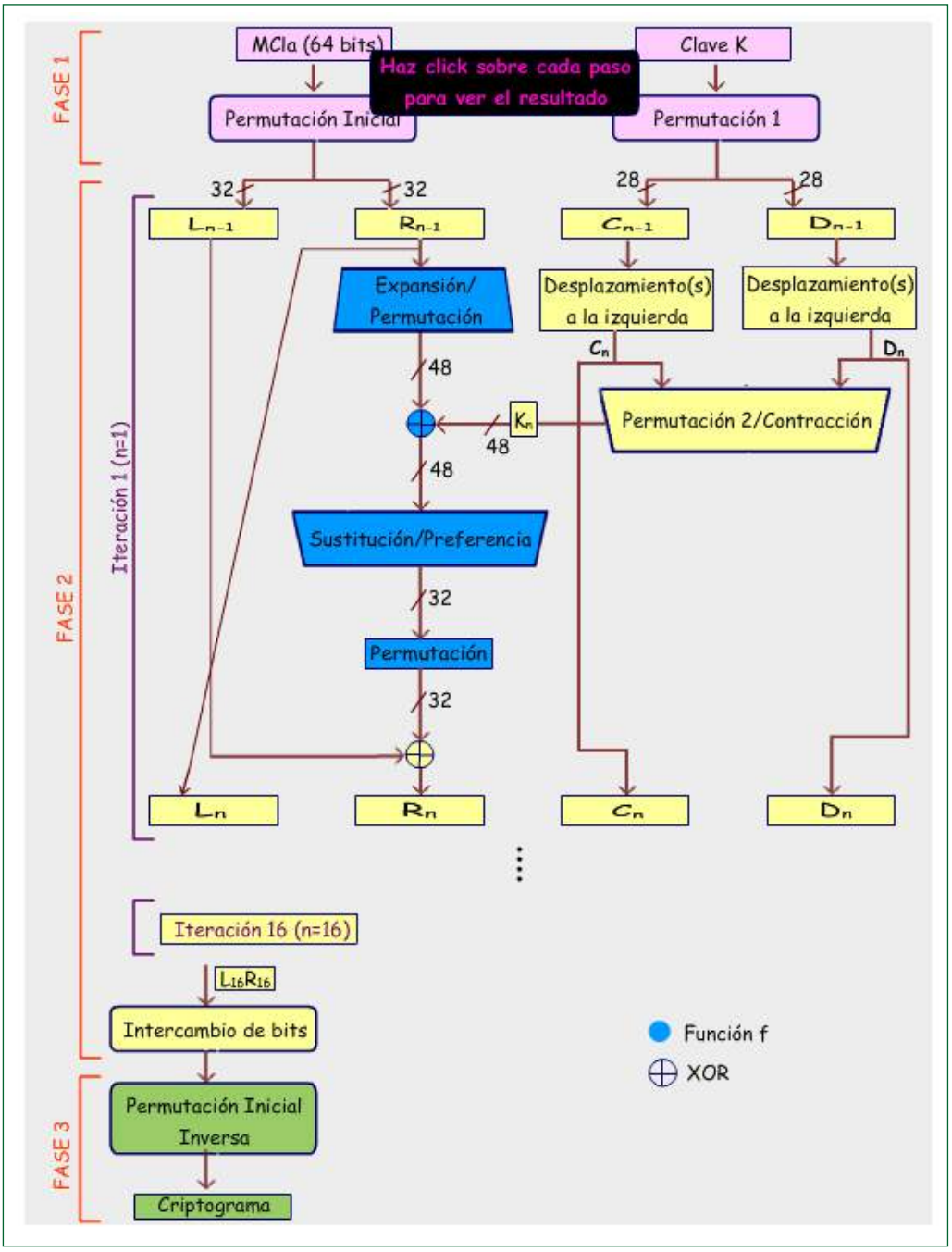


Imagen 1. Etapas del Algoritmo DES.
Fuente:
http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/4-criptografia-simetrica-o-de-clave-secreta/42-des-data-encryption-standard/424-aplicacion-del-algoritmo-caso-practico

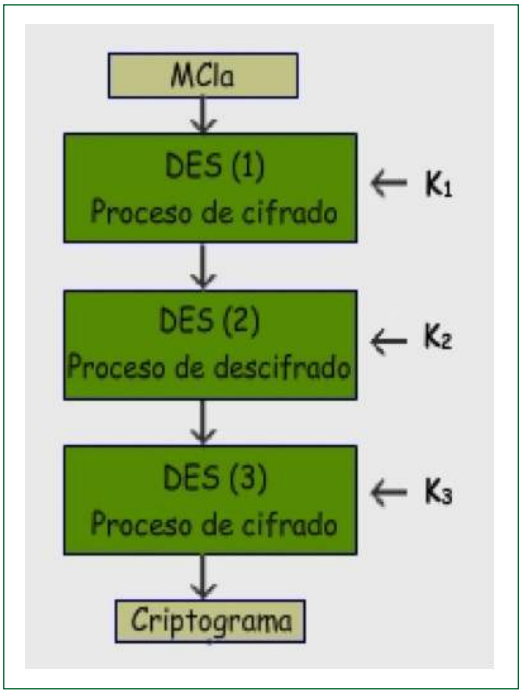


Imagen 2. Algoritmo TDES o Triple DES
Fuente: http://www.slideshare.net/gopalsakarkar/cryptography-and-encryptions

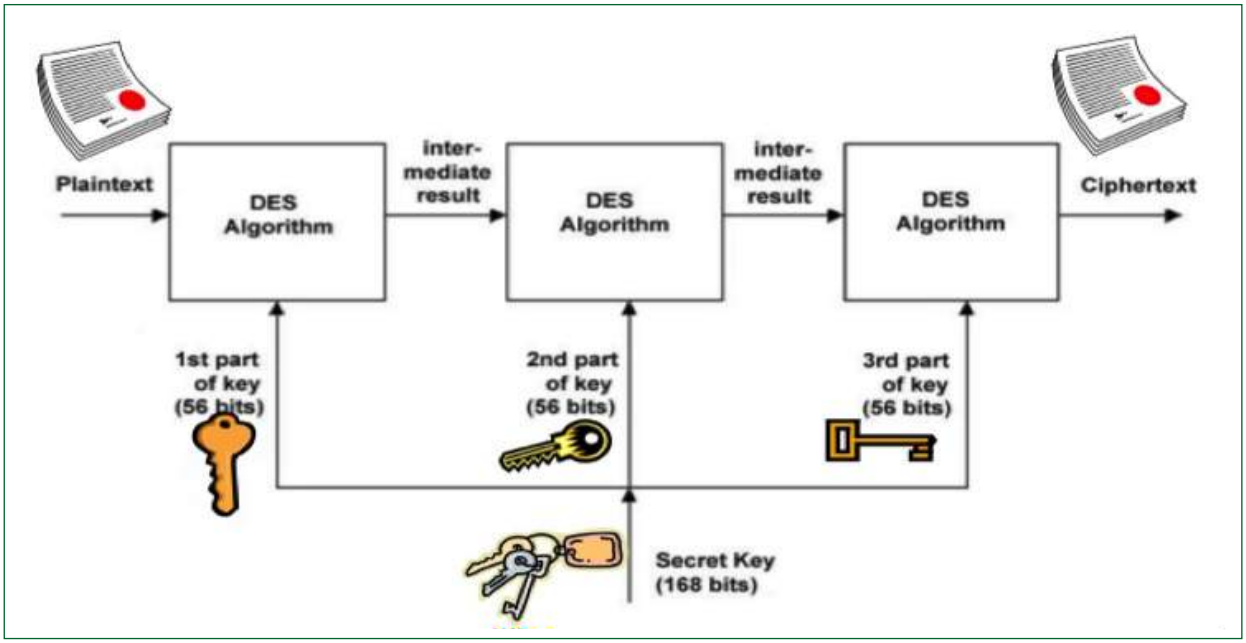


Imagen 3. Algoritmo Triple DES.
Fuente: http://www.slideshare.net/gopalsakarkar/cryptography-and-encryptions

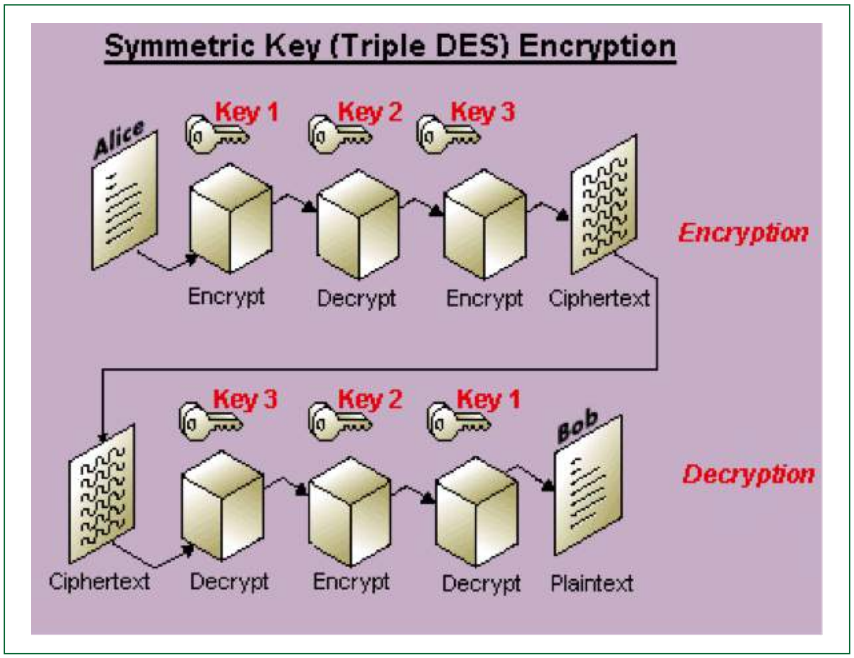


Imagen 4. Algoritmo Triple DES, Encripción y desencripción
Fuente: http://www.slideshare.net/gopalsakarkar/cryptography-and-encryptions

Algoritmo AES

Este algoritmo fue creado por Vincent Rijmen y Joan Daemen de la Katholieke Universiteit Leuven en Bélgica al concurso que organizó el NIST, Instituto Nacional de Normas Técnicas, con el objeto de encontrar el mejor algoritmo simétrico de cifrado. Luego de ganar el premio como el mejor algoritmo de cifrado, fue propuesto como un estándar en 2002 y nombrado como el AES- Advanced Encryption Standard, convirtiéndose en el más utilizado actualmente.

A partir de 2003, el gobierno americano decidió que el algoritmo AES era lo suficientemente seguro para usarlo en la protección nacional de la información.

AES, es considerado como un algoritmo de cifrado por bloques, cada uno de 128 bits, de tal forma que los datos que se van a encriptar se parten en segmentos de 16 byts, o sea 128 bits, o que equivale a un bloque.

Características importantes del algoritmo AES, es que utiliza la misma clave para encriptar y desencriptar, al igual que la clave puede formarse con 128, 192 o 256 bits, estas son especificaciones del estándar. Estas implementaciones reciben los nombres de: AES-128, AES-192 y AES-256.

Para generar una clave, se parte de una clave inicial de 16 bytes -128 bits, y que se representa como una matriz de 4 x 4 bytes, a partir de aquí, se generan diez claves más la clave inicial. A estas claves se les llama subclaves.

Al cifrar una clave se aplica a cada uno de los estados, unas operaciones o llamadas rondas (ver imagen 5), de las que usa once;

cada una de ellas, utiliza una subclave distinta. Estas subrondas a su vez, se clasifican como: la ronda incial, que se aplica a la subclave inicial; luego nuevo rondas estándar, a las cuales se les aplica las nueve claves siguientes, una en cada ronda. Por último, una ronda final, que se aplica a la última subclave.

Dentro de cada ronda se realizan cuatro operaciones como son: SubBytes, ShiftRows, MixColumns, AddRoundKey, que no son otra cosa que matrices.

	Longitud de clave Nk (palabras de 32 bits)	Longitud de bloque de datos Nb (palabras de 32 bits)	Número de rondas
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Imagen 5. Rondas del Algoritmo AES
Fuente: http://goo.gl/tEqEfj

El esquema de cifrado del algoritmo AES, se muestra en la siguiente imagen:

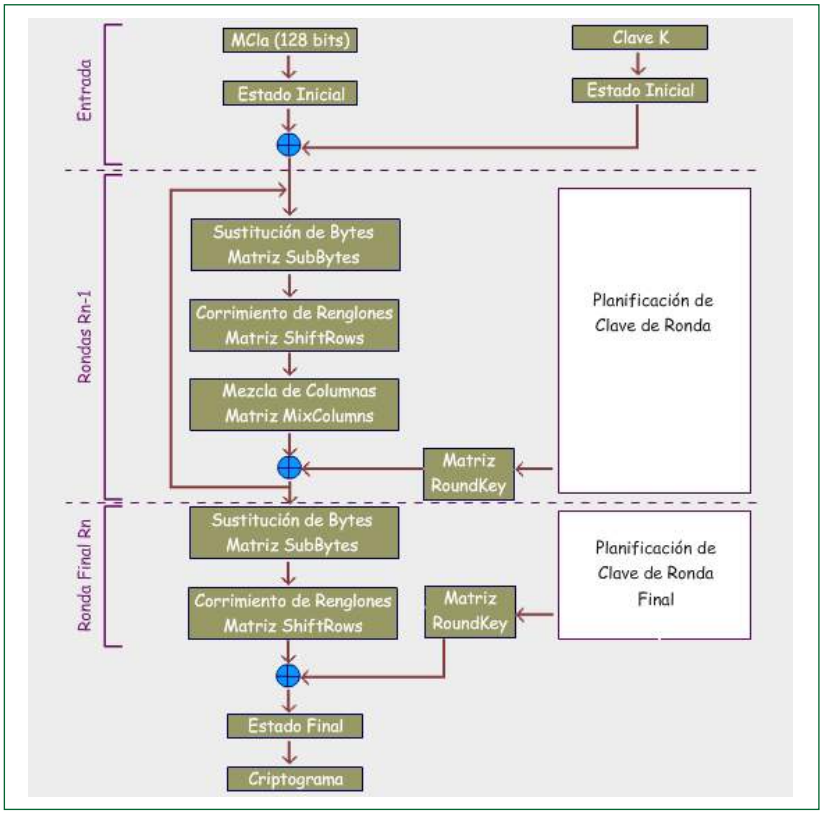


Imagen 6. Algoritmo de Cifrado AES
Fuente: http://goo.gl/tEqEfj

Dependiendo si la clave utiliza 128, 192 o 256 bits, el Algoritmo AES realizan las rondas correspondientes, así como se observa en la imagen 6. Como se mencionó, cada ronda se componen de cinco matrices, de las cuales, la primera es la de entrada o inicio y las otras 4 son transformaciones o funciones.

Para el descifrado, el algoritmo AES utiliza las matrices de Cripto, InvShiftRows (ISR), InvSubBytes (ISB), RoundKey (RK) y Matriz ISB, para lograrlo, también utilizan rondas para la desencripción.

Algoritmo IDEA

IDEA - International Data Encryption Algorithm, creado por Xuejia Lai y James L. Massey en 1991; es un algoritmo usado para encriptar textos, cuyo bloques son de 64 bits, para ello requiere de una llave o clave de 128 bits.

Para la encripción de datos utiliza 8 transformaciones o rondas, lo mismo que una transformación de salida llamada media ronda. Tanto el cifrado como el descifrado es el mismo, lo único que cambia son las llaves de la ronda, utiliza para ello 52 de 16 bits cada una. Su funcionamiento se basa en sumas y multiplicaciones a aplicación del XOR, bit a bit, de las operaciones de la suma se realizan 2 a la 16, de la multiplicación se realizan 2 a la 16 más 1, lo que genera un número primo.

Para generar las claves o llaves, parte la llave original en ocho partes de 16 bits cada una. Las primeras seis partes son las llaves de la k1 a la k6 de izquierda a derecha, es lo que utiliza IDEA en la primera ronda. Luego las llaves o claves k7 y k8 son las últimas partes. A partir de allí, las 44 llaves se generan con un corrimiento circular a la izquierda de 25 bits sobre la llave original, se extraen las llaves correspondientes y nuevamente se hace un corrimiento, de esta forma hasta completar las 52 claves en los 6 pasos.

Algoritmos Blowfish y Twofish

Algoritmo Blowfish. Creado por Bruce Shneier, en 1993. Es un algoritmo de bloques simétrico, sencillo de implementar. La longitud de su clave es variable y puede ir de 32 hasta 448 bits, es decir, 14 bloques de 32 bits, lo que le permite un proceso de alta seguridad y alta velocidad. Es considerado más rápido que el algoritmo DES y el IDEA.

Su trabajo lo realiza por bloques, quiere decir que toma 64 bits de texto plano y entrega 64 bits de texto cifrado. Genera 18 subclaves de 32 bits y toma 4 subclaves de sustitución de 8 X 32.

Las claves que genera el algoritmo las almacena en arreglos de permutación, también llamados P-Box. Para el ciframiento utiliza dos tipos de operaciones, la suma o adición y la operación XOR.

El número de rondas que utiliza este algoritmo es de 16.

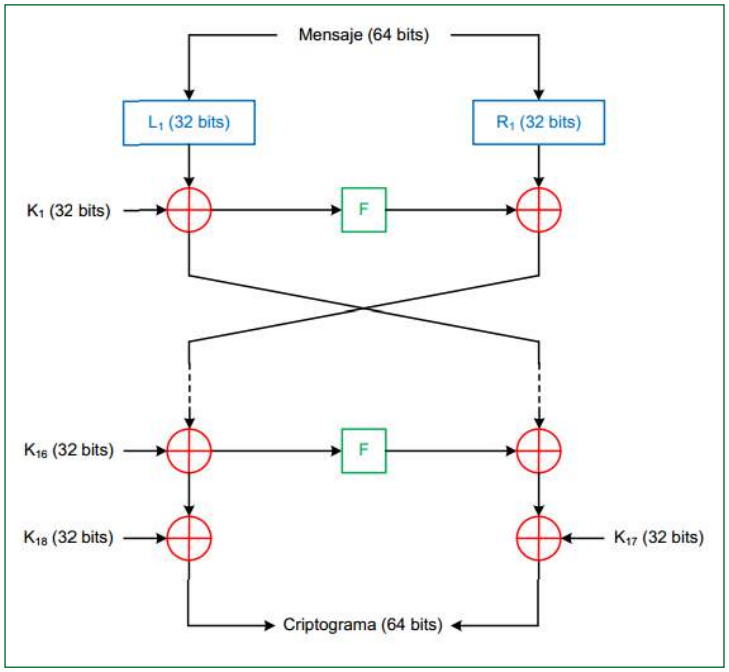


Imagen 7. Algoritmo Blowfish
Fuente: http://lumbreras-criptografia.blogspot.com.co/2013/07/cifrados-por-bloque-blowfish.html

Explicación de la imagen 7, cómo cifra el algoritmo Blowfish: Tomado de: http://p.jdiez.me/index.php/blowfish/

- Un bloque de 64 bits de divide en dos de 32 bits.
- Luego se aplica la operación binaria XOR a los 32 bits altos con la primera de las sub-claves.
- A continuación se aplica la función F al resultado de la operación xL XOR Pi (el paso anterior).
- Se aplica la operación binaria XOR a los 32 bits bajos (xR en el diagrama) con el resultado de la operación F(xL).
- Se intercambian los bits altos con los bajos.
- Se repite el proceso 16 veces.
- Se deshace el último intercambio.
- Se aplica la operación binaria XOR a los bits altos resultantes con la semi-clave 18.
- Se aplica la operación binaria XOR a los bits bajos resultantes con la semi-clave 17.
- Se recomponen los dos grupos de 32 bits en uno de 64 bits, ya cifrado.

Algoritmo Twofish

Creado por Bruce Schneier, John Kelsey, Doug Withing, David Wagner, Chris Hall y Niels Ferguson en 1998. Este algoritmo es de cifrado simétrico, que tiene un tamaño de 128 bits, sus claves son de hasta 256 bits.

El algoritmo consta de 16 rondas y de varios módulos operativos cuando realiza la encripción y desencripción. También cifra los datos a 285 ciclos de reloj por bloque y los

datos a 26500 ciclos de reloj por bloque. Tiene como principal propósito, cumplir con los criterios de diseño del NIST para AES.

Se dice que es un algoritmo simétrico de bloques de 128 bits, con longitudes de clave de 128 bits, 192 bits o de 256 bits, como se puede ver en la imagen 8. Es un algoritmo flexible, que se puede implementar eficientemente en distintas aplicaciones criptográficas.

Como principal característica del algoritmo Twofish, es que utiliza S-boxes con llaves independientes, también usa una clave de horario compleja.

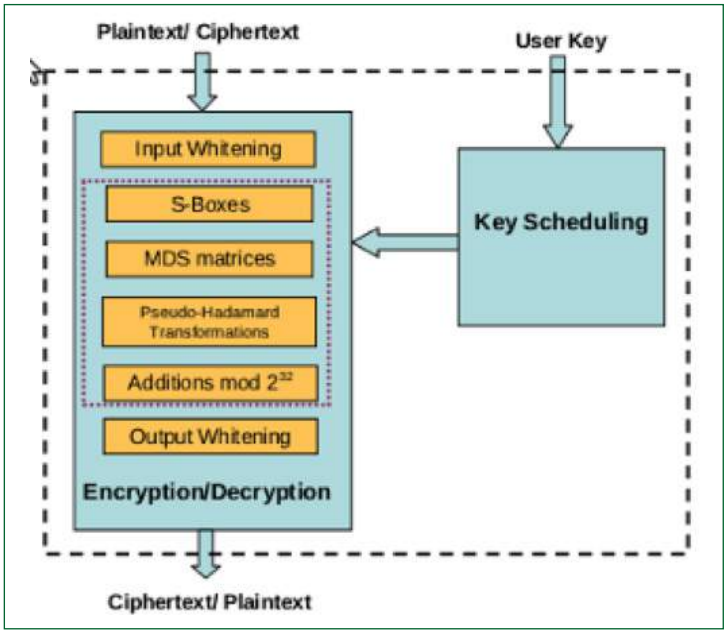


Imagen 8. Algoritmo Twofish
Fuente: http://comunidad.dragonjar.org/f156/lista-de-algoritmos-de-encriptacion-8807/index2.html

Algoritmo RC4

RC4 - Rivest Cipher 4, desarrollado por Ronald Rivest en 1987. Es un algoritmo de cifrado en flujo y es usado por los protocolos para la comunicación como el SSL y WEP. Es un algoritmo simple y rápido, pero débil, pues sus primeros bytes son predecibles.

El cifrado y descifrado en este algoritmo son iguales, debido a la simetría que se presenta a partir de la operación binaria XOR. Este algoritmo genera un flujo pseudoaleatorio de bits, teniendo como premisa la clave original, el flujo de bits se combina con el mensaje claro a través de la operación lógica XOR.

Una desventaja del algoritmo RC4, es que se puede recuperar la clave secreta, cuando se deriva de una concatenación con un vector inicial púbico que se usa como WEP, un gran número de mensajes y vectores iniciales la clave quede expuesta.

Rc4, además de ser rápido y ser usado por aplicaciones comerciales serias, es un generador de números pseudoaleatorios que se inicializa desde una clave secreta superior a los 256 bytes, utilizada por SSL.

RC4, se basa en la construcción de dos arreglos, uno denominado S-box y el otro arreglo K. El vector S-box se llena con valores secuenciales de 0 a 255, mientras que el vector K se llena con el valor de la semilla hasta ser completamente llenado y luego, se mezclan los dos vectores. A continuación, el vector S, se intercambia usando el valor de la semilla. Como resultado, se tiene la Clave para el algoritmo.

Algoritmos de funciones de Hash

Las funciones Hash tienen como entrada un conjunto de elementos, que generalmente son cadenas, las cuales son mapeadas o convertidas en rangos de salida finitos, que también son cadenas de longitud fija.

La fórmula que generalmente usan las funciones Hash es: H: U -> M x -> h(x).

La función Hash actúa como una proyección del conjunto U sobre el conjunto M, el cual puede ser un conjunto de números enteros. Por lo general el conjunto U tiene un número elevado de elementos y el conjunto U es un el conjunto U está formado por cadenas con pocos símbolos.

Según Sergio de Luz (2010), los requisitos que deben cumplir las funciones hash:

- Imposibilidad de obtener el texto original a partir de la huella digital.
- Imposibilidad de encontrar un conjunto de datos diferentes que tengan la misma huella digital (aunque como hemos visto anteriormente es posible que este requisito no se cumpla).
- Poder transformar un texto de longitud variable en una huella de tamaño fijo (como el SHA-1 que es de 160bits).
- Facilidad de empleo e implementación.

Algunos algoritmos de funciones Hash son:

El algoritmo MD5 es un algoritmo de función Hash de 128 bits. Toma unos tamaños determinados a la entrada y como salida, saca los mismos 128 bits, pero números de 64 pasos.

El algoritmo SHA-1, por su parte admite bloques de 160 bits. la función Hash de compresión es más compleja que la del MD5, por ello, es un algoritmo más lento, ya que tiene ochenta pasos y tiene mayor longitud, esto lo hace un algoritmos más seguro y robusto. De este tipo de algoritmos se conocen: SHA-224, SHA-256, SHA-384, y SHA-512. Por obvias razones, el más seguro es el SHA-512. Pues el que tiene mayor número de bits de salida, utiliza ochenta pasos como el SHA-1, pero su diferencia radica en que el tamaño de salida 512 y el SHA-1 es 160. Los tamaños del bloque, de palabra e internos el doble del SHA-1. Como desventaja tienen los algoritmos hash, que entre más seguros, son más lentos.

El algoritmo de Residuo de la división, divide el valor de la clave de entrada entre un número apropiado y utiliza el residuo de la división como dirección relativa para el registro.

El algoritmo Media del cuadrado, es otro algoritmo Hash, el cual toma la clave de en-

trada, la eleva al cuadrado, a continuación se extraen algunos dígitos específicos de la mitad del resultado, con el fin de construir la dirección relativa. Al querer una dirección de N dígitos, se truncan en ambos extremos de la clave elevada al cuadrado y se toman los N dígitos que se encuentren en la mitad. Para la clave, se extraen las mismas posiciones de N dígitos.

Al algoritmo Pliegue, utiliza la llave y la eleva al cuadrado, luego, unos dígitos específicos se sacan de la mitad del resultado, con el fin de construir la dirección relativa, cuando se quiere una dirección de N dígitos, se truncan los dos extremos de la clave elevada al cuadrado y se toman los N dígitos del medio. Para clave se extraen las mismas posiciones de N dígitos.

Imagen 9. Algoritmo Twofish
Fuente: http://www.redeszone.net/2010/11/09/criptografia-algoritmos-de-autenticacion-hash/

Algoritmo MD5

Desarrollado por Ronald Rivest, 1995. Este algoritmo es uno de los más utilizados en el momento. Se basa en dos algoritmos como son el MD2 y el MD4, el cual sale de una mejora que se le hiciera al MD2. Estos protocolos producen un número de 128 bits que sale de un texto de cualquier longitud.

Realmente el algoritmo MD5 reemplazó al MD4, que no ha podido superar el rendimiento del segundo, pero afortunadamente no se ha publicado aún ningún compromiso sobre la integridad y funcionamiento.

Como se nota en la imagen 10, el algoritmo MD5 inicia con que al mensaje original lo rellena hasta completar 448 mod 512, esto quiere decir que la longitud del mensaje es de 64 bits menos que un entero múltiplo de 512. La parte del relleno se conforma de un bit en 1 seguido por todos los bits necesarios, pero en 0 y la longitud original del mensaje se almacena en los últimos 64 bits del relleno.

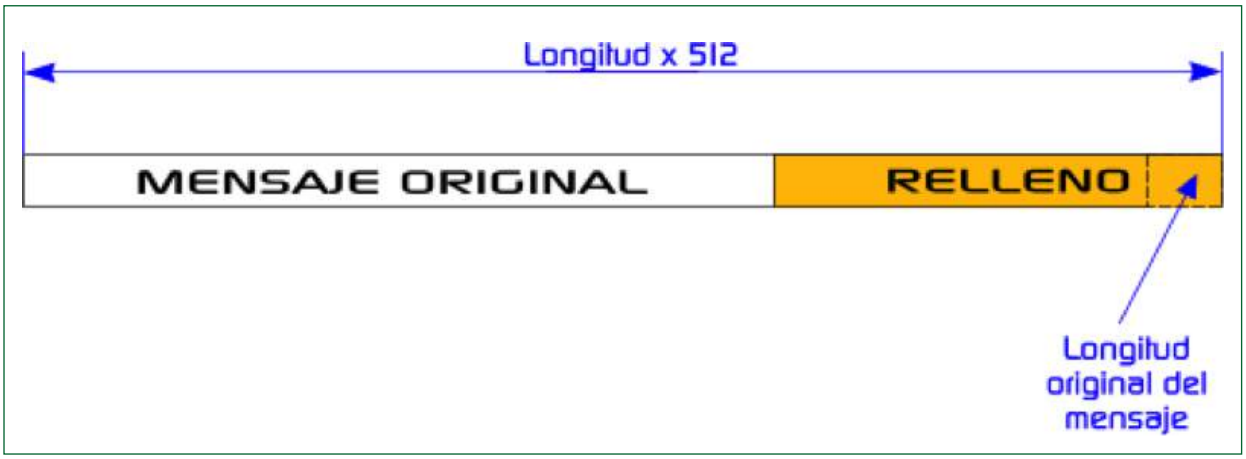


Imagen 10. Algoritmo MD5
Fuente: http://neo.lcc.uma.es/evirtual/cdd/graficos/md5.gif

De otra parte, se tiene un valor fijo de 128 bits en un buffer, el cual se puede observar como cuatro registros de 32 bits (A, B, C, D) y se almacena con los valores siguientes: A=67452301; B=EFCDAB89; C=98BADCFE; D=10325476, que se escriben en Hexadecimal.

Luego de efectuar varias rondas de procesamiento, el algoritmo toma bloques de 512 bits de entrada que son mezclados con los 128 bits del buffer; de esta forma, el algoritmo repite el proceso, hasta que todos los bloques de entrada se consuman y el resultado será un valor en el buffer que no es otra cosa que el hash del mensaje.

Algoritmo SHA

SHA - Secure Hash Algoritm, fue desarrollado por NSA, genera números has de 160 bits que se generan a partir de:

La longitud máxima de entrada de los mensajes es de 2 a los 64 bits, los cuales son procesados en bloques de 512 bits y el resultado producido es de 160 bits.

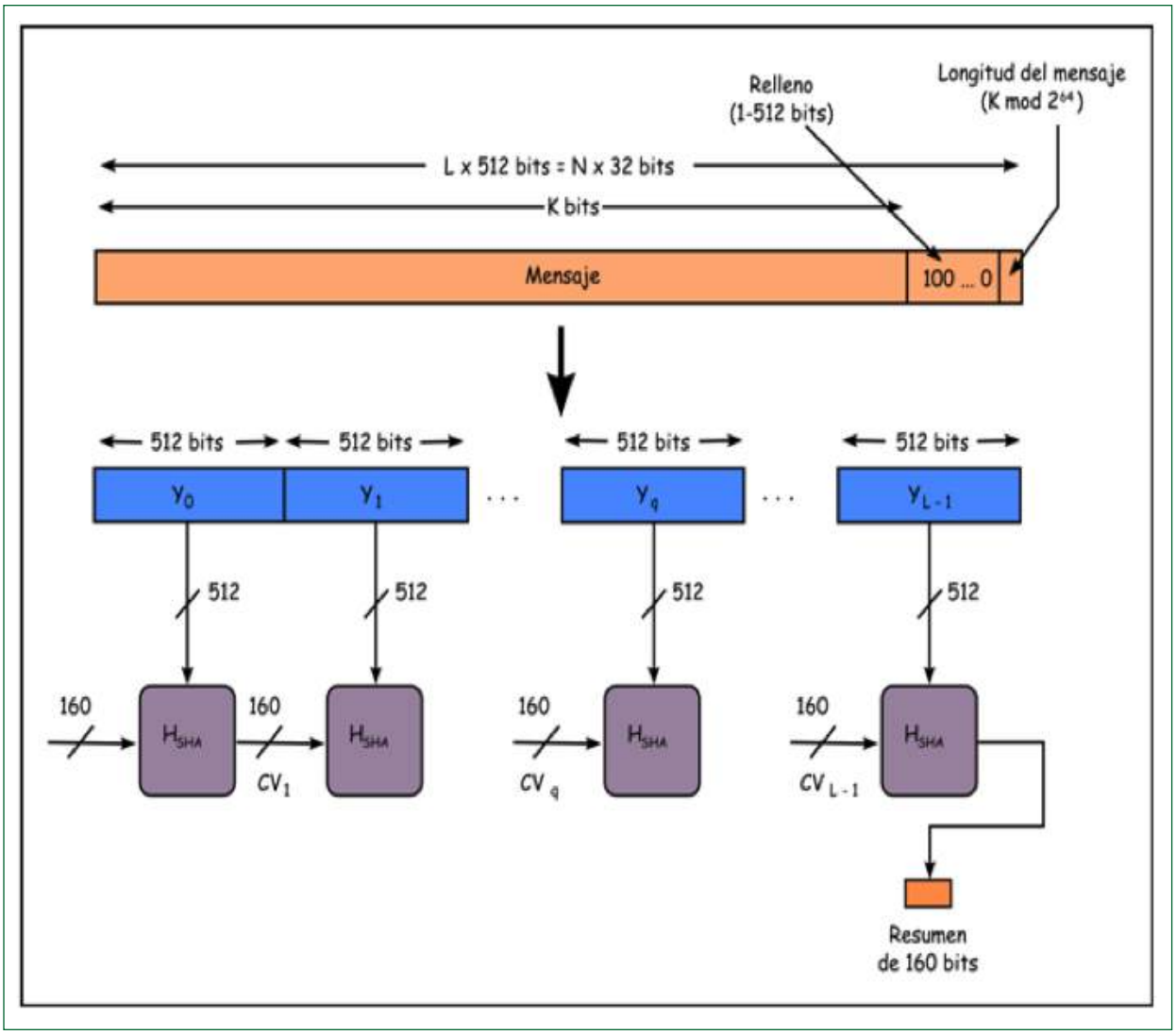


Imagen 11. Algoritmo SHA-1
Fuente: http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/5-criptografia-asimetrica-o-de-clave-publica/55-funciones-hash/552-sha-secure-hash-algorithm

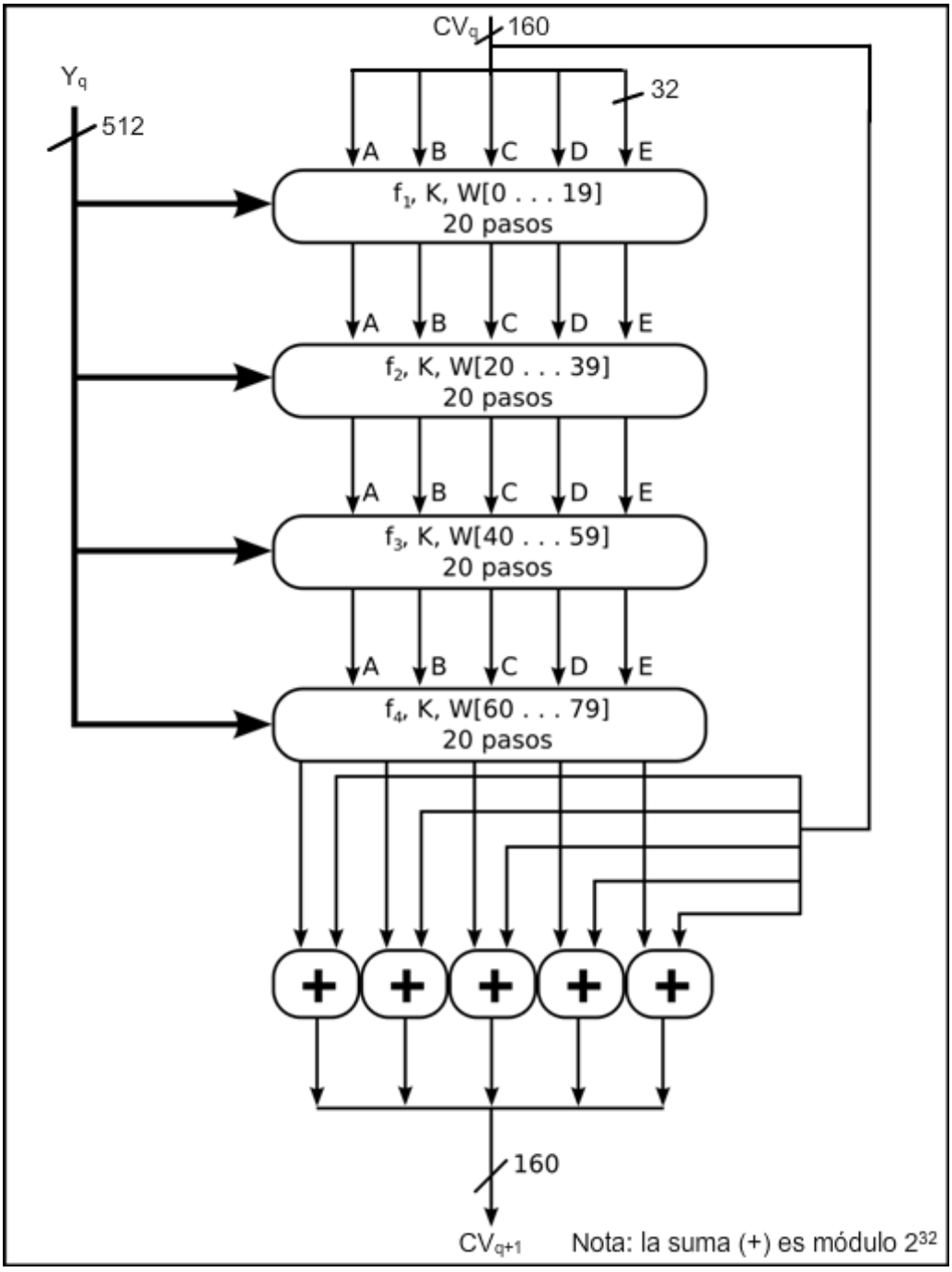


Imagen 12. Procesamiento SHA-1 de un único bloque de 512 bits
Fuente: http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/5-criptografia-asimetrica-o-de-clave-publica/55-funciones-hash/552-sha-secure-hash-algorithm

Según la Universidad Nacional autónoma de México. (2015). El algoritmo es procesado en cinco pasos, a saber:

1. Se incorporan bits de relleno al mensaje de entrada de tal modo que cumpla:

   longitud = 448 mod 512.

   El relleno consiste en un uno seguido de los ceros que sean necesarios. Aunque el mensaje ya tenga la longitud deseada, se debe efectuar el relleno, por lo que el número de bits de dicho relleno está en el rango de 1 a 512 bits.

2. A la salida del paso 1, se le añade un bloque de 64 bits que represente la longitud del mensaje original antes de ser rellenado.

   Se inicializa la memoria temporal MD, la cual consta de 160 bits y su finalidad es almacenar los resultados intermedios y finales de la función de dispersión. La MD consta de 5 registros (A, B, C, D, E) de 32 bits cada uno, los valores con los que se inicializan son los siguientes (valores hexadecimales):

   A=67452301

   B=EFCDAB89

   C=98BADCFE

   D=10325476

   E= C3D2E1F0

3. Se procesa el mensaje por bloques de 512 bits, cada uno pasa por un módulo que consta de 4 rondas de procesamiento de 20 pasos cada una. Las rondas tienen una estructura similar, con la excepción de que cada una ocupa una función lógica primitiva diferente (f1, f2, f3 y f4). Esta parte del algoritmo se muestra en la imagen 12.

   La entrada a cada ronda consta del bloque de 512 bits que se esté procesando (Yq) y los 160 bits de la memoria MD, nótese que cada bloque de 512 bits actualizará el valor de la memoria temporal. Cada ronda también hace uso de la constante aditiva Kt, donde

   $0 \leq t \leq 79$ indica uno de los 80 pasos a lo largo de las cuatro rondas. Los valores para dicha constante se muestran en la tabla de la imagen 13.

Ronda	Número de paso	K_t (hexadecimal)
1	$0 \leq t \leq 19$	5A827999
2	$20 \leq t \leq 39$	6ED9EBA1
3	$40 \leq t \leq 59$	8F1BBCDC
4	$60 \leq t \leq 79$	CA62C1D6

Imagen 13. Valores de la constante aditiva Kt en SHA-1
Fuente: http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/5-criptografia-asimetrica-o-de-clave-publica/55-funciones-hash/552-sha-secure-hash-algorithm

4. Una vez que se procesan los L bloques de 512 bits, el resumen del mensaje son los 160 bits de salida del último bloque.

Códigos de autenticación de mensaje (MAC) y algoritmo HMAC

Los códigos de autenticación de mensaje – MAC, es un trozo de información que se usa para autenticar un mensaje. Estos valores MAC se calculan por medio de la función hash criptográfica con clave secreta K, la cual solo es conocida por el emisor y el receptor.

Las funciones MAC ideales son aquellas que mapean de forma aleatoria todos los posibles valores del mensaje de entrada M en salidas de tamaño de N bits. Estas funciones MAC se dice son seguras si resisten a la computación de nuevos valores Hash.

Existen distintas funciones MAC, como por ejemplo:

CBC-MAC: La idea detrás de este algoritmo es la de convertir un algoritmo de cifrado simétrico en una función MAC. El funcionamiento básico consiste en cifrar el mensaje usando un algoritmo en modo CBC y tirar todo el resultado de texto cifrado a excepción del último bloque.

HMAC: dado que una función MAC es un mapeo aleatorio, y que las funciones hash se comportan como tales, podemos explotar la idea de utilizar una función hash para implementar una función MAC. La opción más popular hoy en día es la de usar HMAC-SHA-256.

UMAC: las funciones UMAC parten de la premisa de que el atacante necesita interactuar con el sistema para comprobar si el resultado MAC que ha generado es válido o no. Es decir, no existe nada parecido a un ataque exhaustivo off-line contra las funciones MAC. Así, argumentan que se puede reducir el resultado a tan solo 64 bits. Sin embargo, no existe un estándar bien definido de funciones MAC como ocurre con las funciones hash, lo que tiene efectos contraproducentes a largo plazo desde el punto de vista de la implementación.

Algunos usos que se le dan a las MAC es para autenticación, es decir que tanto receptor como emisor solamente comparten la clave de la función MAC.

Algoritmo HMAC

HMAC, es un mecanismo para autenticar mensajes. Es un entorno de autenticación en entornos seguros SSL, por medio de la operación MAC en la que interviene funciones hash.

Los HMAC con clave son algoritmos que garantizan un mensaje íntegro. Un algoritmo HMAC tiene dos parámetros, el uno, es una entrada del mensaje y una clave secreta, que solo la conocerán el emisor y los posibles receptores. El emisor utiliza la función HMAC para generar un valor, es decir el código de autenticación de mensajes, formado por la compresión de la clave secreta y la entrada del mensaje. Este mensaje es enviado junto al código de autenticación de mensajes, el receptor calcula el código de autenticación de mensajes en el mensaje recibido con la misma función HMAC que usó el emisor, luego, el receptor hace una comparación con el resultado que calculó el código de autenticación de mensajes que ha recibido, de donde se pude deducir que coinciden o no los mensajes. En el momento que los valores coincidan, el mensaje es recibido correctamente, por lo que el receptor se percata que el emisor si es parte de su comunidad, porque comparten la clave.

Una función criptográfica HMAC, se dice que es fuerte criptográficamente, cuando su tamaño, calidad de clave, tamaño de la longitud del resultado del hash en bits es adecuado. Los algoritmos más comunes de tipo HMAC son:

MD5, ya explicado en este escrito, que recordando, utiliza 128 bits, con un mensaje de longitud variable, cuya clave secreta se

combina y procesa con algoritmos hash HMAC-MD5, cuyo resultado es un hash de 128 bits y el hash se adjunta al mensaje original y se envía al extremo remoto.

SHA: SHA-1 utiliza una clave de 160 bits. El mensaje de longitud variable y la clave secreta compartida de 160 bits se combinan y se procesan con el algoritmo de hash HMAC-SHA1. El resultado es un hash de 160 bits. El hash se adjunta al mensaje original y se envía al extremo remoto.

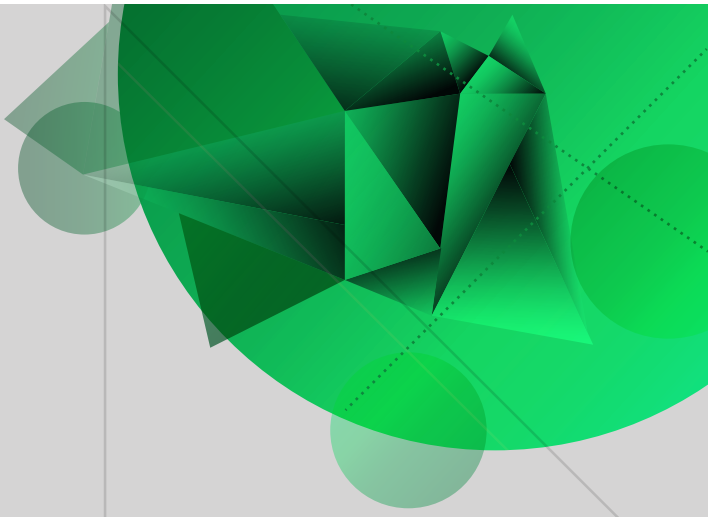
Ejercicio para realizar

Utilizando algunos de los lenguajes de programación que conoce, realice un algoritmo RC4, utilizando dos vectores, el uno denomínelo S y el otro K. en el vector S almacene valores secuenciales de 0 a 250 y en el vector K almacene las semillas producidas por números randómicos. Luego mezcle los dos vectores así: intercambie el valor de una semilla con un número del vector S, así obtendrá la clave para el algoritmo. Imprima los dos vectores

3

Unidad 3

Algoritmos de criptografía simétrica



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

El conocimiento siempre ha sido una de las facultades más preciadas del ser humano, por lo tanto, puede ser convertido en información y esta a su vez en elemento vital para cualquier empresa o ciudadano. De ahí que quiera ser preservado, guardado, secreto e inviolable, por lo que ha acudido a la representación simbólica y que se ha requerido simbolizar de diferentes formas con miras a que la información permanezca siempre en secreto. Para lograrlo, se han inventado los algoritmos criptográficos para transformar dicha información en secreto a través de claves, estas se conocen hoy en día como representaciones o criptogramas, los cuales son ininteligibles para quienes no conocen la clave o la forma cómo leer un mensaje cifrado, pero esto se ha logrado a través de la criptografía y específicamente por medio de los algoritmos que se han desarrollado para tal fin.

Es así que la criptografía cubre dos tipos, de una parte, los algoritmos criptográficos simétricos que se estudiaron en la semana anterior a esta y que se refieren a que se usa la misma clave para cifrar y descifrar información; pero de otra parte existen también los algoritmos asimétricos, los cuales se distinguen por utilizar dos claves diferentes, pero con la característica que si un mensaje es cifrado con una clave, podrá ser descifrada con la otra clave, estos algoritmos utilizan una clave pública y la otra privada.

En el transcurso de este escrito se explicará precisamente cómo es el manejo de los algoritmos asimétricos, cómo funcionan y qué usos se les da.

U3

Metodología

En este momento, los estudiantes tienen la fundamentación de lo que es la criptografía, sus tipos, el uso que ha tenido las matemáticas en este campo, han aprendido la terminología esperada para comprender el tema del ciframiento y lo que éste hace sobre la seguridad. Por ello, es pertinente que en este momento se conozca más a fondo los algoritmos asimétricos, sus usos y características. Para lograr este cometido, se hacen las siguientes recomendaciones:

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Participar en el foro propuesto.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

Algoritmos de criptografía asimétrica

Algoritmo RSA

Rivest, Shamir y Adleman - RSA, creadores del algoritmo en 1978. Es un algoritmo utilizado para realizar transmisión en canales inseguros.

Para generar las claves, este algoritmo utiliza dos números primos muy grandes, positivos y de longitud similar en bits; generalmente entre 100 y 300 dígitos. La clave pública se genera a partir de una multiplicación entre los dos números primos grandes encontrados, mientras que la clave privada se consigue a partir de haber encontrado un número e que no tenga múltiplos comunes con módulo Ø.

RSA trabaja con longitudes variables para las claves, pero nunca menos de 1024 bits. La seguridad de este algoritmo es alta debido a que utiliza lo que comúnmente se denomina función computacional, que consiste en realizar lo que en matemática se llama exponenciación modular antes de crear las claves, pero es muy complicado realizar la operación inversa, mucho menos extraer las raíces del módulo Ø, por cuanto para hacerlo, se requiere conocer el número e calculado.

El procedimiento básicamente es el siguiente:

Cuando se calcula las claves pública y privada, se hace la siguiente operación:

Se consiguen dos números primos distintos, que se llamarán p y q, luego se encuentra n, a través de la siguiente fórmula:

n = p *q, siendo n el módulo de las dos claves.

Luego se calcula la función de Euler así: Ø (n) = (p-1)(q-1)

Se calcula e, para ello se toma un número entero positivo menor que Ø (n), cumpliendo con la propiedad de ser primo relativo de Ø (n).

La e, será el exponente de la clave pública, teniendo en cuenta siempre que e > n/2 y e será la clave pública.

Para calcular la clave privada, se hace por aritmética modular, a la que se le aplica el denominado algoritmo de Euclides extendido (calcula el máximo común divisor-MCD).

Al calcular las claves pública y privada, se almacenan en lugar seguro y deberá protegerse amabas claves por medio de un algoritmo criptográfico simétrico, ya que una vez que se tienen las llaves pública y priva-

da, e y d, respectivamente, se inicia con el cifrado así:
c = me (mod (n)), siendo m el mensaje original y c el mensaje cifrado, haciéndose este ciframiento con la llave pública.

Para descifrar el mensaje se procede así:
m = cd (mod (n)), siendo m el mensaje original pero visto numéricamente y c es el mensaje cifrado.

Este mensaje se descifra con la llave privada.

Es un algoritmo relativamente rápido y muy utilizado como sistema de llave pública, incluye la firma digital y empleado para encriptar y hacer envío de la clave simétrica que será usada luego al realizar comunicación cifrada ver imagen 1.

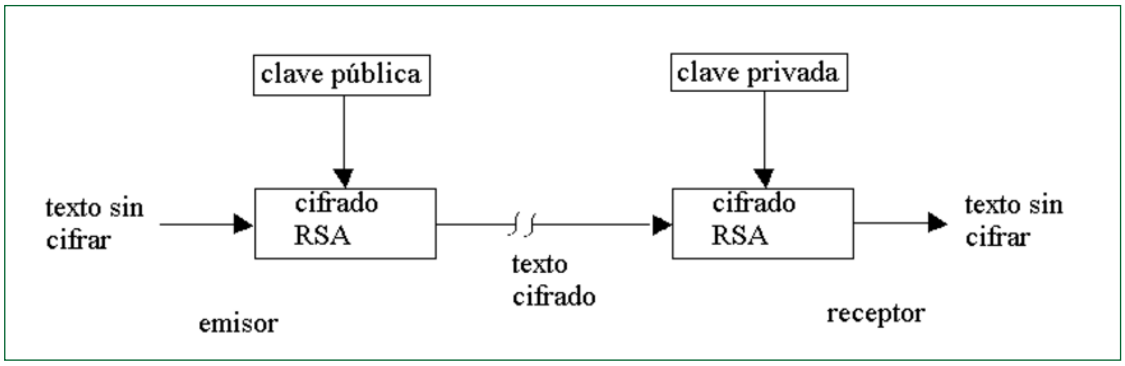


Imagen 1. Algoritmo RSA
Fuente: http://html.rincondelvago.com/redes-y-autenticacion-de-firmas-digitales.html

Una importante característica de RSA, es que un mensaje que ha sido cifrado con la clave pública, se descifra con la llave privada, aunque también, si se ha cifrado con la clave privada, se puede descifrar con la llave pública.

En los algoritmos asimétricos, cuando se cifra un mensaje con clave pública o privada, se afirma que lo que cifra una, sólo la descifrará la otra y viceversa.

Existen tres pasos básicos para realizar la criptografía asimétrica con el algoritmo RSA, a) se generan las claves pública y privada, b) se cifra el mensaje utilizando la llave pública, c) se descifra el mensaje, para ello se utiliza la llave privada.

Ejemplo, para cifrar el mensaje "muc", utilizando el algoritmo RSA:

Primer paso: seleccionar dos números primos grandes.

p: 83

q: 89

Segundo paso: se calcula n.

n = p * q = 83 * 89 = 7387

Tercer paso: calcular Ø (n).

Ø (n)= (p - 1)(q - 1) = (83 - 1)(89 - 1)= 7216

Cuarto paso: se selecciona e como un número relativo de Ø (n).

e = 5009

Quinto paso: se calcula d, utilizando un algoritmo que calcule el mcd entre e y ø (n), luego se aplica módulo ø (n).

d = 2753

Sexto paso: se cifra el mensaje, convirtiendo el texto "MUC" a numérico con su ASCII equivalente.

MUC = M---77 U---85 C---67

Se aplica la fórmula: me (mod(n))

c1 = 77 5009 (mod(7387)) = 6663

c2 = 85 5009 (mod(7387)) = 5440

c3 = 67 5009 (mod(7387)) = 7128

Séptimo paso: se descifra el mensaje

c1= 6663

c2= 5440

c3= 7128

Se aplica la siguiente fórmula para descifrar:

m = cd (mod(n))

m1 = 6663 2753 (mod(7387)) = 77

m2 = 5440 2753 (mod(7387)) = 85

m3 = 7128 2753 (mod(7387)) = 67

Algoritmo ElGamal

Este algoritmo fue creado por Taher ElGamal en 1984, se basó en el algoritmo de Diffie-Hellman, el funcionamiento de ElGamal, es a través del logaritmo discreto, que se basa en la siguiente expresión: $x = \log_g(y) \iff g^x = y$

Son dos los usos que se le dan a este algoritmo, para generar firmas digitales y para cifrar y descifrar información.

ElGamal tiene tres componentes básicos: a) Un generador de claves, b) El algoritmo de cifrado y c) el algoritmo de descifrado.

Ejemplo (tomado de https://es.wikipedia.org/wiki/Cifrado_ElGamal):

Se eligen los siguientes valores:

$p = 17$ (primo elegido al azar)

$g = 3$ (generador)

$a = 6$ (llave privada elegida al azar)

$K = g^a \pmod p = 3^6 \pmod{17} = 15$ (llave pública)

La llave pública será $(17, 3, 15)$ y la privada (6).

El texto claro es $m = 9$. Escoge un $b = 5$ aleatorio:

$y_1 = g^b \pmod p = 3^5 \pmod{17} = 5$

$y_2 = K^b m \pmod p = 15^5 \cdot 9 \pmod{17} = 1$.

El texto cifrado $C_b(m, b)$ está compuesto por la tupla $(y_1 = 5, y_2 = 1)$.

El texto cifrado puede ser descifrado por utilizando la llave privada $(a = 6)$.

Utilizando el Teorema de Fermat:

$m = y_1^{p-1-a} y_2 \pmod p = 5^{10} \cdot 1 \pmod{17} = 9$

Algoritmo DSA

Digital Signature Algorithm – DSA, desarrollado en 1991 por el National Institute of Standards and Technology – NIST, es un algoritmo de firma digital, no para cifrar.

DSA tiene tres fases o etapas para que funcione: a) generación de claves, b) firma y c) verificación. Los ítems a) y b) los ejecuta el emisor, mientras que c) la realiza el receptor (ver imagen 2).

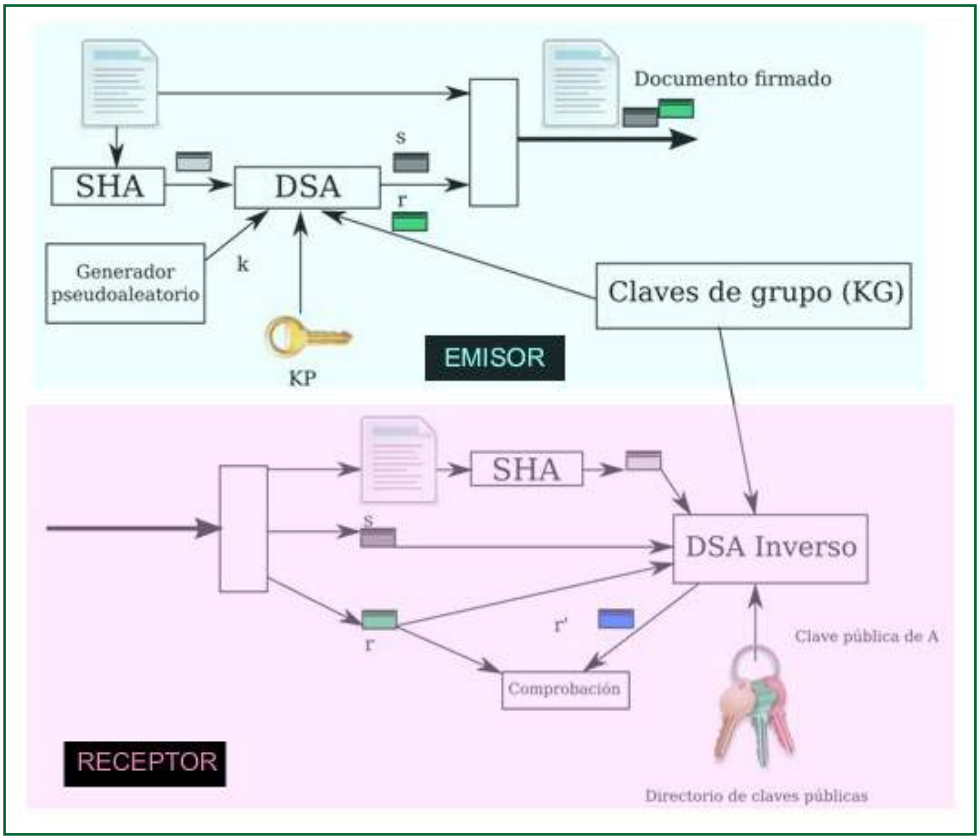


Imagen 2. Algoritmo DSA
Fuente: http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/5-criptografia-asimetrica-o-de-clave-publica/56-firmas-digitales/562-dsa-digital-signature-algorithm

Según Universidad Nacional Autónoma de México. (2013), los parámetros que usa el algoritmo DSA son:

1. Para generación de claves: los datos son públicos excepto x
   - p: número primo de 512 bits de longitud como mínimo.
   - q: número primo de 160 bits de longitud.
   - g: parámetro utilizado para calcular la clave pública.
   - x: clave privada del remitente.
   - y: clave pública del remitente.
2. En la firma
   - k: Número pseudoaleatorio único para cada firma.
   - s: Valor que corresponde a la firma.
   - r: Valor de comprobación de la firma.
3. Para la verificación
   - w: Valor que se requiere en el descifrado de la firma.
   - u1: Dato relativo al valor del hash del mensaje en claro.
   - u2: Dato relativo a la Integridad de la firma.
   - v: Valor de comprobación y verificación de firma.

Algoritmo Diffie-Hellman

Creado por Whitfield Diffie y Martin Hellman. Este algoritmo hace que dos personas que se quieren comunicar por un canal inseguro se pongan de acuerdo en un valor numérico, sin que una tercera persona tenga acceso pleno a la conversación.

Diffie-Hellman fue el primer algoritmo de clave pública. Diffie-Hellman es utilizado para distribuir o intercambiar llaves, pero no para cifrar o descifrar mensajes. Es un algoritmo seguro, debido a la complejidad en el cálculo de los logaritmos discretos en un campo finito

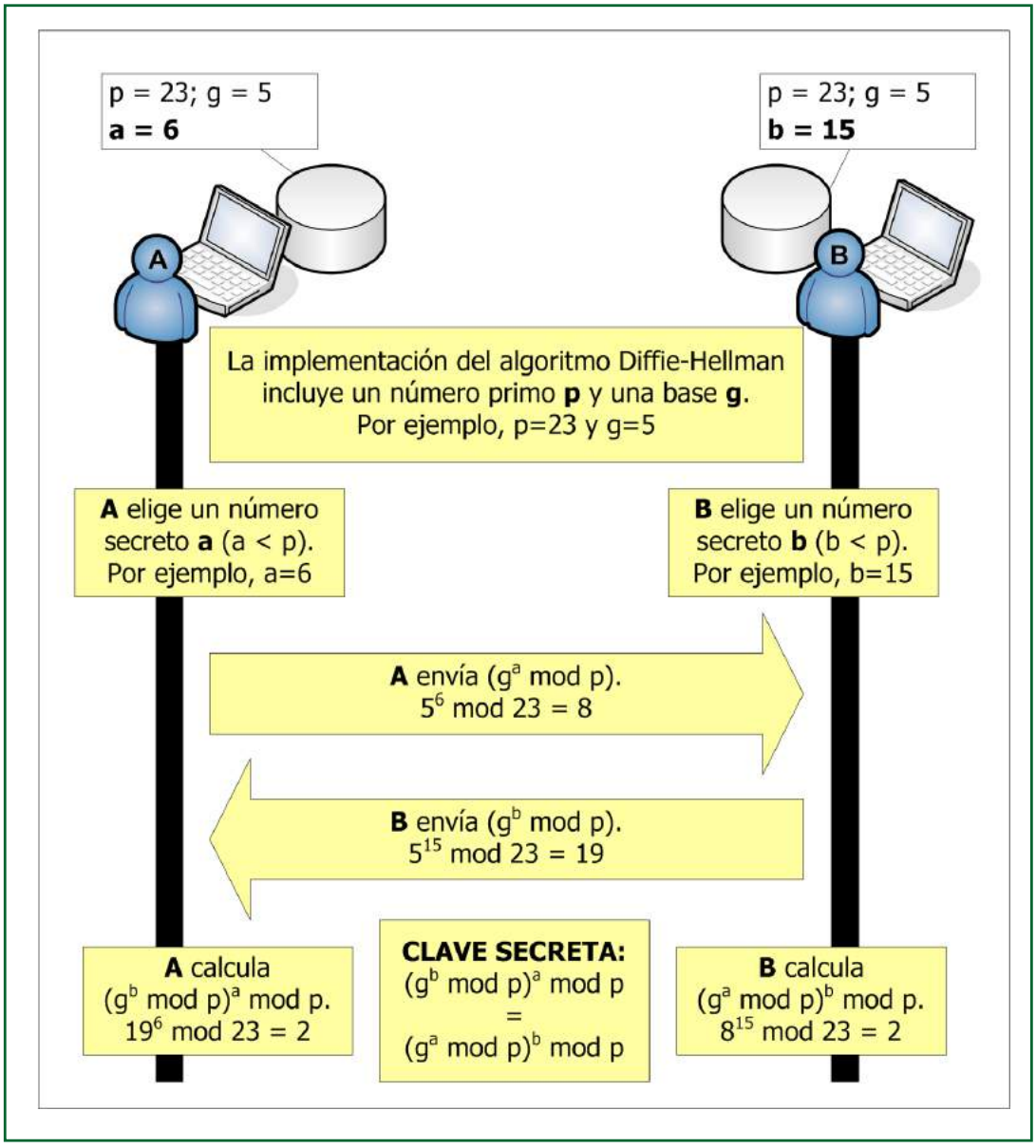


Imagen 3. Funcionamiento del protocolo Diffie-Hellman
Fuente: http://www.javiercampos.es/blog/2011/07/22/el-algoritmo-de-diffie-hellman/

En la imagen anterior, se muestra cómo el emisor selecciona dos números A y B; el receptor también escoge dos números llamados R, S. Estos individuos pueden comunicarse por canales denominados no seguros, que pueden ser como el correo electrónico o cualquier otro medio. Ambos, receptor y emisor hacen que el otro conozca los números A y R.

Enseguida, el emisor calcula un nuevo número Z utilizando la siguiente fórmula: Z = (R ^ B) % A.

También el receptor calcula su número T, usando la siguiente fórmula:

T = (R ^ S) % A.

Tanto emisor como receptor hacen públicos sus números Z y T.

Nuevamente el emisor calcula un número W, utilizando la siguiente fórmula:

W = (T ^ B) % A.

Y el Receptor calcula su número V, con la fórmula: Y = (Z ^ S) % A.

Tanto el número W como V, son los mismos, por lo que emisor y receptor comparten un secreto, es decir, sus claves que las pueden utilizar para cifrar sus mensajes. Los números W y V, han de ser muy grandes y sobre todo, números primos.

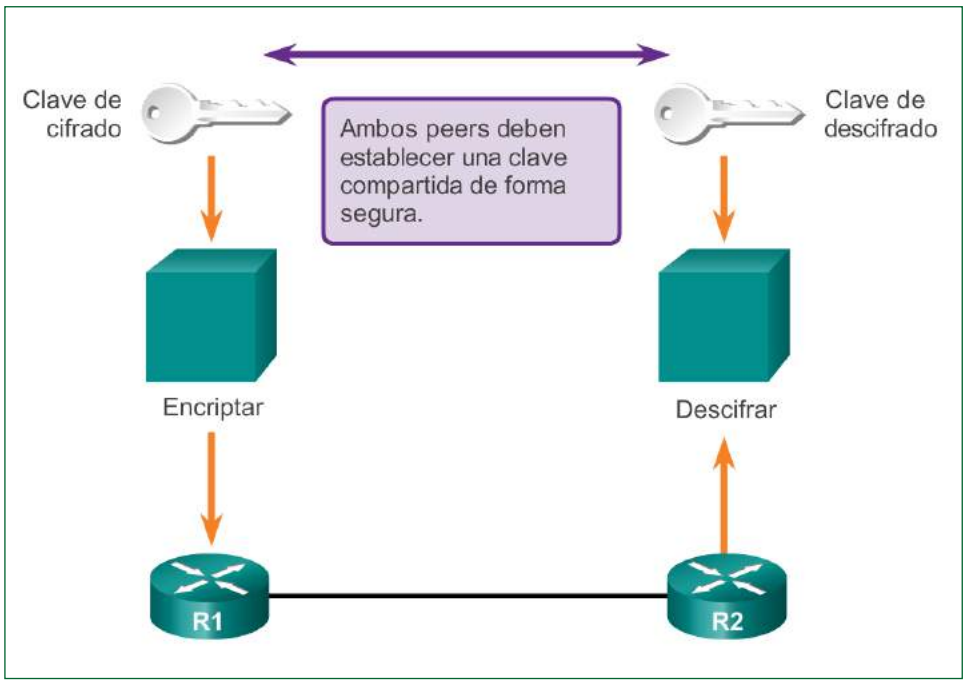


Imagen 4. Intercambio de claves de Diffie-Hellman
Fuente: http://www.itesa.edu.mx/netacad/networks/course/module7/7.3.2.3/7.3.2.3.html

Un sencillo ejemplo del algoritmo de Diffie-Hellman se aprecia en la imagen 4, en donde se aprecia que el método se utiliza para intercambiar llaves que cifran datos con seguridad, es así que tanto el emisor como el receptor, utilizando el algoritmo DH, pueden establecer una clave secreta compartida, la cual pueden usar para cifrar, utilizando también los algoritmos de hash.

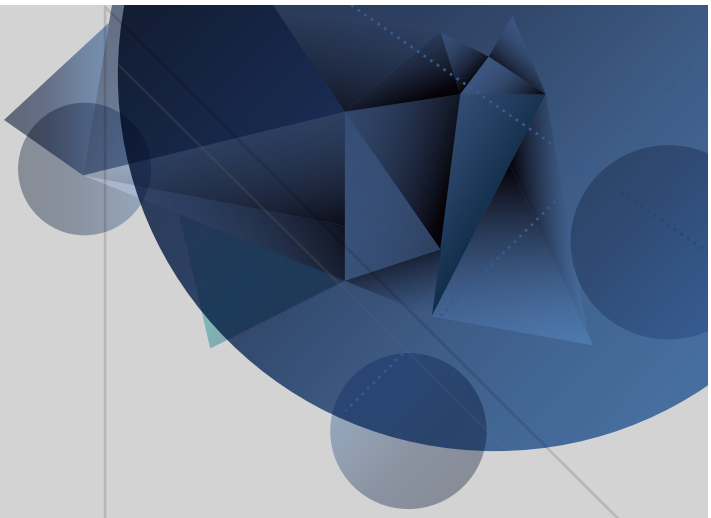
Ejercicio para realizar

Utilizando las fórmulas de los pasos descritos en el algoritmo de Diffie-Hellman, halle las claves que un emisor y un receptor pueden compartir para cifrar sus mensajes.

4

Unidad 4

Criptografía en la capa de aplicación



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

Son dos los temas que en este escrito nos ocupan, de una parte, la capa de Aplicación del modelo TCP/IP y de otra, los protocolos utilizados como medida de seguridad de dicha capa.

En cuanto a la capa de aplicación, esta integra tres de los niveles utilizados en el modelo OSI como son: Aplicación, Presentación y Sesión. En esta capa de aplicación del modelo TCP/IP, se utilizan todas las aplicaciones o programas que pueden comunicarse con otros a través de la red. Esta capa ofrece los servicios para trabajar con las aplicaciones de usuario.

Entre otros protocolos utilizados en esta capa se encuentran:

FTP: File Transfer Protocol (protocolo de transferencia de archivos).

HTTP: Hypertext Transfer Protocol (protocolo de transferencia de hipertexto).

SMTP: Simple Mail Transfer Protocol (protocolo de transferencia de correo simple).

DNS: Domain Name System (sistema de nombres de dominio).

TFTP: Trivial File Transfer Protocol (protocolo de transferencia de archivo trivial).

El modelo TCP/IP es la esencia de Internet, sirve para enlazar computadoras que utilizan diferentes sistemas operativos, que pueden incluir, computadoras de escritorio, minicomputadoras y computadoras centrales sobre redes de área local y área extensa.

En esta semana se revisarán con más detalle los algoritmos criptográficos utilizados como seguridad en la capa de Aplicación del modelo TCP/IP, con el objeto de centrar el tema de la criptografía y hacer que los estudiantes conozcan por qué es tan importante la criptografía en la comunicaciones, especialmente cuando se transmite información sensible entre dos o más máquinas.

U4 Metodología

En este momento, los estudiantes han de entender la criptografía y los algoritmos simétricos y asimétricos, por lo tanto, se revisa en este escrito cómo se usan en la vida práctica los algoritmos y cuáles son los que se emplean para poder realizar transmisiones seguras a través de la red Internet o de las redes LAN en las cuales operamos todos los días, sin prestar demasiado cuidado.

Para lograr el anterior cometido, se hacen las siguientes recomendaciones:

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Desarrollar el taller propuesto.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

Criptografía en la capa de aplicación

SSL/TLS, SET y OPENSSL TOOLKIT

SSL/TLS

a. **SSL - Secure Sockets Layer - Capa de Conexiones Seguras.** Este protocolo utiliza certificados digitales con el fin de establecer comunicaciones seguras por internet.

La importancia de este protocolo radica en que se puede enviar información por la web, pues los datos se ocultan por medio de métodos criptográficos cuando se navega por la red. Este protocolo se utiliza en bancos, tiendas en líneas y todos los servicios que se necesiten al enviar datos personales o contraseñas, pero no todos los sitios web utilizan SSL, por lo que deberá tenerse cuidado.

Vale la pena que se entienda el término Autoridad Certificadora - AC, que hace referencia a una entidad que garantiza que quien posea un certificado digital, sea quien dice ser, de esta manera se brinda confianza a quien envía como a quien recibe el mensaje, esto es lo que se denomina comunicación segura SSL/TLS.

De otra parte, el Certificado digital SSL/TLS, es un documento digital único que certifica la vinculación entre una persona o entidad con su clave o llave pública. Este documento contiene la siguiente información del propietario de la llave: nombre, dirección, correo electrónico, empresa a la que pertenece su llave pública y la información propia del certificado, como por ejemplo: el periodo de validez, el número único de serie, el nombre de la autoridad Certificadora que emitió el certificado, la firma digital de la Autoridad Certificadora cifrada con su llave privada, entre otros.

Para ejemplificar el uso de SSL/TLS, teclee la siguiente dirección: www.facebook.com, verá usted que figura un candado, (ver imagen 1) todo dependerá del navegador que se utilice. Ese Candado indica que el SSL/TLS ha hecho el trabajo adecuadamente.

Imagen 1. Sitio Web oficial de Facebook
Fuente: Propia.

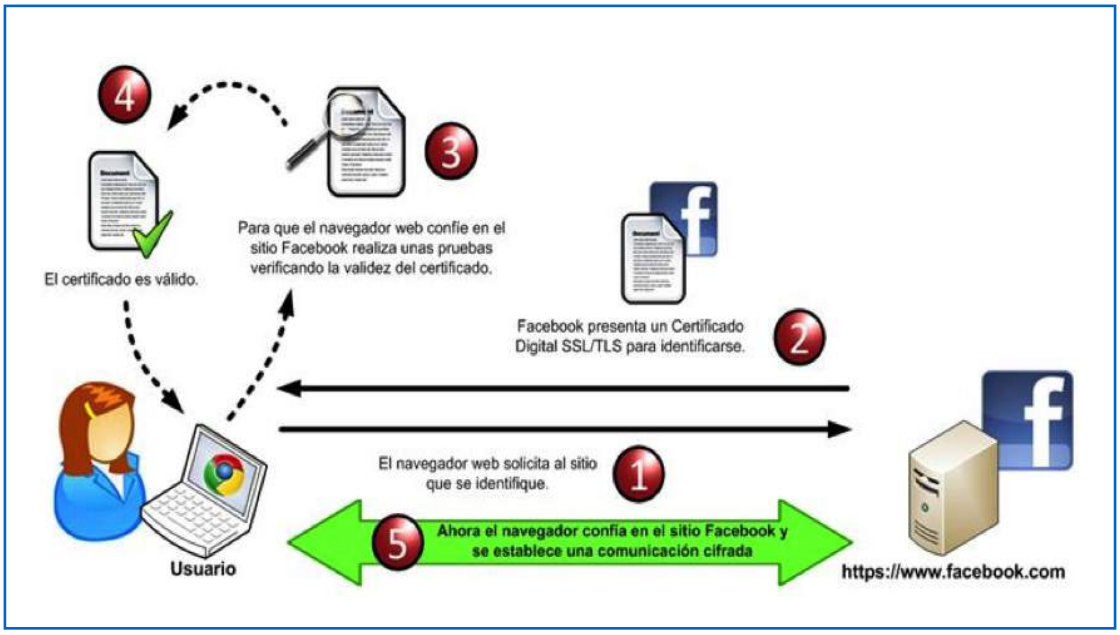


Imagen 2. Funcionamiento general de SSL/TLS
Fuente: http://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls

SSL-Secure Socket Layer: definido como un protocolo de seguridad, que permite la interacción entre usuarios de forma segura. Sus objetivos son: ofrecer seguridad criptográfica, interoperabilidad, extensibilidad y eficiencia relativa, cuando se realiza una comunicación por la red.

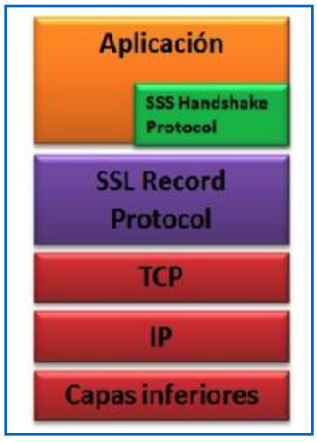


Imagen 3. Capas de protocolos en SSL
Fuente: Propia.

SSL permite que la comunicación fluya entre emisor y receptor de manera que sus comunicaciones no puedan ser interceptadas, modificadas o alteradas de alguna forma. Está conformado por dos niveles o capas: de una parte, se encuentra el SSL Record Protocol, y de otra el SSL Handshake Protocolo, como se muestra en la imagen 3.

En cuanto al SSL Record Protocol, tiene conexión privada y utiliza el cifrado simétrico (una sola clave para emisor y receptor) para el envío de mensajes, para ello utiliza el algoritmo DES, RC4, entre otros.

Cuando SSL Record Protocol va a cifrar un mensaje, negocia, por ejemplo con SSL Handshake Protocol para que en cada conexión se use solo una clave.

Los mensajes en esta capa, incluyen código de autenticidad del mensaje (MD5, SHA...), es decir la MAC.

En cuanto a SSL Handshake Protocol, utiliza el cifrado asimétrico, con el fin de lograr la autenticidad del emisor y receptor, esto lo hace mediante algoritmos como el RSA, DDS, entre otros.

SSL no depende de los protocolos que se implementan en la capa de aplicación, por ello, permite agregar niveles de seguridad adicional, como por ejemplo, Secure Electronic Transaction - SET.

El protocolo SSL permite que haya tres tipos de conexión a saber: Autentificada, semi-autentificada y anónima.

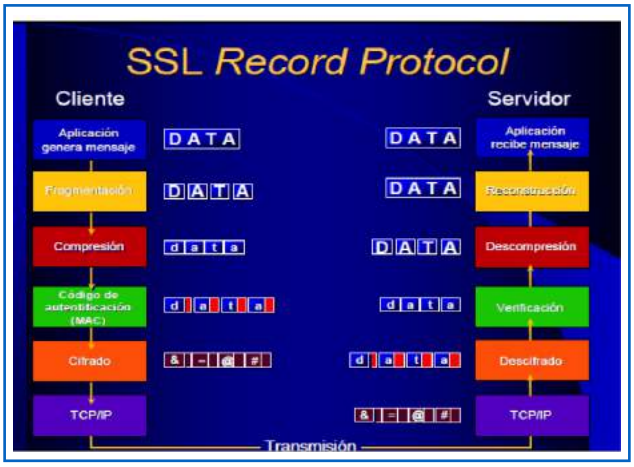


Imagen 4. SSL Record Protocol
Fuente: http://2.bp.blogspot.com/_YXGbhu29rAY/STYaSykeO9I/AAAAAAAAAAs/TEynPsdg-J8/s400/recordssl.jpg

En la imagen anterior, se nota cómo se realiza la comunicación entre un cliente y un servidor, cuando se envía un mensaje a través de la red y cómo se realiza el cifrado y descifrado del mensaje en las capas internas del y más bajas del sistema.

Imagen 5. SSL Handshake Protocol
Fuente: Propia.

En la imagen 5, se muestra el recorrido que se hace desde que se solicita una conexión al servidor, hasta cuando esta es recibida por el mismo servidor. SSL Handshake Protocol, permite mensajes opcionales o dependientes de la situación, como por ejemplo cuando se envía e certificado y se verifica el certificado, así como que se requiere certificado, que figuran en las distintas capas tanto del cliente como del servidor.

b. TLS - Transport Layer Security, protocolo que se basa en SSL y presenta pocas diferencias. Permite la gestión de las comunicaciones para que estas sean seguras cuando se realizan transacciones comerciales por medio de la red.

   TLS es un certificado que hace parte de la nueva generación del certificado SSL, está destinado al e-comerce, que encripta los datos cuando se hace intercambio de información por la web, protege los datos confidenciales como las tarjetas de crédito o cualquier otra información. Para garantizar el intercambio de datos de forma segura utiliza aplicaciones como el HTTP, POP3, IMAP, SSH, SMTP, NNTP.

   La encripción la realiza por medio de dos protocolos encapas distintas, ellos son el protocolo TLS Record Protocol y el TLS Handshake Protocol que hace las veces de mutuo acuerdo.

   Añadiendo algo más de lo visto anteriormente, el Record Protocol, autentica para que la transmisión de datos se haga por medio de una conexión privada, fiable e íntegra entre emisor y receptor. En cuanto al Handshake Protocol, él negocia el mensaje de forma segura, es decir, que en cada mensaje especifica el protocolo en un campo llamado content-type, el cifrado y empaquetamiento lo hace por medio de un código de autenticación o MAC.

TLS trabaja en un canal seguro y se hace un ciframiento de claves entre cliente y servidor, se negocia la criptografía del mensaje, se autentican las claves del cifrado y se realiza la transmisión segura.

Los cifrados de mensajes los asegura TLS para que sean transmitidos en internet con tecnología de cifrado estándar, esto hace que los correos electrónicos enviados no sean interceptados por intrusos.

TLS funciona de la siguiente manera:

1. Habilitar TLS en los servidores de correo del emisor y el receptor, para que cualquier información que se intercambie entre ellos esté cifrada, incluyendo el asunto, el texto y los adjuntos del correo enviado.
2. En el momento que el emisor se conecta con el recetor del mensaje, el sistema debe verificar automáticamente que esté habilitado TLS en el servidor de correos del cliente.
3. De estar habilitado TLS en los dos servidores de correos, se establece la conexión TLS segura, por medio de un "Handshake" o apretón de manos, que un procedimiento de protocolo de intercambio.
4. En el paso anterior se intercambian los certificados TLS, de tal forma que si el servidor del emisor confía en el certificado del servidor del receptor, se inicia la sesión TLS y el correo es enviado por medio de una conexión segura.

Al utilizar TLS no se requiere intervención ni del emisor ni del receptor, lo que lo hace confiable, estable, previo acuerdo entre el servidor de correos de las personas que se quieren comunicar.

Por todo lo anterior, TLS se ha convertido en un estándar que entrega beneficios para quienes intercambian correos por medio de la web, beneficios como la mayor protección, ofrece disponibilidad en la mayor parte de servidores de correo, permite escanear los correos para buscar virus, reduce costos, pues una empresa que lo utilice solo adquirirá el certificado TLS, por último, la implementación de TLS es muy sencillo, solo basta con configurarlo en los servidores de correo y no se necesita tenerlo instalado en las estaciones de trabajo individuales.

SET - Secure Electronic Transactions- Transacciones electrónicas seguras, fue propuesto por Visa y Master Card, ayudadas por American Express, Microsoft, IBM, Netscape, VeriSign, entre otras, con el fin de permitir seguridad en las transacciones electrónicas y brindar mayores garantías, para ello, utiliza certificados digitales, con el fin de verificar la identidad de los usuarios involucrado en la transacción, la verificación se hace por medio del cifrado asimétrico.

Al utilizar el protocolo SET, el comerciante puede disponer de un certificado digital que es emitido por la Autoridad de Certificación requerida, de otra parte, el comprador también dispone de un certificado digital que es emitido por la entidad que emite la tarjeta, el certificado incluye la firma digital de la institución y la fecha de expiración.

SET funciona así:

Mediante certificados digitales, tanto el comerciante como el cliente, se identifican y autentican.

El comerciante no puede ver ni conservar los datos del cliente, ni de su tarjeta una vez que se cierre la transacción.

Los datos viajan por la red, encriptados.

Los pasos que sigue una transacción SET, son los siguientes (tomado de El Portal de los Negocios en Internet):

- Cuando se va a cerrar el pedido, el cliente recibe la firma digital de la tienda y verifica su validez.
- El cliente envía al comerciante la siguiente información firmada digitalmente:
  - Los datos del pedido (básicamente: identificación del comerciante, importe y fecha).
  - La orden de pago, con una encriptación que sólo puede leer el banco.
  - La relación entre el pedido y la orden de pago, que los liga indisolublemente.
  - El comercio recibe el pedido y verifica la validez de la firma digital.
  - El comerciante pasa al banco la orden de pago (que él no ha podido leer) con su firma digital.
  - El banco autoriza la transacción y devuelve dos confirmaciones, una para el comerciante y otra para el titular de la tarjeta.

El protocolo SET proporciona seguridad, confidencialidad de los datos de las tarjetas de crédito, por cuanto el usuario comprado

se identifica ante la entidad financiera, por medio del certificado digital que emite por el banco, lo que hace que la información de la tarjeta de crédito no viaje por la red, lo que hace que dicha información no llegará al comerciante, ni mucho menos puede ser interceptada.

Cuando los datos viajan por la red, lo hacen encriptados y protegidos por la firma digital, lo que hace que no puedan ser alterados por la red, a esto se le denomina "la integridad de los datos".

Entre tanto, el comerciante se autentica ante el comprador, indicando que se encuentra autorizado para hacer cobros con tarjeta de crédito. Por su parte, el cliente o usuario se autentica ante el comerciante como el legítimo titular de la tarjeta de crédito que ejecute la transacción.

SET no es un protocolo gratuito, las entidades bancarias deben pagar por el certificado digital, aunque para el usuario es transparente el uso del mismo. Generalmente, cuando se utiliza SET, la duración de la transacción se establece entre 25 y 30 segundos, es mayor que el tiempo de que gasta SSL. El comerciante prevé mayor complejidad, por cuanto al trabajar con SET, debe hacerlo con SSL, pues no todos los clientes tienen certificado digital.

Hoy, se requiere que las transacciones comerciales que se hacen por medio de la red, tenga mayores seguridades, por ello, se está utilizando el protocolo SET, pues los negocios virtuales B2B, lo utilizan también.

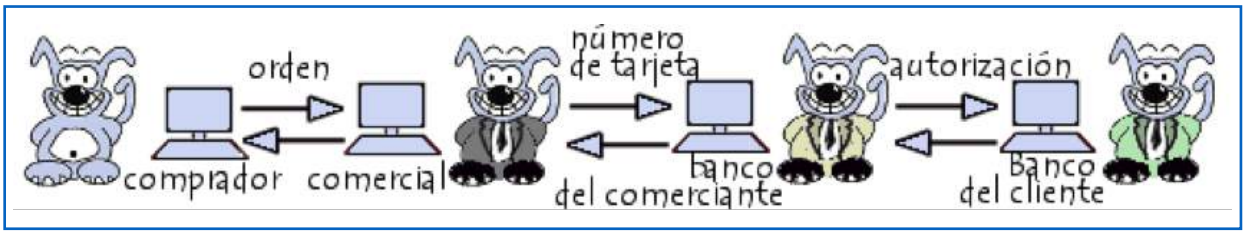


Imagen 6. Una típica compra de un cliente
Fuente: http://es.ccm.net/contents/136-criptografia-el-protocolo-set

En resumen, al utilizar el SET en una transacción segura, los datos del usuario-cliente se envían al servidor del vendedor, y este solo recibe la orden, entre tanto, los números de la tarjeta de crédito de X banco, son enviados al banco del vendedor, él lee los detalles de la cuenta bancaria del comprador y contacta con el banco para que los verifique en tiempo real.

OpenSSL

Proyecto de software libre desarrollado por Eric Young y Tim Hudson. Es un paquete de herramientas que permiten la administración y el uso de bibliotecas para criptografía, incluyen funciones criptográficas que utilizan por ejemplo, los paquetes OpenSSH y los navegadores web.

Las herramientas OpenSSL permiten al sistema la implementación de SSL o TLS, como también crear certificados digitales para aplicarse en servidores como el Apache. Otro de los usos de OpenSSL, es que ofrece certificados que se pueden utilizar con aplicaciones software para asegurar que las credenciales de quien las utilice (empresa o usuario individual) son válidas.

Una de las mejores características de OpenSSL es que ofrece una capa cifrada de transporte sobre una capa normal de comunicación, la cual permite combinar aplicaciones con los servicios de red.

Los usos más valiosos de OpenSSL es que permite la validación cifrada de clientes de correo, como también las transacciones basadas en web, como por ejemplo, el pago con tarjetas de crédito, entre otros.

OpenSSL es utilizado para:

a. Crear certificados digitales, los cuales garantizan técnica y legalmente la identidad de un usuario en Internet, también permiten la firma electrónica de documentos y cifrar en general las comunicaciones.
b. Instalar certificados digitales.
c. Manejar los certificados digitales
    - Generar y firmar certificados.
    - Revocar certificados.
    - Renovar un certificado.
    - Visualizar un certificado.

Para crear un certificado digital con OpenSSL, se hace lo siguiente:

- Se crea un certificado raíz, el cual genera dos partes, una llave primaria y un certificado.
- Se protege el certificado de ataques supuestos o robos y se usará solo para firmar otros certificados.
- Se crean dos archivos, uno denominado index.txt, el cual debe estar vacío y el otro serial, el cual contiene un 01.

Luego se procede a la instalación del certificado, para ello se crea este como certificado raíz y se almacena solo una parte del certificado. A continuación se guarda en el servidor web el archivo cacert.crt para poderlo descargar e instalar.

En seguida se puede proceder a utilizar el certificado, para ello, se crea una clave privada y una solicitud de certificado, se utiliza un nombre común, una dirección web para autentificar la página, el nombre del usuario para autentificar los correos de ese usuario.

OpenSSL es una aplicación que se trabaja desde la línea de comandos para poder usar las funciones criptográficas de su librería criptográfica desde la shell.

Otros usos de esta aplicación es para:

Crear claves RSA, DH y DSA, crear de certificados X.509, CSRs y CRLs, calcular funciones resúmenes de mensajes (MD5), encriptar y desencriptar con distintos algoritmos de cifrado simétricos (DES, IDEA, ...), realizar pruebas cliente/servidor utilizando el protocolo SSL.

PGP, Estándar OPENPGP y GNUPG

PGP

PGP - Pretty Good Privacy. Este programa fue desarrollado por Phil Zimmermann en 1991, su objetivo es la protección de la información que viaja a través de la red, por medio de criptografía de clave pública, además autentica documentos por medio de firmas digitales. Es el sistema para encriptación de comunicaciones por Internet más utilizado en el mundo

En otras palabras PGP respalda el autenticado de mensajes y comprueba su integridad. PGP no solo protege os datos que viajan por la red, sino que también lo hace con los datos que se encuentran en los discos y en las copias de seguridad.

Imagen 7. Envío de un mensaje utilizando PGP
Fuente: Propia

Como características principales de PGP se pueden enumerar las siguientes:

- Aporta privacidad al correo electrónico
- Como funciones principales se indican: encripta mensajes, acepta firma digital, comprime datos, ofrece compatibilidad de correo.
- Incorpora el algoritmo de ecripción AES-256.
- Posee la función de resumen SHA-1.
- Comprime datos BZip2, ZLIB, Zip.

Para utilizar el PGP, ambos, emisor y receptor deben instalarlo en sus máquinas, de esta forma se genera automáticamente las claves privadas y públicas.

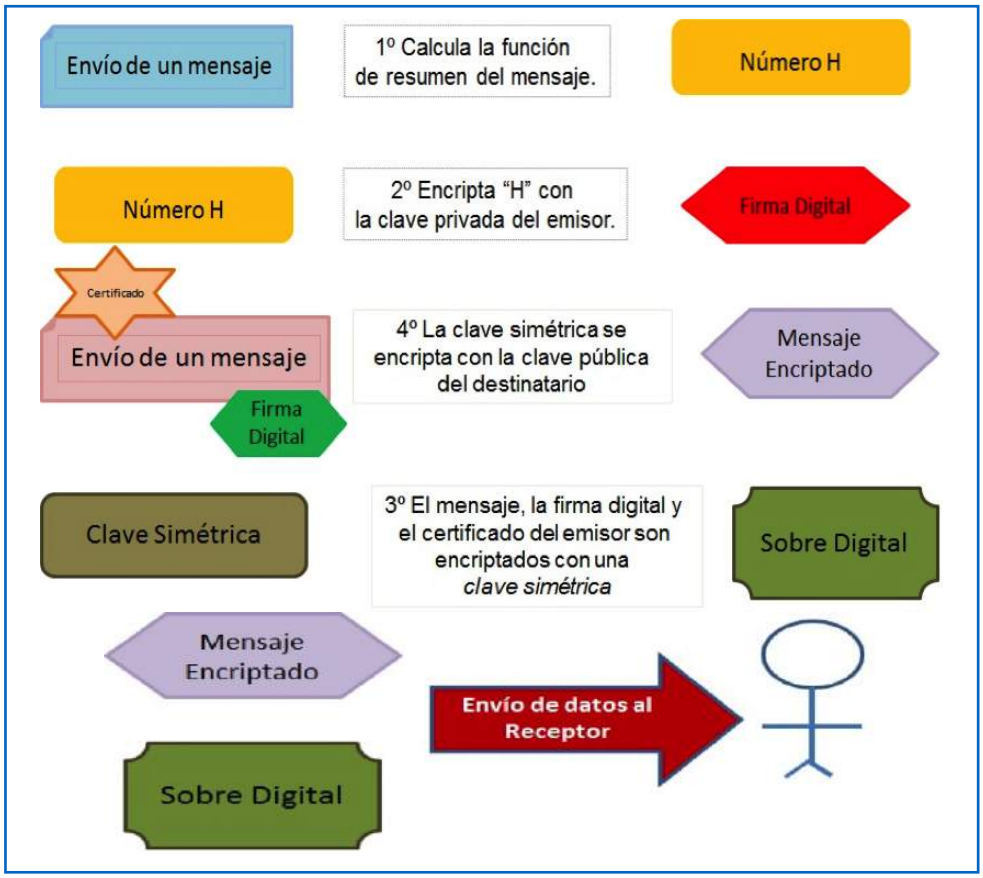


Imagen 8. PGP en el Emisor
Fuente: Propia.

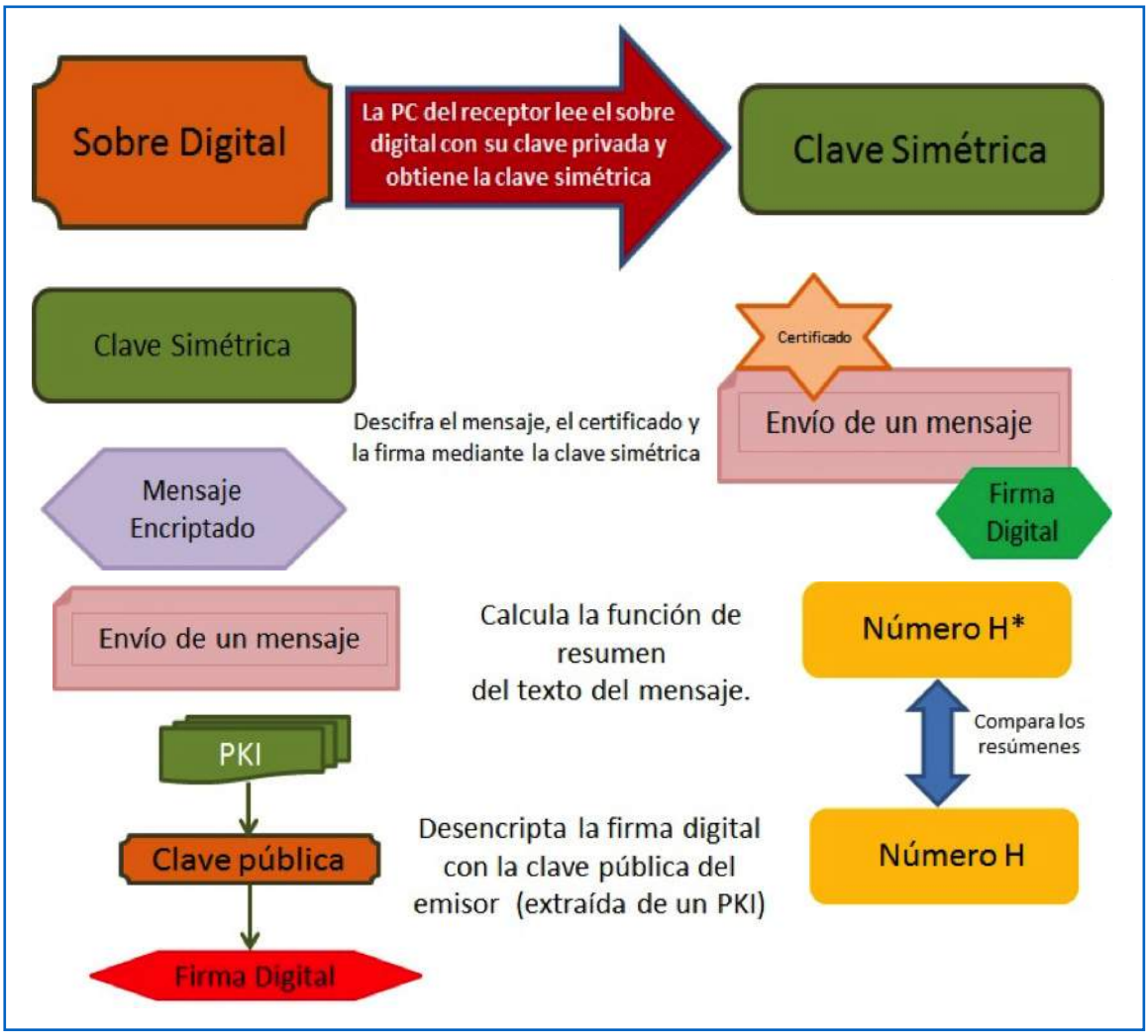


Imagen 9. PGP en el Receptor
Fuente: Propia.

En las imágenes 8 y 9, se muestra cómo procede el protocolo PGP tanto en el servidor Emisor y en el Receptor, desde que se envía el mensaje, hasta que se recibe.

De las imágenes 8 y 9, se puede concluir lo siguiente:

Se presentará Integridad de un mensaje cuando los resúmenes coinciden. Además se revelará Identidad, por medio de la firma digital, porque se sabrá que el mensaje ha sido enviado por el verdadero emisor. También se manifestará Confidencialidad del mensaje porque el cifrado de clave pública garantiza que el mensaje no ha sido leído por algún intruso.

Una característica de PGP es que es muy eficiente cuando se intercambian mensajes, porque cuando es posible, los datos son comprimidos antes de cifrarlos y/o después de formarlos.

Los paquetes, son la estructura de datos predilecta por PGP para codificar esos datos. Es así que un mensaje PGP se puede formar por uno o más paquetes PGP. Estos paquetes no son otra cosa que un flujo de bytes compuestos de una cabecera que identifica el tipo de paquete y su longitud; luego tiene unos campos de datos que dependerán del tipo de paquete.

Los tipos de paquetes PGP son los siguientes:

a. Paquetes de datos literales: utilizados para representar el claro, sin cifrar. Este paquete tiene dos campos, el uno indica el valor de los datos y el otro dice si el valor de los datos se ha de procesar como texto o como datos binarios.

b. Paquetes de datos comprimidos: compuesto por un campo para indicar el algoritmo que hará la compresión de los datos y el otro campo contiene una secuencia de bytes comprimidos, que al descomprimirlos, resultará uno o más paquetes PGP.

c. Paquete de datos cifrados con clave simétrica: compuesto por una secuencia de bytes cifrados por medio de un algoritmo simétrico, al igual que en el caso anterior, al descifrar los datos, dará como resultado uno o más paquetes PGP. Lo que se cifra en este paquete son paquetes de datos en claro o paquetes de datos comprimidos.

d. Paquete de datos cifrados con clave pública: paquete que tiene tres campos, el uno para identificar la clave pública utilizada, el otro para indicar el algoritmo de ciframiento utilizado y el tercer campo usado para almacenar los datos cifrados. Este paquete es usado para cifrar la clave des sesión, que ha sido utilizada para generar un paquete de datos cifrados simétricamente, también se usa para enviar un mensaje con sobre digital. La clave pública que se utiliza en este paquete es la de cada uno de los destinatarios del mensaje.

e. Paquete de firma: contiene campos que almacenan información como por ejemplo, la clase de firma como:

   - firma de datos binarios.
   - Firma de texto canónico.
   - Certificado (agrupación de clave pública y nombre de usuario).
   - Revocación de clave pública.
   - Revocación de certificado.
   - Fechado – timestamp, que contiene, la fecha y hora en que se creó a firma, el identificador de la clave con la que se ha credo, los algoritmos utilizados para el hash y para el cifrado simétrico, la firma obtenida aplicando los algoritmos que se especifican en los datos que se debe firmar concatenados con los campos autenticados.

f. Paquete de clave pública: contiene información relativa a una clave pública como: fecha de creación de la clave, el algoritmo a que corresponde la clave, los valores de los componentes de la clave, cuando se trate del algoritmo RSA, llevará como valores el módulo n y el exponente público e. La clave pública utiliza un número de ocho bytes que sirve para buscar el valor de la clave en una base de datos.

g. Paquete de nombre de usuario: contiene una cadena de caracteres, utilizada para identificar el propietario de una clave pública.

h. Paquete de clave privada: utilizado para almacenar los componentes de la clave privada de un usuario. La confidencialidad se logra a través de un archivo donde se almacena el paquete con los componentes secretos de la clave, los cuales deberán estar cifrados con una clave simétrica derivada de un passphrase, para que cuando el usuario desee descifrar o firmar el mensaje con la clave privada, deba indicar la passphrase para obtener los valores necesarios. De aquí que un usuario puede tener varas claves asociadas al mismo nombre o a diferentes nombres.

i. Paquete de nivel de confianza en una clave: paquete que no se envía nunca, sino que se almacena en el paquete de claves de cada usuario. Es utilizado este paquete para indicar qué confiabilidad tiene la clave certificadora, lo que quiere decir que utiliza otras claves para asociar claves y nombres de usuarios.

Cómo distribuye las claves PGP

La certificación de claves PGP sigue el modelo de las autoridades de certificación X.509 y tiene un modelo descentralizado llamado de confianza mutua o malla de confianza, como se muestra en la imagen 10.

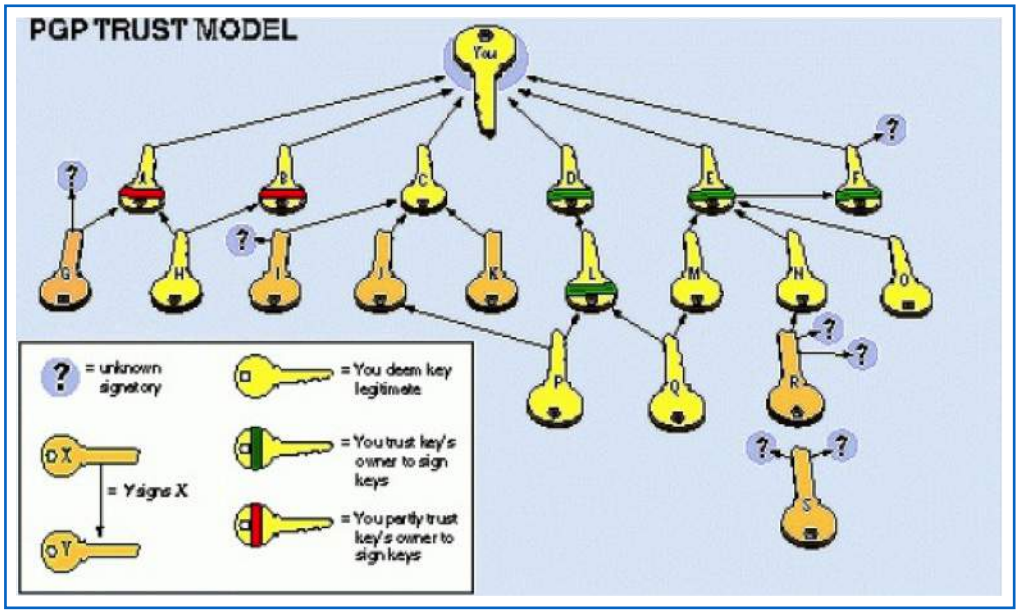


Imagen 10. El modelo de confianza PGP
Fuente: http://datateca.unad.edu.co/contenidos/233011/233011Exe/imagen13.jpg

Cuando un usuario genera su par de claves PGP (pública y privada), a la clave pública le tiene que asociar un nombre de usuario y, a continuación, tiene que autocertificar esta asociación, es decir, firmar con su clave privada la concatenación de la clave pública y el nombre. El paquete de la clave pública, el del nombre de usuario y el de la firma forman un bloque de clave.

Opcionalmente, el usuario puede asociar más de un nombre a la clave (por ejemplo, si tiene diversas direcciones de correo electrónico) y entonces, el bloque estará formado

por la clave pública y tantos pares nombre-firma como sea preciso.

Otra manera en que se realiza la distribución de claves, es a través del servido de claves PGP, que almacena todas las claves públicas con sus certificados, los servidores se sincronizan entre ellos, de tal forma que se realicen las actualizaciones del almacén y se distribuyan de forma automática a todos los servidores, la falla estará en que los servidores almacenan cualquier llave pública sin verificar la identidad de quien guarda la clave, por lo que cada usuario que use el servidor PGP, será quien se asegure de lo autenticidad de su clave.

OpenPGP

OpenPGP es un estándar de cifrado de correo electrónico, muy utilizado a nivel mundial. Se define por el Grupo de Trabajo OpenPGP del Internet Engineering Task Force (IETF) es una propuesta RFC Standard 4880. El estándar OpenPGP fue derivado originalmente de PGP (Pretty Good Privacy).

OpenPGP es un protocolo no propietario para el cifrado de correo electrónico utilizando la criptografía de clave pública. Se basa en PGP, que fue desarrollado por Phil Zimmermann. El protocolo OpenPGP define formatos estándar para los mensajes cifrados, firmas y certificados para el intercambio de claves públicas.

A partir de 1997, el Grupo de Trabajo OpenPGP se formó en la Internet Engineering Task Force (IETF) para definir esta norma que anteriormente había sido un producto propietario desde 1991. Durante la década pasada, PGP, y más tarde OpenPGP, se han convertido en el estándar para casi todos los correos electrónicos cifrados del mundo.

Al convertirse en un estándar IETF propuesto (RFC 4880), OpenPGP puede ser implementado por cualquier empresa sin pagar derechos de licencia.

La Alianza OpenPGP reúne a empresas que persiguen un objetivo común de promover el mismo estándar de cifrado de correo electrónico y aplicar la PKI a otras aplicaciones que no sean de correo electrónico.

OpenPGP es un estándar de Internet orientado a la interoperabilidad de mensajes que se protegen con criptografía. Utiliza dos programas a saber: Pretty Good Privacy (PGP) y GNU Privacy Guard (GPG).

El correo electrónico es una herramienta que a diario millones de personas utilizan, sin imaginar lo inseguros que son; pero existen programas de software que permiten proteger los datos que viajan a través de la red, como es el OpenPGP, cuyo objetivo es el de salvaguardar esa información que se distribuye a través de Internet, por medio de cifrado de clave pública, de esta manera, se facilita la autenticación de documentos por medio de firmas digitales.

El OpenPGP, es un sistema utilizado para encriptar y desencriptar datos que van por medio de la red, entre ellos, el correo electrónico, también almacenamiento en el disco duro. Este sistema utiliza la criptografía de clave pública.

Estas herramientas OpenPGP son de uso libre para todas las personas y corren bajo cualquier sistema operativo y se encuentran en distintos idiomas.

GnuPG

GNU Privacy Guard o GPG. Esta es una herramienta utilizada en la seguridad de las comunicaciones electrónicas.

Algunas de las tareas de GNUPG es la generar el par de claves (pública y privada), así como intercambiar y comprobar la su autenticidad; a la vez, cifra y descifra documentos, permite la firma digital de documentos y las verifica.

Imagen 11. Logo de GnuPG
Fuente: https://lignux.com/crear-claves-gpg-desde-la-terminal-en-gnulinux/

La criptografía que utiliza GnuPG es la de clave pública, con el objeto que los usuarios se comuniquen de forma segura. Recordando, el sistema de claves públicas utiliza dos claves, una privada y una pública, de las cuales cada usuario mantiene la clave privada muy secreta, mientras que la clave pública la conoce cualquier persona con la que la persona desee comunicarse.

El esquema que practica GnuPG consiste en que el usuario tiene el par de claves primario y ninguno o más de un par de claves adicionales subordinadas. Todas estas claves, primarios y subordinados están agrupadas, con el fin de facilitar la gestión de claves, con lo que el grupo se considera como un solo par de claves.

Pero GnuPG tiene la capacidad de crear varios tipos distintos de pares de claves, de las cuales una clave primaria debe ser capaz de generar firmas. De aquí que surgen tres opciones:

1. Se generan dos pares de claves, de las cuales una, es el par de claves DSA, llamado primario y se utiliza para firmar únicamente.Genera también un par de claves ElGamal llamadas subordinadas, utilizadas para el cifrado.
2. Sólo se generarán un par de claves DSA.
3. Se generan un único par de claves ElGamal, utilizadas para cifrar y descifrar.

No olvidar que cuando se escoge una clave, debe indicarse el tamaño, que para las claves DSA es de 412 a 1024 bits, mientras que la clave ElGamal puede ser de cualquier tamaño, pero en el caso de GnuPG, las claves no debe ser inferiores a 768 bits. Por lo anterior, si se escoge la opción 1 y el tamaño de clave mayor a 1024 bits, entonces ElGamal tendrá el tamaño deseado, pero DSA solo alcanzará a 1024 bits. Una vez que se asigne el tamaño de clave, no se podrá cambiar nunca. Luego se escogerá la fecha de caducidad de la clave, por cuanto si se escoge la opción 1, la fecha de caducidad se tomará para los pares de claves de ElGamal y DSA.

PnuPG criptosistema que reemplaza en su totalidad a PGP, pues no utiliza IDEA ni RSA, se utilizan sin ninguna restricción y cumple las normas RFC2440.

Sus principales características:

- No utiliza algoritmos patentados
- Se utiliza como un programa de filtrado
- Implementa los algoritmos: ElGamal en firma y cifrado, DSA, TDESm BlowFish, MD5, SAH-1, entre otros.

- Como tiene módulos de extensión, implementa nuevos algoritmos
- El formato estándar que posee, permite identificar al usuario
- Puede implementar la fecha de caducidad
- Puede ser utilizado libremente de forma personal pero no comercial.

Las funciones básicas de GnuPG. Estas funciones incluyen generar un par de claves, intercambiar y comprobar la autenticidad de claves, cifrar y descifrar documentos, y firmar documentos y verificar firmas digitales.

Utiliza criptografía de llave pública, con el objeto que los usuarios se comuniquen de forma segura.

Cómo utilizar GnuPG

Primero se instala el paquete, luego se generan el par de claves, a continuación se escoge el periodo de validez o de caducidad, luego se crea el usuario que va a utilizar la clave y la contraseña privada,

El usuario intercambia las claves públicas, para poder comunicarse con otros usuarios y puede generar claves públicas.

Se exporta la clave pública para poderla utilizar por correo o con otros equipos.

Ahora se puede cifrar/descifrar archivos o firmar un archivo.

También se puede crear un certificado de revocación, por si se olvida la clave o se pierde la misma.

Se puede descargar desde la página oficial: https://www.gnupg.org/download/index.en.html

Al personalizar el uso de GnuPG, se debe tener en cuenta:

- Elegir el tamaño del par de claves pública y privada, para prevenir ataques de fuerza bruta.
- Proteger la clave privada, para ayudar a prevenir que un atacante llegue a utilizarla para descifrar los mensajes o para firmar mensajes a nombre del dueño de la clave privada.
- Seleccionar la fecha de caducidad y el uso de subclaves
- Gestionar el anillo de confianza, para ayudar a evitar que algún atacante se haga pasar por una persona de confianza.

Seleccionar el tamaño de la clave

El tamaño de la clave depende de la clave en sí. El par de claves pública y privada contienen en su mayoría claves múltiples, por ejemplo, una clave de firmado maestra, posiblemente una o más subclaves de cifrado adicionales. Cuando se utilizan los parámetros que trae por defecto GnuPG para generar claves, la clave maestra será una DSA y las subclaves serán claves ElGamal.

Otro aspecto importante, es el cuidado de la clave privada y su tamaño, por ello es conveniente poder recordar con facilidad una clave, pero que para otros sea difícil de adivinar, esto quiere decir, incluir en una clave, números, letras, caracteres numéricos y caracteres especiales, pues de la fortaleza de la clave dependerá la privacidad de la información que se cifre.

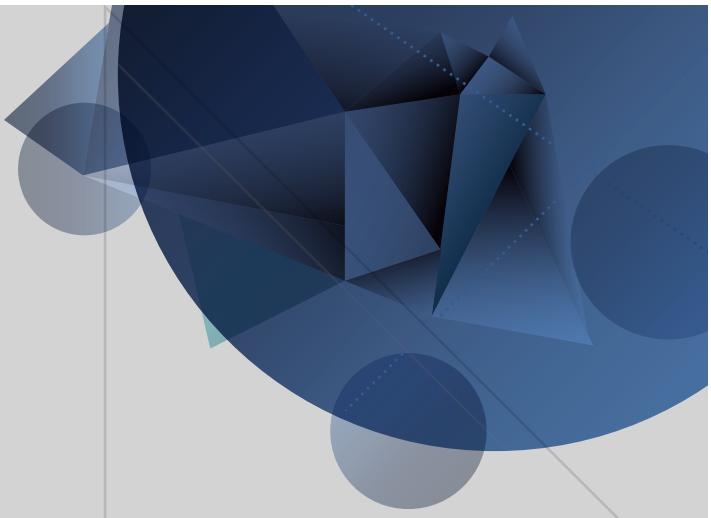
Ejercicio para realizar

De la dirección: https://www.gnupg.org/download/index.en.html descargue el paquete GnuPG y una vez instalado, cree un par de claves pública y privada.

4

Unidad 4

Criptografía en la capa de aplicación



Criptografía y mecanismos de seguridad

Autor: Lucy Medina

AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO

Introducción

En esta semana ocho, se continúa con el tema visto en la semana anterior. Los temas que aquí se estudian, corresponde a las herramientas denominadas OpenSSH que permiten realizar comunicaciones cifradas a través de la red, para ello se utiliza el protocolo SSH.

De la misma manera, se estudia el protocolo SSH, que encripta todo lo que envía y recibe, con el fin de facilitar las comunicaciones seguras entre dos sistemas que utilizan la arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.

Enseguida de este escrito se estudia el protocolo Kerberos, el cual puede verse como un servicio que se utiliza en una arquitectura cliente-servidor, con el fin de ofrecer seguridad a cada una de las transacciones que se lleven a cabo en la red.

Por último, en este escrito, se revisarán los temas que hacen referencia a las aplicaciones criptográficas, TrueCrypt, AxCrypt, STunnel y OpenVPN,

U4 Metodología

Ya se está en la última semana, es hora que el estudiante tenga los conceptos claros que se han analizado en el transcurso de esta asignatura. Por ello, es pertinente que por medio de este escrito avance en el conocimiento sobre los protocolos y herramientas criptográficas que se ubican en la capa de Aplicación del modelo TCP/IP.

Para lograr los cometidos expuestos, se hacen las siguientes recomendaciones:

Recomendaciones:

- Realizar las lecturas asignadas.
- Elaborar cada uno de los trabajos asignados.
- Utilizar a conciencia, los recursos de aprendizaje sugeridos.
- Realizar las evaluaciones propuestas.
- Cuando no entienda algo, comuníquese con su tutor.
- No continúe con el siguiente tema, hasta que esté perfectamente claro el tema actual.
- Siempre que lea, hágalo entendiendo las ideas.

Criptografía en la capa de aplicación

SSH y herramientas OpenSSH

SSH

SSH-Secure Shell. Protocolo utilizado para facilitar las comunicaciones seguras entre los sistemas que utilicen una arquitectura Cliente-Servidor. El trabajo de SSH, es el de encriptar la sesión de conexión, lo que hace más difícil que algún intruso se apodere de contraseñas no encriptadas.

Algunas características del protocolo SSH, es que proporciona protección luego de establecer la conexión inicial entre un cliente y un servidor. SSH es rápido, ya que sólo hace envíos de texto, lo que permite enviar menos datos, como por ejemplo imágenes de pantalla o programas vcn.

Otras particularidades de SSH, es que el cliente puede verificar que sí se encuentra conectado al servidor correspondiente. También el cliente puede transmitir la información que lo autentica ante el servidor, para ello utilizará 128 bits como encriptación, la cual es robusta. Otra característica de SSH, es que tanto los datos enviados como recibidos en la sesión establecida, se encriptan con 128 bits, lo que los hace muy difícil de descifrar para ser leídos. SSH permite utilizar aplicaciones gráficas sobre la red, las cuales son reenviadas desde el servidor, a esta técnica se le llama "reenvío por X11".

SSH también utiliza la técnica denominada "reenvío por puerto", que consiste en que como SSH encripta lo que se envía o recibe desde un cliente-servidor o viceversa, el servidor SSH se convierte en un conducto para que todos los protocolos inseguros se vuelvan seguros.

Al usar SSH desde el inicio de una sesión remota o para copia de archivo, se disminuyen las amenazas de seguridad como son, la intercepción de comunicación entre dos sistemas a través de algún paquete sniffer o la personificación de un host en particular. Lo anterior se entiende como aquella seguridad que el servidor SSH, así como el servidor utilizan firmas digitales para realizar la verificación de las entidades, teniendo presente que la comunicación que existe entre el cliente y el servidor estará encriptada. Debido a lo anterior, hace casi imposible aquellos intentos por falsificar la identidad de alguno de los participantes en la comunicación, pues los paquetes enviados o recibidos, se encontrarán cifrados a través de una clave conocida únicamente por el sistema cliente y por el servidor.

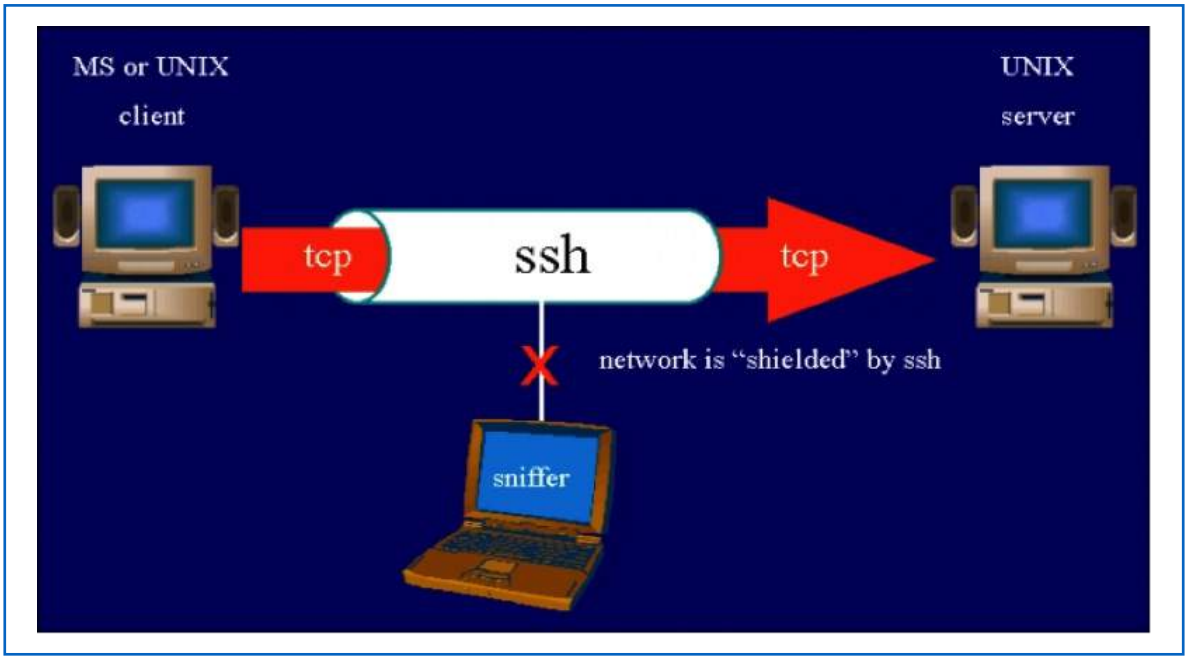


Imagen 1. Protocolo SSH
Fuente: http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/560-protocolo-ssh-secure-shell

En la imagen 1, se puede apreciar la conexión entre un cliente y un servidor, de tal manera que cuando el cliente es autenticado ante el servidor, estos tienen la posibilidad de utilizar distintos servicios seguros a través de la conexión, por ejemplo, usando una sesión Shell interactiva, aplicaciones X11 o túneles TCP-IP, evitando así la interferencia de un intruso, el cual puede utilizar aplicaciones sniffer.

Para utilizar SSH, se requieren los siguientes elementos:

Un servidor y un cliente.

El servidor SSH que deberá contar con:

- Una conexión a la red Internet en lo posible.
- El puerto 22 abierto y un firewall instalado.

Un computador local que tenga lo siguiente:

- Un cliente SSH.
- Tener instalada por ejemplo, la aplicación PuTTY si usted tiene un sistema Windows, PuTTY es un cliente SSH con licencia libre y está disponible para Windows, Unix, Linux. (La aplicación PUTTY, es un emulador gratuito utilizado con SSH entre otros protocolos, para realizar conexiones y conectar máquinas Windows con servidores Linux o Unix).
- Saber la dirección IP del servidor o máquina remota.
- Tener el usuario válido del servidor o máquina remota.
- La contraseña del usuario que se encuentra en la computadora remota o servidor.
- El puerto 22 de salida deberá estar habilitado en el firewall.

Cuando se conecta un cliente SSH a un servidor, se siguen los siguientes pasos:

1. El cliente abre una conexión TCP al puerto 22 del servidor.
2. Cliente y servidor se ponen de acuerdo en la versión del protocolo que van a usar, todo de acuerdo a la configuración y capacidades de cada máquina.
3. El servidor envía al cliente su clave pública, pues él tiene el par de claves pública-privada de RSA, que se llaman "claves de host".
4. El cliente hace una comparación de la clave pública que tiene almacenada con la enviada por el servidor, con el fin de verificar la autenticidad de la clave.
5. Una clave de sesión aleatoria es generada por el cliente, quien deberá seleccionar un algoritmo para realizar cifrado simétrico.
6. Utilizando el algoritmo RSA, el cliente hace envío de un mensaje que contiene clave de sesión y el algoritmo que seleccionó para el cifrado.
7. A partir de ese momento, en la comunicación se usará el algoritmo de cifrado simétrico que fue seleccionado, así como la clave de sesión compartida.
8. A continuación, el usuario se autentica.
9. Se inicia la sesión interactiva entre cliente y máquina remota.

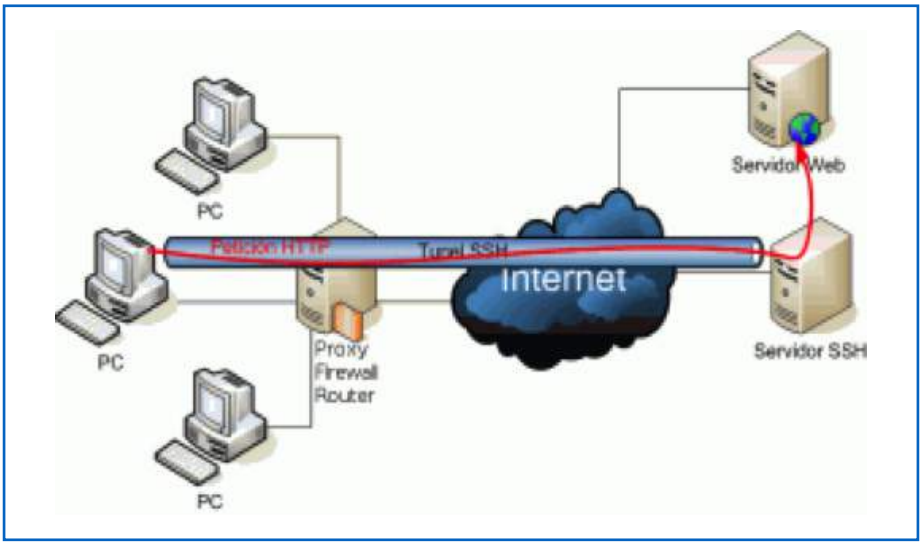


Imagen 2. Túnel SSH
Fuente: http://geekland.eu/que-es-y-para-que-sirve-un-tunel-ssh/

En la imagen 2 se aprecia, la construcción que se ha hecho de un túnel SSH, el cual proporciona seguridad en la comunicación entre un cliente y la máquina remota. El túnel SSH, sirve para minimizar los ataques por parte de intrusos, pues hoy en día la tecnología impulsa a los usuarios a estar conectados casi todo el tiempo a una red, lo que los hace vulnerables, pero si se establece el túnel SSH, será una solución para navegar con tranquilidad sobre la red, así no sea segura. Por ello, la confidencialidad estará asegura, lo mismo que la integridad de los datos que se envían reciben. El túnel también es útil para vulnerar algunas restricciones que tie-

nen los proxis y los firewalls, estos túneles aseguran la comunicación entre dos computadoras y permiten que los protocolos débiles se fortalezcan, de estos, como el SMTP, HTTP, FTP, TELNET, entre otros. Hoy es mucho más sencillo montar un servidor SSH que un servidor VPN o un Proxy.

Luego de mencionar las maravillas que ofrece el túnel SSH, existen dos falencias que también se deben considerar, como por ejemplo, cuando se comunican el cliente y el servidor, hay tramos en los cuales los datos se transmiten sin encriptar. Estos tramos van del servidor SSH y el servidor web, así como del servidor web al Servidor SSH. De otra parte, se deben configurar aplicación por aplicación, para poder realizar peticiones por medio del túnel SSH.

Para utilizar el protocolo SSH en Windows se utiliza la aplicación Putty, que permite hacer conexiones a distintos puertos, entre ellos el puerto 22, que es el puerto por defecto de SSH. En cambio, para Linux y Unix, el protocolo SSH viene incorporado por defecto.

Algunos de los ataques que previene SSH son:

- Ataques proveniente de Sniffers.
- IP Spoofing.
- MACpoofing.
- DNS Spoofing.
- Telnet Hickjacking.
- ARP Spoofing.
- ARP Spoofing.
- IP Routing Spoofing.
- ICMP Spoofing.

OpenSSH

OpenSSH -Open Secure Shell- es una suit compuesta por varias herramientas como ssh, scp (Secure Copy), sftp (Secure file Transfer Progrma), sshd (el demonio SSH), estas herramientas permiten realizar conexiones cifradas para comunicaciones sobre la red utilizando el protocolo SSH. Este conjunto de herramientas ha sido desarrollado como alternativa abierta a SSH que es propietario.

OpenSSH se utiliza para la autenticación y sirve para realizar implementaciones en diferentes sistemas operativos. Es capaz de autenticar usuarios utilizando sistemas como Keyboard interative, Kerberos y clave pública, que ya trae implementadas.

Una característica importante de OpenSSH, es que cifra la información que viaja por la red, con el fin de eliminar cualquier intrusión de personas no autorizadas o aquellas que secuestran conexiones o cualquier otro tipo de ataque sobre la información en la capad de red.

Algunas propiedades de protección de OpenSSH, es que el cliente puede verificar que realmente está conectado a un mismo servidor, de otra parte, la información de autenticación utiliza 128 bits, al igual que los datos que se envían o reciben se encriptan con 128 bits; posibilita en envío de aplicaciones que vienen del intérprete de comandos o lo que se denomina reenvío X11.

En el evento que se desee instalar OpenSSH se deberá descargar el paquete ssh, pues este paquete instala el servidor y el cliente.

Una vez que se instale la suit OpenSSH se puede configurar dos tipos de archivos, uno es el dedicado al cliente, con ssh, scp y sftp; el otro está orientado al servidor.

De otra parte, algunas herramientas que utiliza OpenSSH son: Ssh, scp, sftp, sshd, ssh-keygen, ssh-agent, ssh-add, ssh-keyscan.

Los archivos de configuración que se encuentran en el directorio para el servidor son:

Moduli, que se compone de grupos Diffie-Hellman, estos se utilizan para el intercambio de claves.

ssh_config, archivo encargado de la configuración del sistema cliente SSH.

sshd_config, archivo que permite la confiruación del demonio sshd.

ssh_host_dsa_key, es la clave privada RSA que utiliza el demonio sshd.

ssh_host_dsa_key.pub, es la clave privada DSA que utiliza el demonio sshd.

ssh_host_rsa_key, es la clave privada RSA que utiliza el demonio sshd para la versión 2 del protocolo SSH.

ssh_host_rsa_key.pub, archivo que contiene la clave pública RSA, utilizada por el demonio sshd para la versión 2 del protocolo SSH.

Entre tanto, los archivos para configurar el cliente se encuentran en el directorio de trabajo de cada uno de los usuarios.

Los archivos que se encuentran dentro del directorio .ssh para configurar el cliente son:

Identity.pub, es la clave pública RSA utilizada por ssh para el protocolo SSH en su versión 1.

Known_hosts, archivo que contiene las claves de host DSA de los servidires SSH, este archivo asegura que el cliente SSH está conectado al servidor SSH correcto.

Kerberos

Kerberos es un protocolo de autenticación de redes y fue creado por el MIT. Es un servicio que trabaja en una arquitectura Cliente-Servidor. Tiene la particularidad de que las computadoras puedan interactuar en una red insegura, demostrando su identidad proporcionando seguridad a las transacciones en las redes.

Conocido como un protocolo de seguridad, pues utiliza criptografía de claves simétricas con el fin de evaluar los usuarios con los servicios de red, de esta forma no se envían las contraseñas a través de la red, es así que los usuarios no autorizados no podrán capturar las contraseñas que van por la red.

Quizá una de las principales tareas de Kerberos es la de eliminar la transmisión por la red de la información utilizada para la autenticación, por ello es que Kerberos quita la amenaza de los analizadores de paquetes que pueden interceptar contraseñas en la red.

Resumiendo, Kerberos impide que las claves se envíen por la red y centraliza la autenticación de usuarios, para ello mantiene una sola base de datos de usuarios para toda la red.

El uso de Kerberos simplemente es el de restringir el acceso a los usuarios autorizados y lo que hace también es que autentica los requerimientos a servicios, tomando en consideración que esté en un entorno distribuido abierto, pues esos usuarios acceden a los servicios de los servidores que se distribuyen a través de la red.

El funcionamiento general de Kerberos es el siguiente:

- Cada uno de los usuarios posee una clave.
- Cada uno de los servidores posee una clave.
- En Kerberos, existe una base de datos que contiene todas las claves, tanto de usuario como de los servidores.
- La clave de cada usuario se deriva de su contraseña y estará cifrada.
- Las claves de los servidores se generan de forma aleatoria.
- En Kerberos deberán registrarse los usuarios que necesiten utilizar servicios, así como los servicios de red que requieran autenticación, también estarán registrados.
- Cuando los usuarios se registren pueden negociar sus claves privadas.
- Como Kerberos conoce todas las claves privadas, entonces crea mensajes para indicar al servidor sobre la autenticidad de un usuario que necesita un servicio que está en el servidor.

Kerberos presenta algunos niveles de protección como son:

La autenticación. Porque prueba que el usuario es quien dice ser, en términos generales, garantiza que las identidades tanto el usuario como del servidor son las verdaderas. Este nivel ofrece la Integridad, debido a que se verifica la validez de los datos que se transmiten entre cliente y servidor, a la vez, entrega privacidad, debido a que los datos son cifrados, es así que Kerberos se pueden iniciar sesiones en equipos diferentes pero que tengan instalado el servicio Kerberos, se pueden ejecutar comandos, intercambiar datos y se pueden enviar archivos de forma segura.

Integridad de los datos. Confirma que los datos no se modificaron en tránsito, es lo que comúnmente se denomina Mensaje Seguro.

Privacidad de los datos. Pues cumple con su objetivo principal y es el de que los datos no sean leídos en el transcurso de la transmisión, para ello cifra y autentica cada uno de los mensajes privados.

Cómo funciona Kerberos. Los servidores Kerberos se les denominan Kerberos Distribution Center - KDC y entrega dos servicios muy importantes como son, la Autenticación - AS- Authentication Service y el Ticket - TGS- Ticket Granting Service.

Cuando se utiliza el servicio AS, este tiene como meta autenticar la primera vez a un usuario y le entrega un ticket, que servirá para comunicarse con otro usuario.

El servicio de Tickets entrega a los usuarios, los documentos necesarios para establecer la comunicación con el servidor que le entregará el servicio, pues él es el que posee una base de datos con las claves privadas de los usuarios.

Cuando se habla de Arquitectura de Kerberos, puede indicarse que se basa en tres objetos imprescindibles de seguridad como son: la Clave de Sesión, el Ticket y el Autenticador (ver imagen 3).

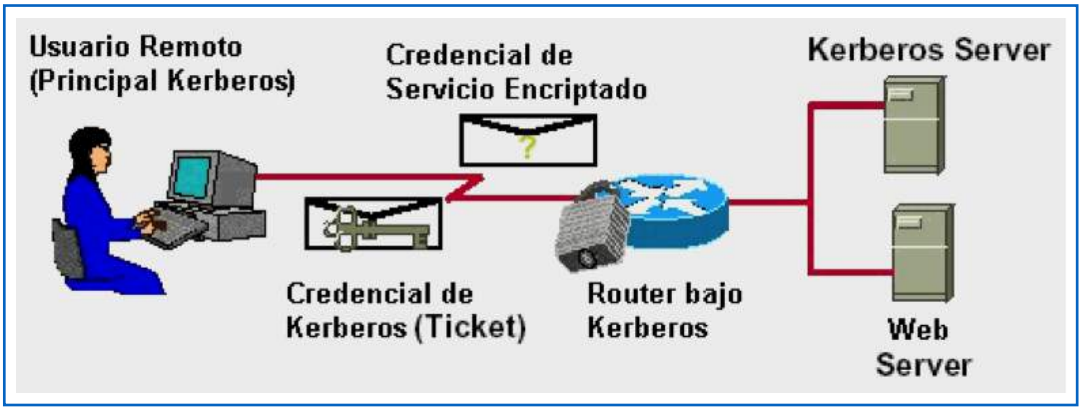


Imagen 3. Arquitectura de Kerberos
Fuente: http://www.monografias.com/trabajos12/rete/rete.shtml

La clave de sesión, Kerberos produce la clave secreta para el cliente, la cual utilizará con el servidor mientras realiza su sesión de trabajo.

El Ticket, o también denominado vale; este proporciona el control de acceso desde la estación de trabajo del usuario hacia el servidor. Se considera como un conjunto de datos electrónicos para identificar a un usuario o un servicio. Este ticket tiene como función la de garantizar que el cliente se ha autenticado en el servidor.

El autenticador, lo construye el cliente y lo envía al servidor, con el objeto de probar la identidad del usuario, así como para ver la actualidad de la comunicación, este objeto solo puede ser usado una sola vez.

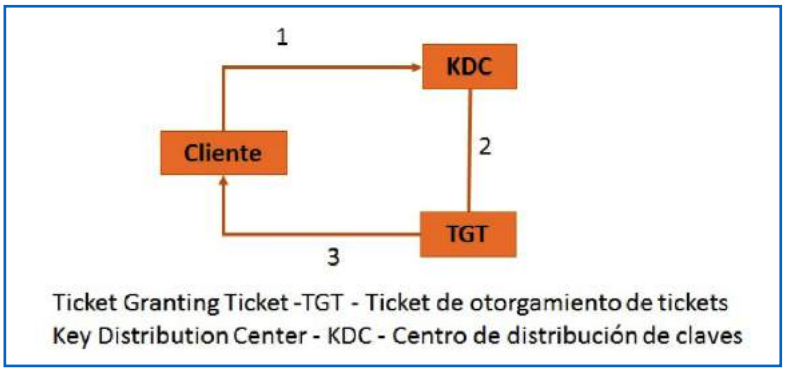


Imagen 4. Autenticación inicial de un cliente para establecer una sesión Kerberos
Fuente: Propia.

Cómo se autentica un usuario inicialmente

Para poderse identificar un usuario debe cumplir dos pasos, el primero consisten en autenticarse por primera vez para que luego pueda autenticarse por sí mismo. En la imagen 4 puede apreciarse esta autenticación inicial, en donde el usuario-cliente o servicio como el NFS inicia la sesión Kerberos pidiendo se le asigne un Ticket -TGT- desde el

Centro de Distribución de claves -KDC- esta petición es automática en el inicio de sesión. A continuación KDC crea el ticket para entregárselo al cliente y se lo envía en formato cifrado. El cliente descifra el ticket de otorgamiento de tickets con la contraseña del cliente. Una vez el cliente tiene el ticket válido, puede pedir otros tickets para realizar operaciones de red (operaciones como rlogin, telnet) mientras el periodo de otorgamiento de tickets en el servidor sea válido, pues generalmente solo dura unas horas, teniendo claro que si requiere hacer una operación de red única, solicitará al KDC un ticket para esa operación.

El segundo paso consiste en que si el usuario ya recibió la autenticación inicial, cada autenticación posterior sigue el patrón que se muestra en la imagen 5.

Cuando el cliente o usuario ha hecho su autenticación inicial, cada una de las autenticaciones siguientes, deben seguir los pasos que se muestran en la imagen 5.

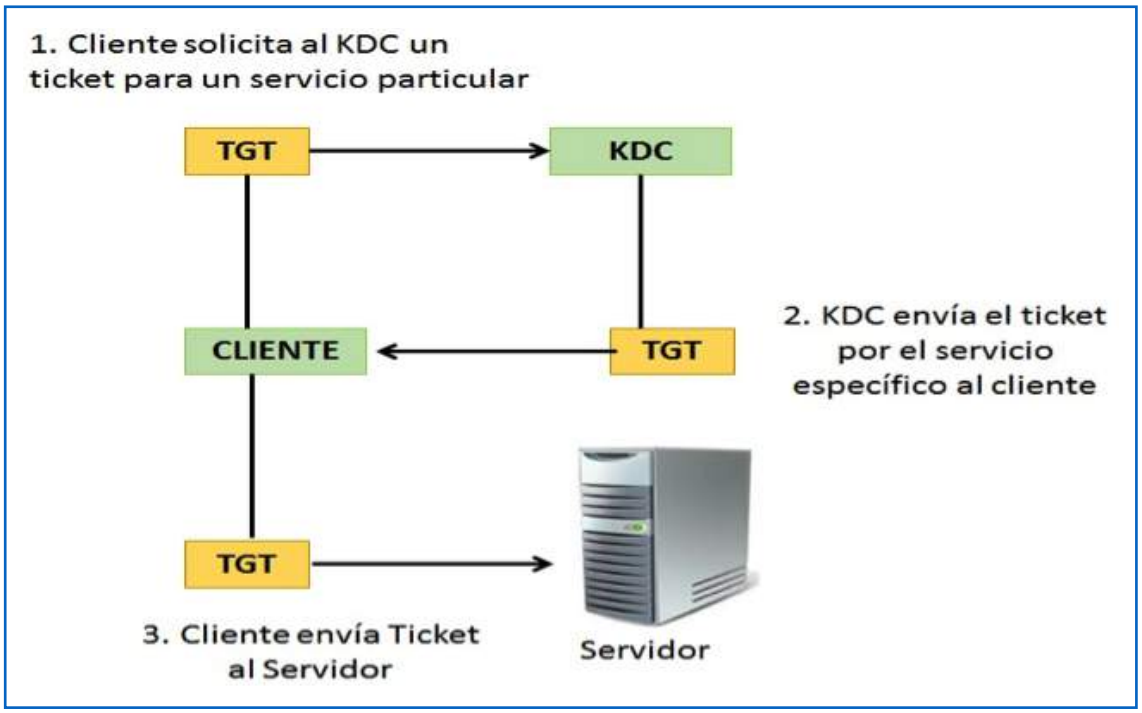


Imagen 5. Acceso a un servicio con la autenticación Kerberos
Fuente: Propia.

En la imagen anterior se aprecia cómo el usuario o cliente accede a un servicio con Kerberos, para ello, se siguen cuatro (4) pasos, en el primer paso, se hace la petición al KDC de un ticket para un servicio en particular, podría ser para iniciar sesión de manera remota en otra máquina, por lo tanto deberá enviar al KDC el ticket que le han otorgado como prueba de su identidad.

- En segunda medida, el KDC envía el ticket por el servicio específico al cliente.
- El tercer paso, consiste en que el cliente envía el ticket al servidor.
- El cuarto paso es cuando el servidor permite el acceso de clientes.

Cuando un cliente se une a un server por Kerberos de manera remota, los comandos que se utilizan son: ftp, rcp, rdist, rlogin, rsh, ssh, telnet. Los anteriores, son llamados comandos de usuario Kerberos.

Imagen 6. Ejemplo de dominios Kerberos
Fuente: Propia.

Los dominios de Kerberos (ver imagen 6) son redes lógicas las cuales se componen de un grupo de sistemas que tienen el mismo KDC principal. Los dominios pueden relacionarse entre sí y algunos de ellos, son jerárquicos, pero el tipo de jerarquía deberá ser definida y entre ellos deberán autenticarse; este proceso de Kerberos se llama Autenticación entre dominios.

De los dominios que se observan en la Figura 4, cada uno debe incluir un servidor que almacene una copia de las bases de datos del principal, a este servidor se le denomina servidor KDC maestro y cada uno de los dominios también deberá tener por lo menos un servidor KDC esclavo, el cual tendrá copias de la base de datos del principal. El servidor maestro y el esclavo han de crear tickets usados para la autenticación.

Aparte del servidor KDC maestro y esclavo, el dominio incluirá un servidor de aplicaciones Kerberos. En la imagen 7, se visualiza un ejemplo de dominio.

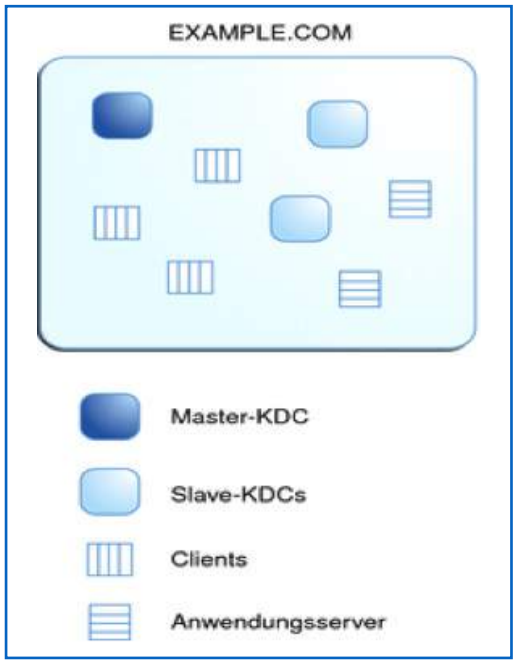


Imagen 7. Cómo puede ser un dominio Kerberos
Fuente: https://docs.oracle.com/cd/E24842_01/html/E23286/intro-25.html

Otras aplicaciones criptográficas

TrueCrypt

TrueCrypt es una aplicación informática freeware descontinuada que sirve para cifrar y ocultar datos que el usuario considere reservados empleando para ello diferentes algoritmos de cifrado como AES, Serpent y Twofish o una combinación de los mismos.

TrueCrypt es una aplicación que se utiliza en la codificación y decodificación de datos en tiempo real. Tiene un volumen normal, que permite el acceso por medio de una contraseña, luego el sistema mostrará un nuevo disco duro. El volumen oculto, accede a

la información tras digitar una clave y solo muestra el volumen oculto, el cual tendrá los archivos más importantes, como las claves de banco, claves de correos electrónicos documentos privados.

De otra parte, TrueCrypt sirve para crear y mantener un dispositivo de almacenamiento de datos cifrado; los datos se pueden cifrar al instante, de forma automática sin la intervención del humano y todo el sistema se cifra, desde los nombres de los archivos, carpetas, contenido, espacio libre, meta datos y demás.

Como se indicó en un párrafo anterior, los algoritmos de cifrado que soporta TrueCrypt son: AES, Serpent, Twofish.

Además utiliza diferentes combinaciones de los anteriores algoritmos como son:

- AES-Twofish.
- AES-Twofish-Serpent.
- Serpent-AES.
- Serpent-Twofish-AES.
- Twofish-Serpent.
- Algoritmos de hash, como RIPE-MD-160, SHA-512 y WHIRLPOOL.

La característica más notoria de TrueCrypt e que crea un área completa protegida dentro de la PC, allí se pueden almacenar los archivos asegurados, pues todo el volumen estará cifrado con una clave que el usuario mismo crea.

TrueCrypt, esta aplicación de código abierto está hecha para funcionar en sistemas operativos, Windows, Linux y Mac. Los discos virtuales cifrados que crea, se comportan como unidades de disco externos sin serlo; también se puede proteger todo el disco duro o las USB; el procedimiento anterior hace casi imposible la intrusión de extraños, además que la unidad cifrada funciona como si no lo estuviera

AxCrypt

Esta herramienta permite proteger la confidencialidad de los archivos. La codificación por medio de esta herramienta es de 128 bits, que hace que un documento esté totalmente protegido. Se utiliza para el envío de archivos por correo electrónico, teniendo la seguridad que sólo la persona interesada y receptora que conoce la contraseña podrá revisarlos.

AxCrypt es el software para cifrar archivos de código abierto para Windows. Se integra perfectamente con Windows para comprimir, cifrar, descifrar, almacenar, enviar y trabajar con archivos individuales.

Una característica importante es que AxCrypt tiene la opción de auto desencripción de archivos y reencripción automática, después de haberlo editado, elimina de forma total y segura de archivos, además valida la integridad de los datos y contraseñas criptográficas. Los archivos que cifra la herramienta tienen la extensión *.axx.

AxCrypt está integrado al navegador Internet Explorer, quien encripta, desencripta, visualiza y edita cualquier archivo. Esta herramienta usa los algoritmos AES-128 y SHA-1, quienes cumplen los requerimientos de seguridad de los Estados Unidos y de las normas, reglas y convenios de Internet.

AxCryt es una herramienta de software gratuita, de código abierto, que realiza cifrados de alta seguridad, y al ser de código abierto asegura que no hay puertas traseras cuando se cifran los archivos.

Las características más notables de AxCrypt son las siguientes:

- El cifrado de archivos de alta seguridad lo ofrece AxCryt sobre un número X de archivos.
- Se integra con el browser Internet Explorer.
- Con un solo doble clic sobre un archivo, es posible abrir, editar y guardar cualquier archivo.
- Es una herramienta muy sencilla, solo se instala y funciona, sin ningún tipo de configuración, además de poder descifrar archivos.
- Tiene la capacidad de cifrar archivos que se envían a través de Internet, como por ejemplo, a través del correo electrónico.
- Esta herramienta se encuentra en muchos idiomas, lo que hace que pueda ser usada casi por todo el mundo.

La página en la que puede descargarse gratuitamente es: http://www.axantum.com/axcrypt/

STunnel

STunnel es un programa que funciona con el protocolo de encripción SSL entre el cliente y el servidor. Al usar OpenSSL para la criptografía, lo hace compatible con los algoritmos criptográficos de la biblioteca.

Las principales características de STunnel:

En cuanto al rendimiento y escalabilidad, reparte carga entre varios servidores de backend, para grupos trabaja con la caché de sesión externo.

Sirve como soporte para las funciones OpenSSL, controlando el acceso basado en certificados, ofrece certificado de remoción CRL y OCSP, soporta servidores virtuales basados en nombre, llamados SNI-Server Name Indication; para el cumplimiento, utiliza el modo FIPS y configura los motores de hardware.

En las plataformas Windows utiliza una GUI-Interfaz Gráfica muy sencilla, ofrece la posibilidad de almacenar en caché cadenas de certificados de pares a los archivos y utiliza el modo servicio de Windows. Además de las características mencionadas, también redirecciona conexiones de cliente SSL cuando se presentan errores de autenticación, soporta IPv6.

OpenVPN

OpenVPN es una solución de software libre, creada por James Yonan (2001), está basado en SSL-Secure Sockets Layer y VPN - Virtual Private Network.

Una de las principales características es que OpenVPN proporciona conectividad entre máquinas conectadas a dos extremos (punto a punto), además permite validar jerárquicamente a los usuarios y a los host conectados de forma remota. Utilizada a menudo en redes inalámbricas o wifi.

La seguridad que ofrece OpenVPN se combina con el nivel empresarial, la simplificación en el uso y el cúmulo de características que posee. Además es una solución multiplataforma que proporciona un uso sencillo cuando se desea usar con configuraciones VPN's. Este tipo de soluciones se emplea por ejemplo para, acelerar los procesos de negocios, por cuanto se tendría intercambio rápido y flexible de la información, en el evento de comunicar toda una empresa con sus sucursales, además, si un empleado

requiere comunicarse con su oficina o con compañeros de trabajo, podrán intercambiar información de manera rápida. Lo anterior quiere decir que OpenVPN ofrece autenticidad, integridad, disponibilidad de la información a través de las redes.

Las VPN's surgen de aquella necesidad de conectar a través de Internet distintos usuarios. Es el caso de la utilización de estas redes VPN - Redes Virtuales y Privadas, que no son otra cosa que conexiones virtuales que mediante software pueden operar sobre Internet, la privacidad se encuentra, porque solo los autorizados pueden capturar o leer los datos que viajan por la red, de esta manera se logra seguridad usando los nuevos mecanismos de criptografía. Es así, como se observa en la ver imagen 9 que solo se requieren cuatro (4) conexiones a la red Internet en lugar de las seis (6) que se requerirían en una conexión dedicada (Ver imagen 8), en donde se tiene una red de cuatro nodos, los cuales quieren comunicarse entre ellos (todos con todos), por lo que se requieren seis (6) líneas para poderse comunicar.

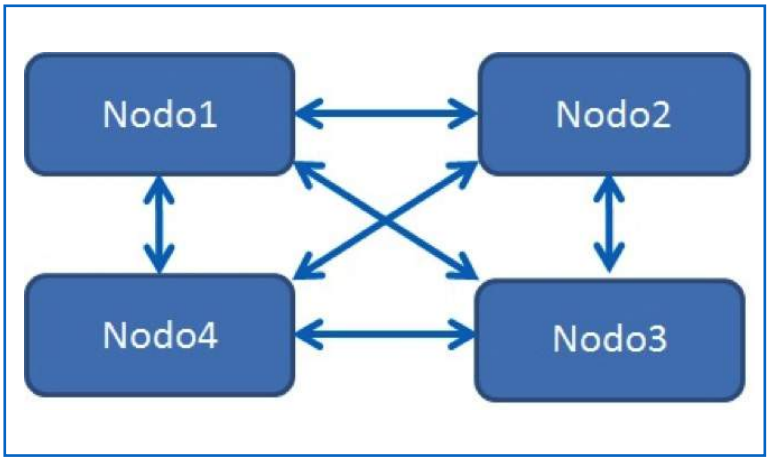


Imagen 8. Comunicación de nodos por redes dedicadas
Fuente: Propia.

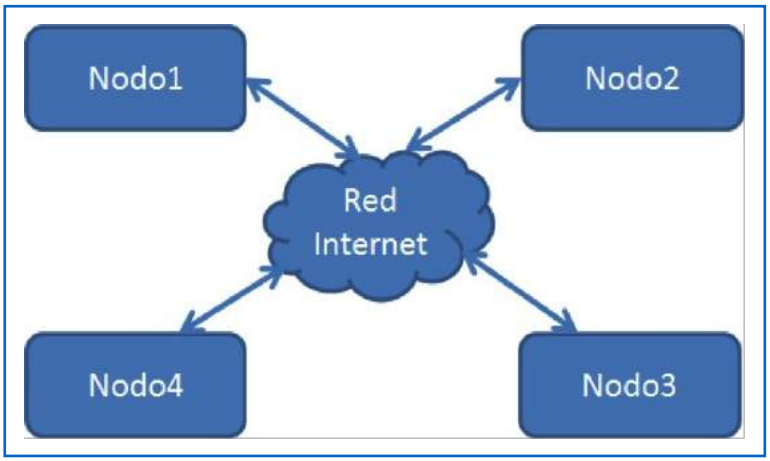


Imagen 9. Comunicación de nodos por red VPN
Fuente: Propia.

El uso que se les da a las redes VPN van desde conectar varios puntos de una empresa a través de Internet o conectar a usuarios corrientes con IP dinámica, pasando por soluciones que dan acceso a clientes de forma extrared con los cuales se necesita intercambiar información pero no deberán ingresar al resto de la red propia de la empresa. En este tipo de redes, la fiabilidad en la comunicación es excelente en los usuarios móviles.

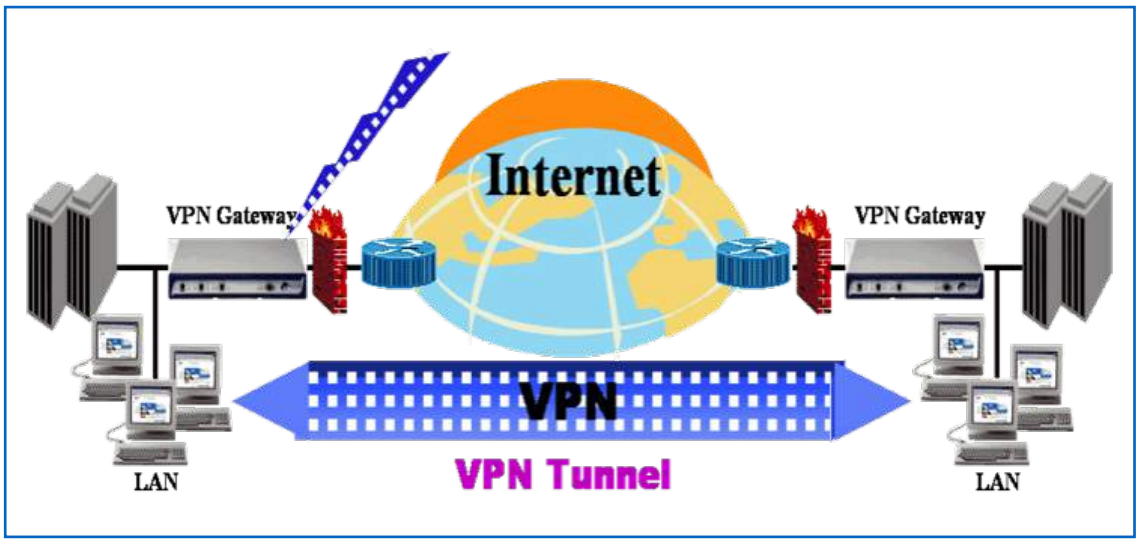


Imagen 10. Una red Virtual Private Network –VPN o Red privada Virtual
Fuente: http://www.telypc.com/vpn.html

En la Imagen 10 puede verse una red VPM, que se monta de la siguiente manera:

- El servidor VPN, es el que acepta peticiones de un cliente VPN.
- El cliente VPN, es el equipo que inicia la conexión hacia el servidor VPN, el equipo VPN cliente, puede ser un máquina individual o en su defecto un enrutador.
- El túnel, es una parte de la conexión sobre la cual se encapsula la información.
- Conexión VPN, en esta parte se cifran los datos, es así que cuando se utilizan conexiones VPN seguras, los datos se cifran y encapsulan en la misma parte de la conexión.

Los componentes de la red VPN, se pueden ver en la imagen 11, que consisten en el Servidor VPN, El Túnel, la conexión VPN, la red por donde transitan los dato y por supuesto el cliente VPN.

Es así que los protocolos de túnel, son usados para la administración de túnele, así como para encapsular los datos, los cuales también deberán estar cifrados, con ello se puede construir una conexión VPN.

En cuanto a los datos en el túnel, estos se envían por medio de una conexión punto a punto. En un ambiente VPN, existen redes compartidas o privadas, por donde viajan los datos encriptados y estas redes pueden ser una intranet basada en IP o la red internet.

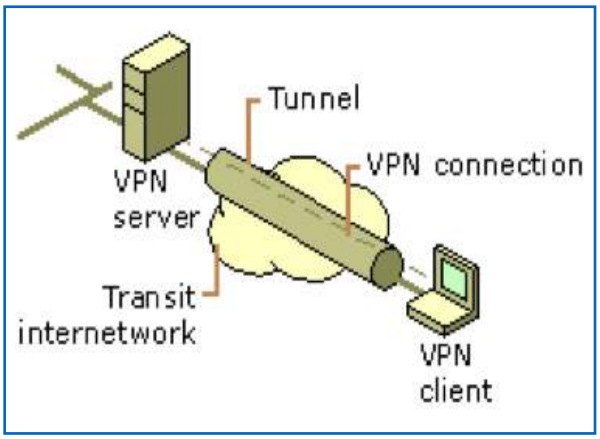


Imagen 11. Componentes de una red privada virtual
Fuente: https://msdn.microsoft.com/es-es/library/cc786563(v=ws.10).aspx

Una red VPN da la posibilidad de extender la red local sobre la red pública Internet, un claro ejemplo, es el de conectar dos o más sucursales de una misma empresa por medio de Internet o en su defecto que un usuario pueda comunicarse con su PC personal desde su trabajo a su casa.

Pero para poder ratificar las virtudes de una buena conexión, se debe garantizar la confidencialidad, la integridad y por supuesto la autenticación. En la autenticación o también llamada autorización se debe tener en cuenta quién está al otro lado de la red, qué usuario, en qué equipo y el nivel de acceso que tenga el usuario. En cuanto a la integridad, es la certeza de que los datos que se envían no sean alterados, para lograrlo, se utilizan las funciones Hash, como ejemplo sería el MD2, MD5 o el SHA (Secure Hash Algoritm).

En cuanto a la confidencialidad, los datos viajan por una red potencialmente insegura como lo es Internet, a la cual se le puede interceptar, por ello, la información deberá estar encriptada para que viaje de forma segura y no pueda ser leída o interpretada sino solo por el interesado y para ello, se podrá cifrar por medio de los algoritmos DES-Data Encryption Standar, o por el Triple DES-3DES o en su defecto por AES-Advanced Encrytion Standar.

Para complementar, OpenVPN es una solución de software del tipo cliente-servidor, que funciona en máquina Linux o Windows. Como se ve en las imagenes 10 y 11, el túnel que se forma entre el cliente VPN y el servidor VPN a través de Internet. De aquí que a OpenVPN se le denomine un modelo de seguridad basado en SSL, pues este es un estándar de seguridad utilizado para las comunicaciones seguras a través de internet.

El software OpenVPN es utilizado para crear las VPN que se basan en SSL, que permiten la conexión segura de oficinas remotas, además de ofrecer seguridad a clientes móviles que se sirven en redes privadas o LAN.

Algunas características del software OpenVPN son:

- Ofrece soluciones VPN a las empresas basadas en software libre.
- Se pueden diseñar túneles VPM, utilizados para conexiones punto a punto, así como para usuarios móviles.
- El medio de transporte que utiliza son los protocolos TCP o UDP.
- Utiliza un único puerto TCP o UDP con múltiples conexiones a una misma instancia.
- Al crear los túneles VPN estos funcionan sobre una Network Address Traslation - NAT, así como direcciones IP dinámicas.

- Para ofrecer comunicaciones seguras y con autenticación, se basa en los estándares SSL/TLS, para cumplir con su cometido utiliza todas sus características para el cifrado, autenticación y certificación, esto con el fin de preservar el tráfico que se hace por la red.
- El cifrado que utiliza es muy amplio, pues toma cualquiera de ellos, también el tamaño de la clave es cualquiera.
- El cifrado que utiliza OpenVPN es flexible tanto en tamaño como en el tipo de llaves estáticas, esto cuando se realiza cifrado convencional y si se hace cifrado asimétrico, utiliza llaves públicas que usan certificados x509.
- Las llaves estáticas que utiliza, pueden ser compartidas o dinámicas basadas en TLS cuando realiza intercambio de claves.
- Cuando se usan redes VPN, se puede entregar por medio del servidor DHCP que se integra a OpenVPN información como la dirección IP virtual dinámica o estática, la dirección de los servidores DNS, la dirección del gateway predeterminado y el servidor Wins.
- Integra el firewall con el fin de filtrar tráfico VPN.
- Ofrece soporte a los clientes cuando utilice sistemas operativos Linux, solaris, OpenBSD, Windows, Mac OSX, FreeBSD, entre otros.
- Las conexiones OpenVPN se hacen utilizando casi todos los firewall del mercado.

Ejercicio para realizar

Aprenda a utilizar la herramienta Vantir, utilizada para definir el tamaño del bloque a encriptar, el tamaño de la clave utilizada y el número de fases de encriptación. Con estas características se puede encriptar y desencriptar texto con uno de los siguientes Algoritmos: RC2, TripeDES, AES, DES.

Para lograr el cometido, ingrese a la dirección: http://www.vantir.com/Encriptar.aspx

Escoja todos los algoritmos criptográficos que allí aparecen en la margen izquierda, haga el ejercicio uno por uno: RC2, TripeDES, AES, DES. Escriba un texto claro o listo para ser encriptado. Pulse botón Encriptar. Notará que el texto se encripta y la herramienta le otorga una Llave-key y un campo IV, copie en algún archivo: la clave generada, el campo IV y el mensaje cifrado. Oprima el botón limpiar (le limpiará todos los campos). A continuación, copie el texto cifrado que figura en la parte inferior de la pantalla y péguelo en la parte superior. A continuación oprima el botón Desencriptar, luego de escribir la clave que le generó el sistema y llene el campo IV con la información guardada.

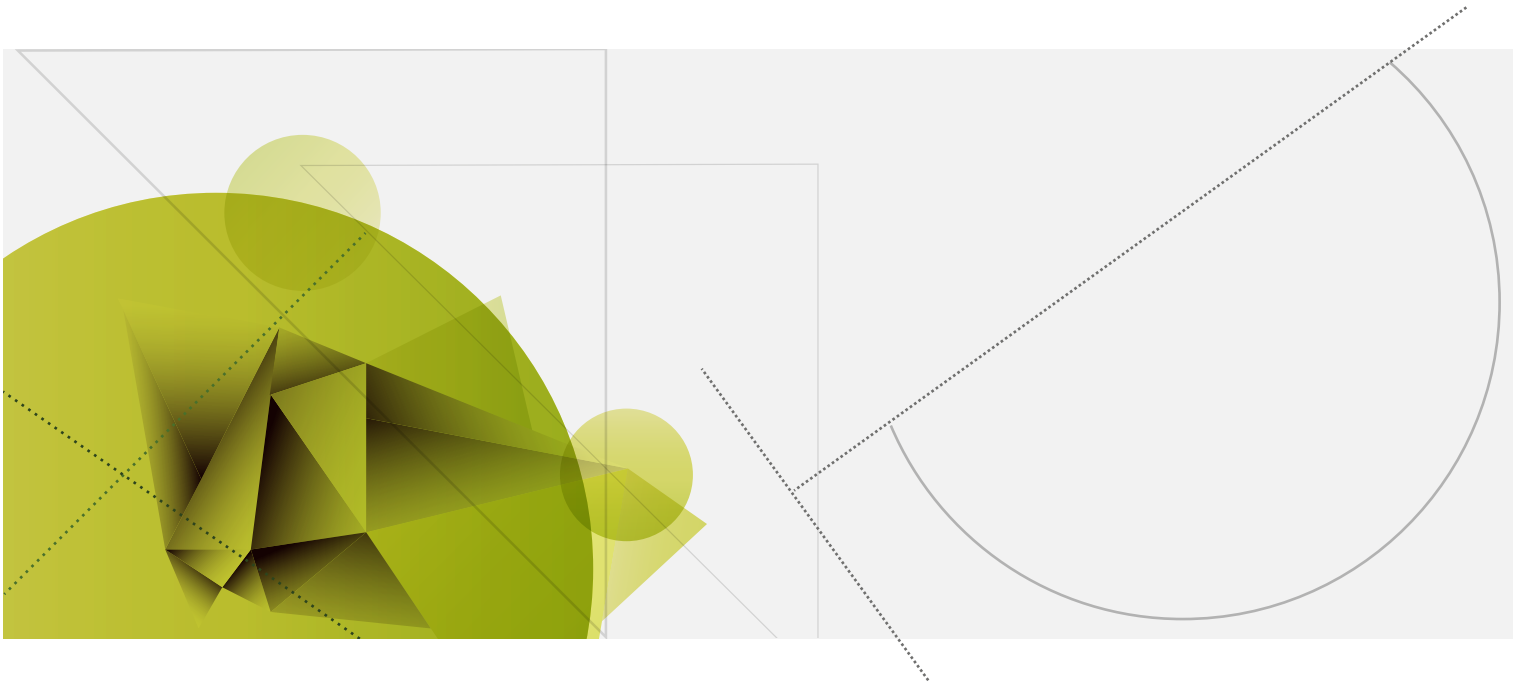
Bibliografia

- Alonso de armiño, A. (2014). Kerberos: un servicio de autenticación para redes. Dpto. de Informática y Estadística. Fac. De Economía Y Administración-Universidad Nacional Del Comahue. Buenos Aires: Argentina.
- Ángel, J. (s.f.). Criptografía para principiantes.
- Arteaga, A. (2014). El gamal. Criptografía de clave pública. Ver Link
- Ayuso, J. (2003). Protocolos criptográficos. Universidad de Valladolid. Ver Link
- Bart, P. (2014). Cryptographic Primitives for Information Authentication. Criptografia- - State of the Art. Katholieke Universiteit Leuven.
- Bejarano, K., Londoño, M., Amador, S. & Muñoz, L. (2014). Criptografía simétrica y asimétrica. Ver Link
- Caballero, P. (2002). Introducción a la criptografía. 2ª edición. RAMA Editorial.
- Chinchilla, E. (2008). Aplicaciones criptográficas. Ver Link
- Delgado, V. (2006). Introducción a la criptografía: Tipos de algoritmos. Anales de mecánica y Electricidad. Vol. 83.
- Durán, R., Hernández, L. & Muñoz, J. (2005). El criptosistema. RSA. Rama-editorial.
- Franco, J., Sarasa, M. &, Salazar, J. (1998). Criptografía digital: fundamentos y aplicaciones. Ed. Prensas Universitarias de Zaragoza.
- Fuster, A., De la Guia, D., Hernández, L. & Montoya, F. (2004). Técnicas.
- García, E., López, M. & Ortega, J. (s.f.). Una introducción a la criptografía. Ver Link
- Hernández, S. (s.f.). La Criptografía Clásica. Ver Link
- Huguet, L., Rifá, J. & Tena, J. (2013). Protocolos criptográficos. Universidad Oberta de Cataluña.
- Instituto Nacional de Tecnologías de la Comunicación. (2014). Esteganografía, el arte de ocultar información. Observatorio de la Seguridad de la Información.
- Jarauta, J., Palacios, R. & Sierra, J. (2008). Seguridad informática, criptografía asimétrica. Universidad Pontificia Comillas. Madrid: España.
- Kioskea.net. (2014). Criptografía -Secure Sockets Layers- SSL.
- Lucena, M. (2010). Criptografía y seguridad en computadores.
- Maiorano, A. (2010). Criptrografía. Técnicas de desarrollo para profesionales. RA-MA Editorial.

Bibliografia

- Massachusetts Institute of Technology-MIT. (2013). La criptografía - Escritura oculta.
- Orozco, G. (s.f.). Introducción a la criptografía. México.
- Pousa, A. (2011). Algoritmo de cifrado simétrico aes. Aceleración de tiempo de cómputo sobre arquitecturas Multicore. Universidad Nacional de la Plata. Argentina.
- Stallings, W. (2006). Fundamentos de seguridad en redes, aplicaciones y estándares. Pearson, Prentice Hall, Tercera Edición.
- Wanumen, L. (2006). Cómo crear herramientas seguras de desarrollo usando código seguro. Revista Vínculos. Universidad Distrital Francisco José de caldas. vol.3, No.1.

Esta obra se terminó de editar en el mes de noviembre
Tipografá Myriad Pro 12 puntos
Bogotá D.C,-Colombia.



AREANDINA
Fundación Universitaria del Área Andina

MIEMBRO DE LA RED
ILUMNO