ANTOLOGIA

Redes de Computadoras III

Licenciatura en Informática Administrativa

Quinto

Marco Estratégico de Referencia

ANTECEDENTES HISTORICOS

Nuestra Universidad tiene sus antecedentes de formación en el año de 1979 con el inicio de actividades de la normal de educadoras "Edgar Robledo Santiago", que en su momento marcó un nuevo rumbo para la educación de Comitán y del estado de Chiapas. Nuestra escuela fue fundada por el Profesor de Primaria Manuel Albores Salazar con la idea de traer Educación a Comitán, ya que esto representaba una forma de apoyar a muchas familias de la región para que siguieran estudiando.

En el año 1984 inicia actividades el CBTiS Moctezuma Ilhuicamina, que fue el primer bachillerato tecnológico particular del estado de Chiapas, manteniendo con esto la visión en grande de traer Educación a nuestro municipio, esta institución fue creada para que la gente que trabajaba por la mañana tuviera la opción de estudiar por las tarde.

La Maestra Martha Ruth Alcázar Mellanes es la madre de los tres integrantes de la familia Albores Alcázar que se fueron integrando poco a poco a la escuela formada por su padre, el Profesor Manuel Albores Salazar; Víctor Manuel Albores Alcázar en septiembre de 1996 como chofer de transporte escolar, Karla Fabiola Albores Alcázar se integró como Profesora en 1998, Martha Patricia Albores Alcázar en el departamento de finanzas en 1999.

En el año 2002, Víctor Manuel Albores Alcázar formó el Grupo Educativo Albores Alcázar S.C. para darle un nuevo rumbo y sentido empresarial al negocio familiar y en el año 2004 funda la Universidad Del Sureste.

La formación de nuestra Universidad se da principalmente porque en Comitán y en toda la región no existía una verdadera oferta Educativa, por lo que se veía urgente la creación de una institución de Educación superior, pero que estuviera a la altura de las exigencias de los jóvenes que tenían intención de seguir estudiando o de los profesionistas para seguir preparándose a través de estudios de posgrado.

Nuestra Universidad inició sus actividades el 18 de agosto del 2004 en las instalaciones de la 4ª avenida oriente sur no. 24, con la licenciatura en Puericultura, contando con dos grupos de cuarenta alumnos cada uno. En el año 2005 nos trasladamos a nuestras propias instalaciones en la carretera Comitán – Tzimol km.

57 donde actualmente se encuentra el campus Comitán y el Corporativo UDS, este último, es el encargado de estandarizar y controlar todos los procesos operativos y Educativos de los diferentes Campus, Sedes y Centros de Enlace Educativo, así como de crear los diferentes planes estratégicos de expansión de la marca a nivel nacional e internacional.

Nuestra Universidad inició sus actividades el 18 de agosto del 2004 en las instalaciones de la 4ª avenida oriente sur no. 24, con la licenciatura en Puericultura, contando con dos grupos de cuarenta alumnos cada uno. En el año 2005 nos trasladamos a nuestras propias instalaciones en la carretera Comitán – Tzimol km. 57 donde actualmente se encuentra el campus Comitán y el corporativo UDS, este último, es el encargado de estandarizar y controlar todos los procesos operativos y educativos de los diferentes campus, así como de crear los diferentes planes estratégicos de expansión de la marca.

MISIÓN

Satisfacer la necesidad de Educación que promueva el espíritu emprendedor, aplicando altos estándares de calidad Académica, que propicien el desarrollo de nuestros alumnos, Profesores, colaboradores y la sociedad, a través de la incorporación de tecnologías en el proceso de enseñanza-aprendizaje.

VISIÓN

Ser la mejor oferta académica en cada región de influencia, y a través de nuestra Plataforma Virtual tener una cobertura Global, con un crecimiento sostenible y las ofertas académicas innovadoras con pertinencia para la sociedad.

VALORES

- Disciplina
- Honestidad
- Equidad
- Libertad

ESCUDO

El escudo de la UDS, está constituido por tres líneas curvas que nacen de izquierda a derecha formando los escalones al éxito. En la parte superior está situado un cuadro motivo de la abstracción de la forma de un libro abierto.

ESLOGAN

"Mi Universidad"

ALBORES

Es nuestra mascota, un Jaguar. Su piel es negra y se distingue por ser líder, trabaja en equipo y obtiene lo que desea. El ímpetu, extremo valor y fortaleza son los rasgos que distinguen.

REDES DE COMPUTADORAS.

Objetivo de la materia:

Esta asignatura aporta al perfil profesional, la capacidad para desarrollar aplicaciones en un lenguaje de programación de alto nivel, para la solución de problemas relacionados con las diferentes disciplinas en el área.

Criterios de evaluación:

No	Concepto	Porcentaje
1	Actividad uno en plataforma	20%
2	Actividad dos en plataforma	20%
3	Examen	60%
4	Total	100%
5	Escala de calificación	7- 10
6	Mínima aprobatoria	7

INDICE

UNIDAD III

ALGORITMOS DE CLAVE PÚBLICA Y FIRMAS DIGITALES

3.1. El algoritmo RSA

3.2. Otros algoritmos de clave pública

3.3. Firma de claves simétricas

3.4. Firmas de claves públicas

3.5. Compendios de mensajes

3.6. El ataque de cumpleaños

3.7. Funciones de dispersión

3.8. Firmas digitales

3.9. Certificados digitales

3.10. Listas de anulación de Certificados

3.11. Infraestructura de Clave

3.12. Protocolo SSL

3.13. Aplicaciones e implementaciones

3.14. Modelos de seguridad

3.15. Dominios protegidos

UNIDAD IV

PROTOCOLOS DE AUTENTICACIÓN

4.1. Autenticación basada en una clave secreta compartida

4.2. Establecimiento de una clave compartida: el intercambio de claves de Diffie-Hellman

4.3. Autenticación que utiliza un centro de distribución de claves

4.4. Autenticación utilizando Kerberos

4.5. Autenticación utilizando criptografía

4.6. Métodos de autenticación

4.7. TLS, PEAP

4.8. Configuraciones

4.9. Servidores

4-10. Raíz de Confianza

4.11. Reconexión rápida

UNIDAD I
CRIPTOGRAFÍA

1.1.- INTRODUCCIÓN A LA CRIPTOGRAFÍA

Por Naturaleza el ser humano es curioso, y, por ende, deseoso de conocer lo desconocido, aunque, en algunos casos lo desconocido es de carácter privado y confidencial, ocasionando entonces molestias por parte del poseedor, dueño o custodio de aquella información considerada de carácter reservado, personal u organizacional.

Así, desde el principio hace miles de años, los seres humanos se han visto en la necesidad de ocultar toda aquella información que sea considerada privada, a fin de resguardar y mantenerla a salvo de intrusos que pudieran hacer un mal uso de ella si lo conocieran. Y con deseo de esconder la información valiosa para su poseedor es como nace la criptología.

Del griego Kryptós, criptos "ocultar" y graphé, grafos "escribir"

La criptografía es: escritura oculta.

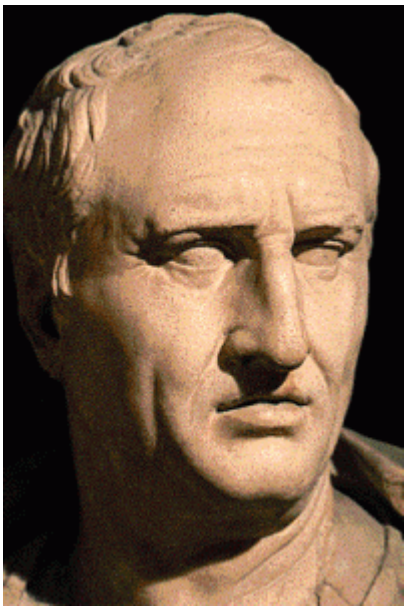
Y se refiere al arte de escribir mensajes en clave secreta o en forma enigmática, así, en nuestros días, criptografía es la ciencia encargada de transformar la información de manera tal que ésta quede descubierta y sea incompresible para todo aquel que no tenga la autorización correspondiente para acceder a ella. Sus principales objetivos son resguardar la confidencialidad de la información y la autenticidad del par remitente/emisor; y a todos aquellos que se decidan a estudiar y desarrollar métodos para resguardar así la información se les llama criptógrafos.

A continuación, se presentan los momentos más importantes en la historia de la criptografía.

-En el siglo V a.C. durante la guerra entre Atenas y Esparta, se encuentra el primer registro formal del uso de escritura secreta, en el 400 a.C. los espartanos utilizaron la Scítala o Escítalo, que puede considerarse el primer sistema de criptografía por transposición.

El mensaje solo podía leerse cuando éste se enrollaba sobre un bastón del mismo largo y grosor, que poseía un destinatario lícito.

-En el siglo II a.C. Polybio (203 a.C – 118 a.C), historiados griego miembro de las clases gobernantes que vivió en carne propia los acontecimientos tanto políticos como militares de su época, le permiten escribir la historia de tal manera que le convierten en uno de los historiadores más prestigiosos de la antigüedad.



Las características principales de su pensamiento fueron el cuidado y la veracidad que lo llevaron a inventar un cuadro de 5×5 que permitía intercambiar los distintos signos entre si y fue tan importante en su época que se utilizó en muchos sistemas criptográficos posteriores.

Su cifrador es como el que aparece en la imagen, con la peculiaridad de que éste se ha adaptado al español, ya que definitivamente era otro el alfabeto que en aquel entonces se

empleaba. Algunos historiadores lo han adaptado al inglés y algunos prefieren utilizar números, pero la idea es la misma.

-En el siglo I a.C surge el cifrado César, el cual se considera que fue utilizado por Julio César (101 a.C – 43 a.C), de ahí al nombre que se le ha dado el nombre al cifrado, aun cuando hay algunos historiadores que indican que éste nunca lo utilizó directamente, pero que fue utilizado por otros emperadores romanos de la época. El cifrado consiste en mover el carácter a representar 3 posiciones adelante dentro del alfabeto a utilizar.

-En el siglo I d.C Carlomagno luchó contra diversos pueblos obteniendo la victoria, fusionando las culturas germánicas, romana y cristiana, y estableciendo su dominio en la mayor parte de Europa central y occidental. Carlomagno es importante no sólo por el número de victorias y la dimensión de su imperio en la que sin duda fue un factor determinante al que en a comunicación que mantenía con sus ejércitos y aliados sustituía las letras por símbolos extraños, de manera que sus textos secretos los escribía de forma cifrada; sino además por sus valores.

-En el siglo XI el Papa Urbano II (1088-1099) pronuncia un sermón en el que invita a la lucha y a la guerra santa, se cuenta que los caballeros que oyeron la exhortación papal cortaron unos paños rojos en forma de cruz y se los colgaron en el pecho como signo de que querían participar en la expedición, de manera que a los guerreros que pelearon esas batallas se les llamó cruzados, y a las campañas siguientes "cruzadas", algo relevante en cuanto a criptografía, es que los mensajes iban ocultos en el cuero cabelludo.

-En el siglo XV se escribe la que se considera por muchos cómo la primera y más antigua obra que existe sobre criptografía, "Liber Zifrorum", escrita por Cicco Simoneta (1410-1480) consejero y secretario de la cancillería de los duques Sforza en Milán, en la que se esudian diversos sistemas basados en la sustitución de letras y diversas representaciones en las que incluyen símbolos convencionales.

Hacia 1466, Alberti escribe otra obra "De Compendis Cifris" y concibe el sistema poli alfabético, esto es, un cifrador que emplea varios alfabetos, saltando de uno a otro cada tres o cuatro palabras.

-En el siglo XVI, eso es, un siglo después, Giovan Battista Belasco de Brescia instituyó una nueva técnica. La clave formada por una palabra o frase, debía transcribirse letra a letra sobre el tecto original y cada letra del texto se debía cambiar por la correspondiente en el alfabeto que comienza en la letra clave.

-Dentro del período renacentista italiano, otros estudios de la criptografía también hicieron contribuciones importantes, como Gerolamo Cardano también conocido como Jérome Cardan (1501-1576) nacido en Pavía y quien fue un célebre matemático físico y astrólogo. En el campo de la criptología aportó en 1550 un sistema basado en una carta o tarjeta con agujeros perforados, de tal manera que el mensaje en claro se obtenía al colocarlo sobre determinado texto preconcebido, lo que sería llamado después "Mascaras Rotativas".

Es así como enel iglo XVI se generaliza el uso de la criptología en los ambientes diplomáticos y para 1586 Blaise Vigenere (1523-1596) diplomático, criptógrafo y químico francés , secretario de la cámara del rey Enrique III de Francia, publica una obra titulada Traicté des Chiffers, donde recoge diferentes métodos utilizados en la época y en la que describe a detalle el cifrado polialfabética desarrollado por Battista, motivo por el cual se le otorgó erróneamente el cifrado de Vigenere, siendo el primer cifrado y robusto, difícil de romper.

En el siglo XVIII, con toda esta serie de acontecimientos, el uso de la criptografía se extendió en todos los ambientes donde la información se encontraba vinculada con todo aquello que representa el poder, esto es, se relaciona directamente con secretos de estado, asuntos militares, de espionaje y diplomáticos, y siempre rodeada de de todo lo que representa misterio, así, a finales del siglo XVIII y principios del XIX nace el primer dispositivo mecánico conocido como la rueda de Jefferson, la cual tiene sus ideas fundamentales en lo estudiado por Alberti y Polybios y que más tarde se le conocería como disco Wheatstone. Wheatstone llamó a sudesarrollo "cifrador Playfair en honor a su amigo Lord Playfair.

En 1883, el Dr. Auguste Kerckhoffs (enero de 1835 – agosto 1903), lingüista y criptógrafo francés quien rompe el sistema llamado "gran cifrador" creado por Rossignols y descifra mensajes cifrados con el sistema de transposición militar francés, lo que hace que el ministro de guerra decida cambiar el esquema de cifrado por uno nuevo. El cilindro Bazeries.

El método de Vigenére fue llevado hasta su última extensión lógica muchos años más tarde por un criptógrafo américano, el ingeniero Gilbert Sandford Vernam, quien lo demostró que para el cifrado de Vigenére fuera seguro no solamente era necesario que la clave de cifrado fuera mpas larga que el mensaje, sino que además ésta debía ser utilizada una sola vez.

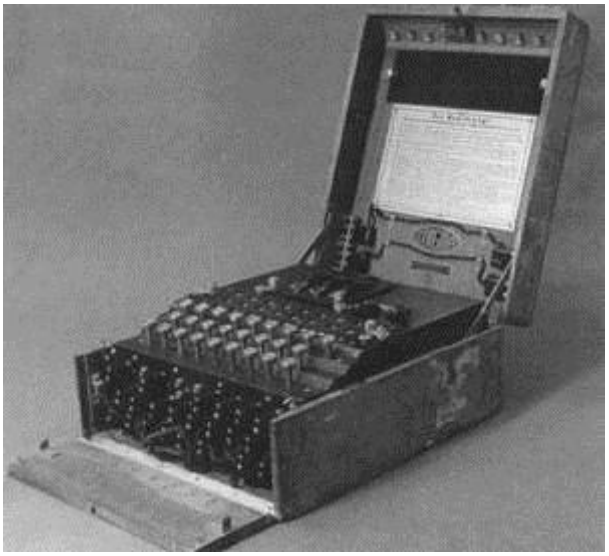
Para 1919 se registra la primer patente de una máquina criptográfica, la cual corresponde a una máquina llamada Enigma, obra del holandés Alexander Koch y el alemán Arthur

Scherbius, este último realizó varias versiones de Enigma junto con Richard Ritter, y conjuntamente fundaron en Berlín la compañía Chiffriermaschinen Aktien Gesellschaft para llevar a cabo la producción comercial de la máquina. Así la primera versión comercial fue puesta en venta en 1923 y se llamó Enigma-A.

A esta primera versión le siguieron tres modelos comerciales Enigma B, C y D, siendo la última la más importante, ya que tuvo un éxito rotundo después de haber sido adquirida en 1926 por la marina alemána. Aunque no fue hasta 1929 que el ejército alemán comenzó a usar el modelo básico, el cual se extendió a la totalidad de las organizaciones militares alemanas y la jerarquía nazi.

Cabe mencionar que la existencia de Enigma y el hecho de que los aliados conocieran sus secretos, fueron durante mucho tiempo, dos de los secretos mejor guardados de la II Guerra Mundial, tal vez porque de esta forma era posible seguir sacándole partido tras a guerra, potenciando su uso en diversos países, que, al instalarla hacían transparentes sus secretos. Y tras la inclusión de la II Guerra Mundial, la criptografía tiene un desarrollo teórico importante; siendo Claude Elwood Shannon (1916-2001) y su investigación sobre teoría de la información, escenciales en dicho desarrollo.

Claude Shannon, ingeniero electricista y matemático, profesor e investigador en el MIT y colaborador de los laboratorios Bell, estuvo siempre interesado en el álgebra boolenada y los circuitos digitales. En 1938 publicó cómo álgebras digitales, trabajo que desarrolló en su tesis de maestria y por el cual obtuvo el premio Nobel a jóvenes ingenieros en 1940.



La criptografía, en términos sencillos es la ciencia que se basa en la escritura de códigos y cifrados para proteger las comunicaciones, es uno de los elementos más importantes que hacen posible la existencia de criptomonedas modernas y blockchains. Las técnicas criptográficas que se utilizan hoy en día, sin embargo, son el resultado de una historia de desarrollo increíblemente larga. Desde tiempos antiguos, la gente ha empleado la criptografía para transmitir información de una forma segura. A continuación, presentamos la fascinante historia de la criptografía que ha conducido hasta los avanzados y sofisticados métodos utilizados en la encriptación digital moderna.

Los Antiguos Orígenes de la Criptografía

Se conoce la existencia de técnicas criptográficas primitivas desde tiempos remotos, ya que la mayoría de civilizaciones antiguas parecen haberlas usado de una forma u otra. El reemplazo de símbolos, la forma más básica de criptografía, se puede encontrar tanto en antiguas escrituras mesopotámicas como egipcias. El ejemplo más antiguo conocido de esta forma de criptografía se encontró en la tumba de un noble egipcio llamado Khnumhotep II, que vivió hace aproximadamente unos 3.900 años.

El propósito del reemplazo de símbolos en la inscripción de Knhumhotep no era ocultar información, sino incrementar su "atractivo lingüístico". El caso más antiguo conocido de criptografía enfocada a proteger información sensible, es el de un escriba mesopotámico

de hace 3.500 años que empleó la técnica para ocultar una fórmula para glaseado de cerámica en una tableta de arcilla.

En periodos posteriores de la antigüedad, la criptografía sería ampliamente utilizada para la protección de importantes informaciones militares, una función que aún hoy en día cumple. En la ciudad-estado griega de Esparta, los mensajes se encriptaban al ser escritos en un pergamino colocado en un cilindro de una medida particular, lo que hacía que el mensaje fuera indescifrable hasta que el recipiente lo enrollaba en un cilindro similar. De forma parecida, se sabe que los espías de la antigua India empleaban mensajes codificados ya en el siglo II a.C.

Probablemente, la criptografía más avanzada del mundo antiguo fue la de los romanos. Un ejemplo destacado de criptografía romana, conocida como el cifrado del César, consistía en cambiar las letras de un mensaje encriptado en base a cierto número de posiciones en el alfabeto latino. Si se conocía el sistema y el número de posiciones que debían moverse las letras, cualquier recipiente podía decodificar con éxito el otrora ilegible mensaje.

En la Edad Media y Renacimiento

A lo largo de la Edad Media, la criptografía se volvería cada vez más importante, pero los cifrados por substitución -de los cuales el cifrado del César es un ejemplo- seguirían siendo el estándar. El criptoanálisis, la ciencia encargada de resolver códigos y cifrados, empezó a ponerse al nivel de una todavía relativamente primitiva ciencia criptográfica. Al-Kindi, un célebre matemático árabe, desarrollaría en torno al 800 d.C. una técnica conocida como análisis de frecuencia, que dejaba en situación de vulnerabilidad a los cifrados por sustitución. Por primera vez, la gente que intentaba descifrar mensajes encriptados tenía a su disposición un método sistemático para lograrlo, lo que obligó a la criptografía a evolucionar para seguir siendo útil.

En 1465, Leone Alberti desarrolló el cifrado poli alfabético, considerado la solución contra la técnica de análisis de frecuencia de Al-Kindi. En un cifrado polialfabético, el mensaje se codifica utilizando dos alfabetos distintos. Uno es el alfabeto en que el mensaje original se escribe, mientras el segundo es un alfabeto enteramente diferente, en el que el mensaje se

muestra después de ser codificado. En combinación con los cifrados de sustitución tradicionales, los cifrados poli alfabéticos incrementaban enormemente la seguridad de la información codificada. A no ser que el lector conociera el alfabeto en que el mensaje había sido originalmente escrito, el análisis de frecuencia resultaba inútil.

Nuevos métodos para codificar información serían también desarrollados durante el Renacimiento, incluyendo un temprano método popular de codificación binario inventado en 1623 por el célebre erudito Sir Francis Bacon.

Avances en Siglos Más Recientes

La ciencia criptográfica continuaría progresando en los siguientes siglos. Un notable avance en criptografía sería descrito, pero quizás nunca construido, por Thomas Jefferson en la década de 1790. Su invento, conocido como rueda de cifrado, consistía en 36 anillos de letras en ruedas móviles, que podían ser utilizados para lograr codificados complejos. Este concepto era tan avanzado que serviría como base de la criptografía militar americana hasta el periodo de la Segunda Guerra Mundial.

La Segunda Guerra Mundial traería consigo el ejemplo perfecto de criptografía analógica: la máquina Enigma. Igual que la rueda de cifrado, este dispositivo, empleado por las potencias del Eje, utilizaba ruedas rotatorias para codificar un mensaje -haciendo que fuera virtualmente imposible leerlo sin otra máquina Enigma. Tempranas formas de tecnología informática serían empleadas para eventualmente ayudar a romper el cifrado de Enigma. El exitoso desencriptado de los mensajes de Enigma aún se considera un componente crítico de la posterior victoria aliada.

La Criptografía en la Edad de las Computadoras

Con el auge de las computadoras, la criptografía alcanzó niveles de progreso mucho mayores que en la era analógica. La encriptación matemática de 128-bits, mucho más fuerte que cualquier cifrado antiguo o medieval, es ahora el estándar para muchos dispositivos sensibles y sistemas informáticos. En 1990, se pondría en marcha toda una nueva forma de criptografía, apodada criptografía cuántica, por parte de científicos

computacionales que esperaban elevar una vez más el nivel de protección ofrecido por la encriptación moderna.

Más recientemente, técnicas criptográficas han sido también utilizadas para hacer posibles las criptomonedas. Las criptomonedas aprovechan varias técnicas criptográficas avanzadas, como las funciones hash, la criptografía de clave pública y las firmas digitales. Estas técnicas se utilizan principalmente para garantizar la seguridad de los datos almacenados en blockchains y para autenticar las transacciones. Una forma especializada de criptografía, llamada Elliptic Curve Digital Signature Algorithm (ECDSA), sirve de puntal a Bitcoin y a otros sistemas de criptomonedas, al proporcionar una seguridad complementaria y garantizar que los fondos sólo pueden ser utilizados por sus legítimos dueños.

La criptografía ha recorrido un largo camino en los últimos 4.000 años, y no parece que vaya a detenerse pronto. En la medida en que información sensible siga requiriendo protección, la criptografía continuará avanzando. A pesar de que los sistemas criptográficos empleados actualmente en las blockchains de las criptomonedas representan algunas de las formas más avanzadas de esta ciencia, son también piezas de una tradición que abarca buena parte de la historia humana.

1.2.- CIFRADOS POR SUSTITUCIÓN.

El **cifrado César** es uno de los primeros métodos de cifrado conocidos históricamente. Julio César lo usó para enviar órdenes a sus generales en los campos de batalla. Consistía en escribir el mensaje con un alfabeto que estaba formado por las letras del alfabeto latino normal desplazadas tres posiciones a la derecha. Con nuestro alfabeto el sistema quedaría así:

Alfabeto en claro:	A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
Alfabeto	D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z A B C

cifrado:	

Por ejemplo, si se quiere enviar el mensaje ATACARALAMANECER, lo que se escribirá realmente es DWDFDUDÑDODPHFHU

El receptor del mensaje conocía la clave secreta de éste (es decir, que estaba escrito con un alfabeto desplazado **tres** posiciones a la derecha), y podía descifrarlo fácilmente haciendo el desplazamiento inverso con cada letra del mensaje. Pero para el resto de la gente que pudiese accidentalmente llegar a ver el mensaje, el texto carecía de ningún sentido.

Aparentemente es un cifrado muy débil y poco seguro, pero en la época de Julio César no era de conocimiento general la idea de ocultar el significado de un texto mediante cifrado. De hecho, que un mensaje estuviese por escrito ya era un modo de asegurar la confidencialidad frente a la mayoría de la población analfabeta de la época.

Como dato curioso, más de 1500 años después, un cifrado similar al de César fue utilizado por la reina María Estuardo de Escocia, para conspirar junto con los españoles contra su prima Isabel I (en realidad, fue incitada a conspirar por agentes al servicio de Isabel I; una trampa bien urdida.) Los mensajes cifrados de María fueron fácilmente descifrados mediante sencillos análisis estadísticos por los agentes de Isabel I, y así pues quedó al descubierto la conspiración de la reina escocesa. Junto con la pérdida del secreto de la comunicación, María perdió la cabeza en su ejecución el 8 de febrero de 1587. Después de esto el cifrado César quedó definitivamente descartado como método de cifrado seguro para los gobernantes del mundo. Desde entonces a hoy, los cifrados usados por los estados para preservar sus secretos han mejorado considerablemente.

Lo que a nosotros nos interesa del cifrado César es que es un claro ejemplo de utilización de la aritmética modular para garantizar la confidencialidad de la información mediante el cifrado o encriptación. Matemáticamente, podemos describir el método usado por Julio César como una función lineal del tipo

$$E(x)=x+3 \pmod{27}$$

para un alfabeto con 27 caracteres como el español. La **x** indica la posición que la letra "en claro" ocupa en alfabeto. **E(x)** indica la posición de la letra cifrada correspondiente a **x** en el alfabeto. Según esto, E(0)=3, y E(26)=2 (esto es, la a se cifra como d, y la z como c)

Para descifrar se emplea la función **D(x)=x-3 (mod 27)** Para cifrar y descifrar el mensaje los comunicantes han de conocer y usar una misma clave secreta, que en este caso es el desplazamiento aplicado sobre el alfabeto (desplazamiento=3). Por eso el cifrado César pertenece a los cifrados de **clave privada**, también llamados **cifrados simétricos**.

Cifrados por sustitución mono alfabeto

Una sustitución mono alfabeto como la del cifrado César puede expresarse mediante una transformación congruente lineal (también conocida criptográficamente como **transformación afín**). En el cifrado César esta se escribiría como **E(M)= (M+3) mod N**, siendo N la longitud o cardinal del alfabeto original.

Puede extenderse la transformación afín a un caso más general con la siguiente congruencia lineal:

$$E_{(a,b)}(M) = (aM + b) \bmod N$$

siendo **M** el valor numérico de un carácter del alfabeto original, **a** y **b** dos números enteros menores que el cardinal **N** del alfabeto, y cumpliendo que a y N sean primos entre sí, esto es, que mcd(a,N) = 1, ya que de no ser así diferentes letras del alfabeto original darían lugar a una misma letra en el alfabeto cifrado equivalente. La clave de cifrado k viene entonces dada por el par (**a,b**).

a es una constante que determina el intervalo de separación entre dos letras del alfabeto cifrado cuando estas son consecutivas en el alfabeto original. Esta constante se

denomina coeficiente o factor de decimación. **b** es una constante que determina el desplazamiento entre las letras del mensaje claro y las correspondientes en el cifrado.

El cifrado César sería pues una transformación afín con una clave k = (1,3).

Criptoanálisis de los Métodos de Cifrado Monoalfabéticos

El cifrado monoalfabético constituye la familia de métodos criptográficos más simple de criptoanalizar, puesto que las propiedades estadísticas del texto claro se conservan en el criptograma. Supongamos que, por ejemplo, la letra que más aparece en Castellano es la E. Parece lógico que la letra más frecuente en el texto codificado sea aquella que corresponde con la E. Emparejando las frecuencias relativas de aparición de cada símbolo en el mensaje cifrado con el histograma de frecuencias del idioma en el que se supone está el texto claro, podremos averiguar fácilmente la clave.

Distribución de frecuencias de letras en español para un texto literario

E - 16,78%	R - 4,94%	Y - 1,54%	J - 0,30%
A - 11,96%	U - 4,80%	Q - 1,53%	Ñ - 0,29%
O - 8,69%	I - 4,15%	B - 0,92%	Z - 0,15%
L - 8,37%	T - 3,31%	H - 0,89%	X - 0,06%
S - 7,88%	C - 2,92%	G - 0,73%	K - 0,00%
N - 7,01%	P - 2,77%	F - 0,52%	W - 0,00%
D - 6,87%	M - 2,12%	V - 0,39%	-

Cifrados de sustitución poli alfabeto

Como ya se vio en el apartado dedicado a los criptosistemas monoalfabéticos, su principal debilidad es que el texto cifrado mantiene la misma distribución de frecuencia de caracteres que tiene el texto claro original, lo que hace que los cifrados mono alfabeto sean criptoanalizables por métodos estadísticos sencillos. Una posible mejora de los cifrados por sustitución es intentar métodos que destruyan esa correspondencia de frecuencias entre el mensaje en claro y el criptograma. Por ejemplo, utilizando varios alfabetos a la vez para el cifrado.

En los cifrados polis alfabéticos la sustitución aplicada a cada carácter varía en función de la posición que ocupe este dentro del texto claro. En realidad, corresponde a una aplicación cíclica de n cifrados de sustitución mono alfabeto.

Cifrado de Vigenère

Es un ejemplo típico de cifrado poli alfabético cuya invención fue imputada erróneamente a Blaise de Vigenère, y que data del siglo XVI. La clave está constituida por una secuencia de símbolos del alfabeto K = {k_0, k1, ..., kd_{-1}}, de longitud **d**, y que emplea la siguiente transformación congruente lineal de cifrado:

$$E_k\ (m_i) = m_i + k_{(i \bmod d)} \pmod n$$

siendo m_i el i-ésimo símbolo del texto claro y n el cardinal (longitud) del alfabeto de entrada. Como clave se puede utilizar cualquier palabra de una longitud por ejemplo entre 6 y 8 caracteres, que no tenga letras repetidas.

Para ver mejor esto, supongamos que, con nuestro alfabeto español de 27 símbolos, queremos cifrar el texto en claro **"PLAN"**, y que para el cifrado utilizamos como clave la palabra **"SOL"**. La primera letra del mensaje, la **P** se cifrará con la primera letra de la clave, **S**, lo que indica que tenemos que hacer la sustitución monoalfabeto **E("P") = E (16) = (16 + 19) mod 27 = 8 = "I"**, ya que, si **A** ocupa la posición 0, **S** ocupa la posición 19 en nuestro alfabeto. La letra **L** del mensaje se cifrará usando la letra **O** de la

clave, y la letra **A** del mensaje se cifrará usando la letra **L** de la clave. Para la última letra del mensaje (**N**) volveremos a usar a primera letra de la clave (**S**).

Por lo tanto, tenemos como resultado:

Mensaje P L A N

Clave S O L S

Cifrado I Z L F

Para facilitar las operaciones con este criptosistema, se dispone el llamado cuadro de Vigenère, que está formado por una matriz cuadrada de 27x27 en el caso de un alfabeto de 27 letras como el español. La primera fila de la matriz está formada por el alfabeto empezando por la letra A y acabando en la letra Z, la segunda por el alfabeto que empieza por la B y acaba en A, y así hasta la última fila, la 27ª, que empieza por las letras ZAB... y acaba con la letra Y.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig- Tablero Vigènere para el alfabeto inglés

Criptoanálisis del cifrado Vigenère

Para criptoanalizar este tipo de cifrado es necesario con efectuar d análisis estadísticos independientes agrupando los símbolos del criptograma en grupos distintos según la

k_i empleada para codificarlos; cada grupo estará codificado con el mismo alfabeto de cifrado. Para estimar la longitud d de la clave, (o sea, el número de alfabetos distintos empleados en el cifrado) buscaremos la periodicidad de los patrones comunes que puedan aparecer en el texto cifrado. Obviamente, para el criptoanálisis, necesitaremos al menos d veces más cantidad de texto cifrado que con los métodos monoalfabéticos.

1.3.- CIFRADOS POR TRANSPOSICIÓN.

Cambiamos las letras de lugar

La manera más sencilla de cifrar un texto y crear por tanto un criptograma, consiste en cambiar de lugar las letras de ese texto en claro, es decir desordenándolas de forma que su lectura nos lleve a algo sin sentido. Como ya sabemos por lecciones anteriores, con ello se logra el efecto de difusión, pero, en cambio, su talón de Aquiles es que el criptograma contiene exactamente las mismas letras que el texto en claro y, por lo tanto, se manifiesta de forma patente en dicho criptograma la redundancia característica del lenguaje que aparece en el texto en claro.

En la criptografía clásica este método de cifra por permutación muestra un corto recorrido y, por lo tanto, conocemos muy pocos algoritmos. Los sistemas clásicos se centrarán preferentemente en el método de cifrado por sustitución, donde sí encontramos una gran variedad de algoritmos de cifra que veremos en los tres siguientes capítulos de este MOOC.

No sucederá lo mismo en la cifra moderna, orientada como es obvio a bits y a bytes, en donde la técnica de permutación se usa frecuentemente. Como ejemplo, baste nombrar al algoritmo DES, que presenta varias operaciones de permutación de bits, tanto en el texto en claro como en las claves de cada vuelta, e incluso en el algoritmo AES -actual estándar

mundial de cifra simétrica-, dentro de la operación denominada ShiftRows, aunque en este caso la permutación se haga sobre bytes.

Recordando a la escítala

Figura 2.1. Cifrador de permutación escítala.

Como se supone ya has repasado la escítala en esa lección 2, a continuación, sólo vamos a realizar un par de prácticas en el laboratorio para fortalecer conceptos usando el software Criptoclásicos.

El método de cifrado por transposición consiste en reordenar datos para cifrarlos a fin de hacerlos ininteligibles. Esto puede significar, por ejemplo, reordenar los datos geométricamente para hacerlos visualmente inutilizables.

La técnica consiste en:

1. Enrollar una tira de papiro alrededor de un cilindro llamado scytale,
2. Escribir el texto a lo largo en la tira enrollada (el mensaje del ejemplo mostrado arriba es “comment ça marche”).

3. Cuando se desenrolla el mensaje, ya no tiene significado (“cecaeonar mt c m mh”). Para descifrar el mensaje, el destinatario simplemente necesita tener un cilindro del mismo diámetro. En realidad, un descifrador de códigos (¡había descifradores de códigos en esa época!) puede descifrar el mensaje probando cilindros con una serie de diámetros diferentes; esto significa que el método puede romperse estadísticamente (los caracteres sólo tienen que tomarse uno a uno, separados por una determinada distancia).

Los cifrados por transposición reordenan el texto de acuerdo con algún esquema. Este reordenamiento se hacía clásicamente con la ayuda de algún tipo de figura geométrica. Primero el texto a cifrar se escribía en la figura de una forma determinada y después se extraía de la figura de una forma diferente, quedando cifrado. La llave (clave) consiste pues en la forma de introducir y sacar el texto de la figura.

La figura escogida la mayoría de las veces era una matriz bidimensional. Como ejemplos podemos distinguir:

1. Cifrado por transposición columnar
2. Cifrado por transposición

El objetivo de las sustituciones es crear confusión. Una transposición es un cifrado en el que las letras del mensaje son cambiadas de posición. Su objetivo es el de la difuminar el mensaje. También se conoce como una permutación. En este caso al reordenar el criptograma aparecerán exactamente los mismos caracteres que en el texto en claro. Es fácil detectar que nos enfrentamos ante un cifrado por transposición si comprobamos que la frecuencia de aparición de caracteres cumple la estadística para algún idioma. Estas técnicas de cifrado son atacadas mediante técnicas de “ANAGRAMACIÓN”.

Cifrado por transposición columnar

Dado un texto a cifrar, se escribe por filas en una matriz de una anchura predeterminada y se obtiene el texto cifrado leyendo las columnas en algún orden. Por ejemplo, para cifrar el texto "El cristal roto empezaba a crecer de nuevo", con una anchura de bloque de 6 caracteres hacemos:

E L C R I S
T A L R O T
O E M P E Z
A B A A C R
E C E R D E
N U E V O

Ahora lo que se hace es leer el texto por columnas en cualquier orden. Por ejemplo, en el orden 2-4-6-1-3-5 tenemos:

"LAEBCU RRPARV STZRE ETOAEN CLMAEE IOECDO"

La llave (clave) de este cifrado es la permutación que se ha usado y las dimensiones de la tabla.
Como se ha explicado anteriormente lo único que hace este método es considerar el texto escrito por filas en una matriz y volver a escribir este texto cogiendo las columnas de dicha matriz. Debemos tener en cuenta que la llave (clave) en este cifrado es la permutación que se ha utilizado, es decir, la forma de elegir las columnas, y las dimensiones de la matriz donde se escribe el texto.

Teniendo en cuenta lo expuesto en el párrafo anterior, y que al escribir el texto cifrado hay espacios en blanco entre los "trozos" de texto que se corresponden con las columnas, entonces, una forma de realizar el criptoanálisis al texto anteriormente cifrado puede ser la siguiente:

Coger el texto cifrado y volverlo a poner en una matriz como la explicada anteriormente.

Intercambiar las columnas de dicha matriz hasta obtener un texto con sentido. Para facilitar la labor debemos tener en cuenta que la columna de menor longitud será la última columna de la matriz original. Entonces, partiendo de la tabla:

L R E C I S A
R T L O T E
P O M E Z B
A A A C R C
R E E D E
U V N E O

Realizando distintas permutaciones entre las columnas de la misma podemos volver a obtener la matriz original:

E L C R I S T
A L R O T O
E M P E Z A
B A A C R E
C E R D E
N U E V O

Y, por tanto, podemos descifrar el mensaje: "El cristal roto empezaba a crecer de nuevo"

Cifrado por transposición

Dado un texto a cifrar, se escribe por filas en una matriz de una anchura predeterminada y luego se cambian las columnas de sitio. Por ejemplo, para cifrar el texto "A quien madruga Dios le ayuda", con una anchura de bloque de 5 caracteres hacemos:

A Q U I E
N M A D R

U G A D I
O S L E A
Y U D A H

Metemos cualquier carácter de relleno al final, para que la matriz quede completamente rellena y cambiamos las columnas de sitio, por ejemplo, las ponemos en el orden 3-5-2-1-4 y obtenemos:

U E Q A I
A R M N D
A I G U D
L A S O E
D H U Y A

Con lo que el texto cifrado queda:

EQAI ARMND AIGUD LASOE DHUYA

Más formalmente, lo que se hace es dividir el texto en bloques de una longitud fija y aplicar a cada bloque una permutación p. En el ejemplo anterior, se dividiría el texto en bloques de 5 caracteres:

AQUIE NMADR UGADI OSLEA YUDAH

Se le añaden letras al final para terminar de llenar un bloque y se aplica la permutación (1, 3, 2, 5,4), con lo que se llega el resultado UEQAI ARMND AIGUD LASOE DHUYA, que coincide con el obtenido anteriormente.

Para realizar el criptoanálisis de un texto que ha sido cifrado con este método debemos seguir los pasos considéranos en el apartado de cifrado por transposición columnar. Además de lo expuesto anteriormente hay que tener en cuenta que ahora los huecos son rellenados con caracteres. Entonces, dado un texto cifrado con este método y sin

espacios en blanco para delimitar las columnas, podríamos obtener el número de columnas de la matriz sin más que conocer la longitud del texto cifrado, pues el número de columnas debe ser un divisor de este.

Como ejemplo de lo expuesto en el párrafo anterior consideremos el texto cifrado del anterior ejemplo:

```
UEQAIARMNDAIGUDLASOEDHUYA
```

Como la longitud de este texto es de 25 caracteres, y como: $25 = 5 \cdot 5$

Entonces, sabemos que la matriz debe contener 5 columnas. Con esta información podemos dividir el texto anterior y obtener la matriz del texto cifrado, con la que realizando distintas permutaciones entre sus columnas podríamos obtener la matriz del texto llano correspondiente a dicho texto cifrado, y con lo cual descifraríamos el mensaje oculto en el texto cifrado.

1.4.- RELLENOS DE UNA SOLA VEZ.

La construcción de un cifrado inviolable es bastante sencilla. La técnica se conoce desde hace décadas y consiste en escoger una cadena de bits al azar como clave. Luego se convierte el texto normal en una cadena de bits, por ejemplo, usando su representación ASCII. Por último, se calcula el or exclusivo (XOR) y cuya tabla de valores lógicos puede verse en la siguiente figura, de estas dos cadenas, bit por bit.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Figura 1.4: Tabla lógica de la función o-exclusivo (XOR).

El texto cifrado resultante no puede descifrarse porque cada texto normal posible es un candidato igualmente probable. El texto cifrado no proporciona al criptoanalista ninguna información en absoluto. En una muestra suficientemente grande de texto cifrado, cada letra ocurrirá con la misma frecuencia, al igual que cada diagrama (combinación de dos letras) y cada trigrama (combinación de tres letras). Como ejemplo, cifremos el mensaje "texto cifrado" con la cadena "En un lugar de la Mancha de cuyo nombre…"

Texto original	t	e	x	t	o		c	i	f	r	a	d	o
Codificación ASCII (hex)	74	65	78	74	6F	20	63	69	66	72	61	64	6F
Texto de cifrado	E	n		u	n		l	u	g	a	r		d
Codificación ASCII (hex)	45	6E	20	75	6E	20	6C	75	67	61	72	20	64
Codificación cifrada (hex)	*31*	*0B*	*58*	*01*	*01*	*00*	*0F*	*1C*	*01*	*13*	*13*	*44*	*08*

Figura 1.4.1: Cifrado de un texto mediante relleno de una sola vez.

Si procedemos ahora a descifrarlo con la clave de codificación, obtenemos el mensaje original:

Codificación cifrada (hex)	*31*	*0B*	*58*	*01*	*01*	*00*	*0F*	*1C*	*01*	*13*	*13*	*44*	*08*
Texto de cifrado	E	n		u	n		l	u	g	a	r		d
Codificación ASCII (hex)	45	6E	20	75	6E	20	6C	75	67	61	72	20	64
Codificación ASCII (hex)	74	65	78	74	6F	20	63	69	66	72	61	64	6F
Texto original	t	e	x	t	o		c	i	f	r	a	d	o

Figura 1.4.2: Descifrado de un texto cifrado mediante relleno de una sola vez.

Sin embargo, este método tiene varias desventajas prácticas. En primer lugar, la clave no puede memorizarse, por lo que tanto el transmisor como el receptor deben llevar una copia por escrito consigo. Además, la cantidad total de datos que pueden transmitirse está limitada a la cantidad de clave disponible. Otro problema es la sensibilidad del

método a la pérdida o inserción de caracteres. Si el transmisor y el receptor pierden la sincronía, todos los datos a partir de ahí aparecerán alterados.

Criptografía clásica

La criptografía clásica se basa en algoritmos sencillos y claves muy largas para la seguridad. Las técnicas criptográficas clásicas son básicamente dos, el cifrado por sustitución y el cifrado por trasposición.

1.5.- DOS PRINCIPIOS CRIPTOGRÁFICOS FUNDAMENTALES.

Aunque la criptografía es muy variada existen dos principios fundamentales que sostienen la criptografía y que es importante entender. El primer principio es que todos los mensajes cifrados deben contener redundancia, es decir, información no necesaria para entender el mensaje.

Redundancia

Si una persona lee un mensaje en el que faltan algunas letras, normalmente puede reconstruirlo.

Esto ocurre porque casi todos los símbolos de un mensaje en lenguaje natural contienen información que se puede extraer de los símbolos de alrededor —información que, en la práctica, se está enviando dos o más veces, o, en otras palabras, porque el lenguaje natural es redundante.

Definiremos redundancia como cierta repetición de la información contenida en un mensaje, que permite, a pesar de la pérdida de una parte de este, reconstruir su contenido.

Puesto que tenemos mecanismos para definir la cantidad de información que presenta un suceso, podemos intentar medir el exceso de información (redundancia) de un lenguaje.

Todos los mensajes deben contener redundancias (es decir información innecesaria para la comprensión del mensaje) para evitar que un intruso pueda modificar aleatoriamente un mensaje con cierta probabilidad de obtener un mensaje válido desde el punto de vista del descifrado, pero con datos alterados. Al mismo tiempo esta redundancia hace que sea más fácil para el criptoanalista romper el código. La redundancia no puede ser simplemente agregar n ceros al comienzo o el fin del mensaje, puesto que ello permitiría que ciertos algoritmos puedan obtener ciertos resultados predecibles. Por ello es preferible agregar cierta secuencia randómica de caracteres.

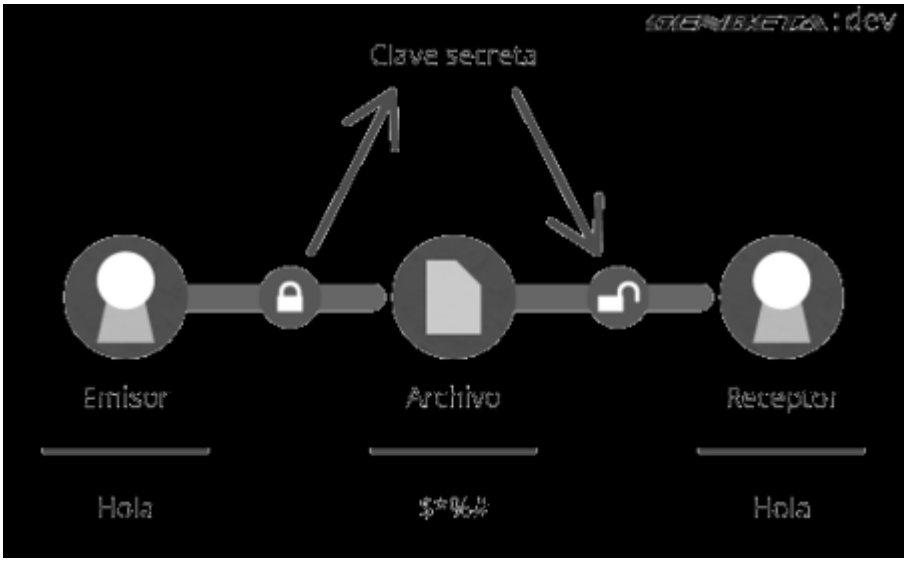
Sin embargo, la adición de redundancia simplifica a los criptoanalistas el descifrado delos mensajes, pues ahora un criptoanalista puede saber que ha descifrado correctamente un mensaje al comprobar que los 8 primeros bytes son ceros. Por ello, una cadena aleatoria de palabras sería mejor para incluir en la redundancia. Otro ejemplo de este primer principio, puede ser el concepto de un CRC (CyclicRedundant Check), es decir una información redundante puesta al final de un mensaje, evitando al máximo que otros puedan generar información que pueda ser interpretada e introduzca vulnerabilidad al proceso de comunicaciones.

Actualización.

El segundo principio criptográfico es el de actualización el cual implica que se deben tomar medidas para asegurar que cada mensaje recibido se verifique a fin de saber siesta actualizado. Esto permite evitar que posibles intrusos activos reproduzcan mensajes antiguos. Una de las medidas es incluir en cada mensaje una marca de tiempo válida por ejemplo durante 10 segundos, para compararlo con los mensajes nuevos que lleguen y filtrar los duplicados. Los mensajes con una antigüedad mayor a10 segundos pueden descartarse.

Criptografía simétrica

La criptografía simétrica solo utiliza una clave para cifrar y descifrar el mensaje, que tiene que conocer el emisor y el receptor previamente y este es el punto débil del sistema, la comunicación de las claves entre ambos sujetos, ya que resulta más fácil interceptar una clave que se ha transmitido sin seguridad (diciéndola en alto, mandándola por correo electrónico u ordinario o haciendo una llamada telefónica).



Teóricamente debería de ser más fácil conocer la clave interceptándola que probándola una por una por fuerza bruta, teniendo en cuenta que la seguridad de un mensaje cifrado debe recaer sobre la clave y nunca sobre el algoritmo (por lo que sería una tarea eterna reventar la clave, como comenté en un ejemplo de ataque por fuerza bruta).

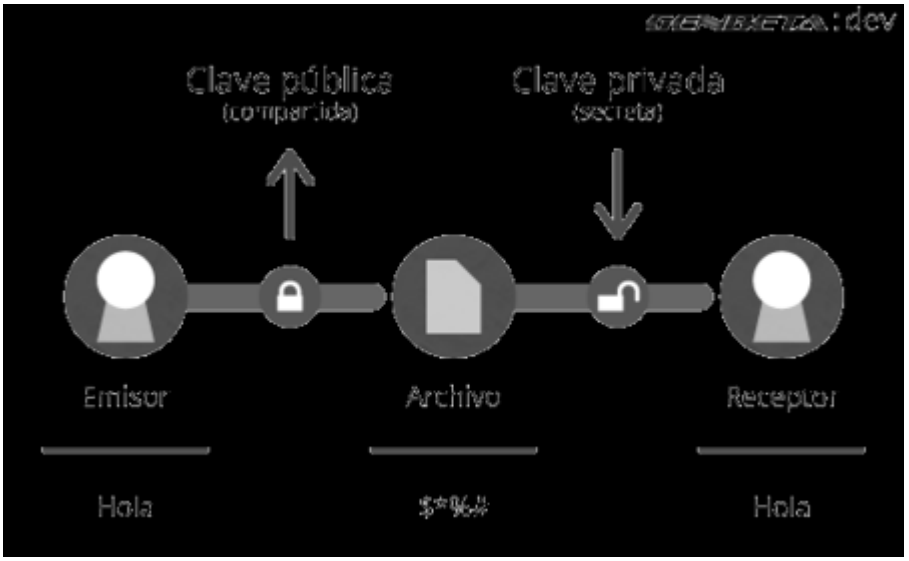
Para poner un ejemplo la máquina **Enigma** (que era una máquina de cifrada electromecánica que generaba abecedarios según la posición de unos rodillos que podrían tener distintas órdenes y posiciones) usaba un método simétrico con un algoritmo que dependía de una clave (que más que clave parece un ritual) que está formada por: los rotores o rodillos que usaba, su orden y la posición de cada anillo, siendo esto lo más básico.

La máquina **Enigma** contaba también con un libro de claves que contenía la *clave del día* y hacia un poco más difícil encontrar la clave, pero no es una clave lo suficientemente segura como para que no se pudiese reventar, sobre todo cuando los ingleses gracias a los polacos consiguieron el algoritmo, por este motivo la mayoría de los días conseguían la clave.

Y otro inconveniente que tiene este sistema es que si quieres tener un contenido totalmente confidencial con 10 personas tienes que aprenderte o apuntarte (siendo esta forma menos segura) las 10 claves para cada persona.

Criptografía asimétrica

La criptografía asimétrica se basa en el uso de **dos claves**: **la pública** (que se podrá difundir sin ningún problema a todas las personas que necesiten mandarte algo cifrado) y **la privada** (que no debe de ser revelada nunca).



Sabiendo lo anterior, si queremos que tres compañeros de trabajo nos manden un archivo cifrado debemos de mandarle nuestra clave pública (que está vinculada a la privada) y nos podrán mandar de forma confidencial ese archivo que solo nosotros podremos descifrar con la clave privada.

Puede parecer a simple vista un sistema un poco *cojo* ya que podríamos pensar que sabiendo la clave pública podríamos deducir la privada, pero este tipo de sistemas criptográficos usa algoritmos bastante complejos que generan a partir de la frase de paso (la contraseña) la clave privada y pública que pueden tener perfectamente un tamaño de 2048bits (probablemente imposible de reventar).

Como os habréis dado cuenta solo cifra una persona (con la clave pública) y la otra se limita a mirar el contenido, por lo que la forma correcta de tener una comunicación

bidireccional sería realizando este mismo proceso con dos pares de claves, o una por cada comunicador.

Otro propósito de este sistema es también el de poder firmar documentos, certificando que el emisor es quien dice ser, firmando con la clave privada y verificando la identidad con la pública.

Nota: todo esto puede parecer lioso (y lo es) pero hablaré de cómo poner en práctica esto con **GnuPG** (una herramienta de cifrado libre muy usada para este propósito) y será más fácil de comprender.

Diferencias entre criptografía simétrica y asimétrica

Para empezar, la criptografía simétrica es más insegura ya que el hecho de pasar la clave es una gran vulnerabilidad, pero se puede cifrar y descifrar en menor tiempo del que tarda la criptografía asimétrica, que es el principal inconveniente y es la razón por la que existe la criptografía híbrida.

Criptografía híbrida

Este sistema es la unión de las ventajas de los dos anteriores, debemos de partir que el problema de ambos sistemas criptográficos es que el simétrico es inseguro y el asimétrico es lento.

El proceso para usar un sistema criptográfico híbrido es el siguiente (para enviar un archivo):

- Generar una clave pública y otra privada (en el receptor).
- Cifrar un archivo de forma síncrona.
- El receptor nos envía su clave pública.

- Ciframos la clave que hemos usado para encriptar el archivo con la clave pública del receptor.
- Enviamos el archivo cifrado (síncronamente) y la clave del archivo cifrada (asíncronamente y solo puede ver el receptor).

1.6 SEGURIDAD

Criptografía y seguridad informática son dos elementos que crean una llave perfecta que abre tus entornos digitales. Si bien cada elemento surgió y evolucionó de manera autónoma para ganar por mérito propio su correspondiente sitial de honor; criptografía y seguridad informática se combinan para garantizar el acceso exclusivo únicamente a quienes autorices.

1.7 CIFRADO DE CLAVE SIMETRICA

La criptografía de clave simétrica es un término utilizado para los algoritmos criptográficos que utilizan la misma clave para el cifrado y el descifrado. La clave se suele llamar "clave simétrica" o "clave secreta".

Esto generalmente se contrasta con criptografía de clave pública en el que las claves se generan en pares, y la transformación realizada por una clave solo se puede revertir utilizando la otra clave.

Los algoritmos de clave simétrica son seguros y altamente eficientes cuando se usan de manera adecuada, de modo que pueden usarse para cifrar grandes cantidades de datos sin tener un efecto negativo en el rendimiento.

1.8 CIFRADO DE CLAVE ASMETRICA

La criptografía de clave asimétrica también es conocida como clave pública, emplea dos llaves diferentes en cada uno de los extremos de la comunicación para cifrarla y descifrarla. Cada usuario de la comunicación tendrá una clave pública y otra privada. La clave privada tendrá que ser protegida y guardada por el propio usuario, será secreta y no la deberá conocer absolutamente nadie ni tampoco debe ser enviada a nadie. La clave pública será accesible por todos los usuarios del sistema que quieran comunicarse.

1.9 CIFRADO HASH

Una funcion hash es un algoritmo que mapea un conjunto grande de datos de tamaño variable, llamados claves, en pequeños conjuntos de datos de longitud fija.

Los algoritmos hash son de vtal importancia en la firma digital ya que garantizan la integridad)

Las propiedades mas improtantes de las funciones hash son:

- Independientemente del tamaño del mensaje original, al aplicare la funcion hash, la huella resultante siempre tendra el mismo tamaño.
- Si el hash es cambiado quiere decir que con tan solo cambiar un bit del mensaje original, salta la alarma.
- Resistencia a la segunda preimagen: dado un mensaje x, no es posible encontar otro mensaje x que produzca el mismo valor hash.
- Resistencia a colisiones: no es posible encontar dos entradas que den lugar al mismo valor hash.

1.10 SISTEMA HIBRIDO

Los criptosistemas híbridos tratan de aprovechar lo mejor de cada uno de los sistemas de cifrado de clave simétrica y asimétrica. En resumen, se trata de obtener un criptosistema rápido y eficiente que permite el intercambio de contraseñas en canales de comunicación inseguros.

1.11 EJERCICIOS

PASIVO	ACTIVO
Escaneo completo de la maquina	Tener siempre un usuario auxiliar
Firewall	No abrir link desconocidos
Airbacg	Backups
SAI	Antivirus

Seguridad activa y pasiva – PC

Seguridad activa y pasiva - coche

PASIVO	ACTIVO
Cinturón de seguridad	Sistema de direccion
Airbac	Frenos
Cristales	Luces
Chasis y carrocería	los neumáticos

3. ¿Cual es el activo mas valioso para una empresa?

Los datos y obviamente los servidores.

a) ¿Que vulnerabilidad podrían afectarle?

Un ataque por parte de loso trabajadores.

b) ¿Que amenazas son las que podrían afectarle? Clasificalas

4.¿Crees que la evaluación de riesgos sera igual para todas las empresas? ¿Por que?

Todas las empresas no necesitan la misma evaluación de riesgo, porque dependiendo de la empresa de la que se quiera evaluar tendrá puntos fuerte y puntos débiles a evaluar a diferencia de otra empresa.

5. Enumera posibles preguntas que podrían hacerse en la realización de una evaluación de riesgos.

6. Busca en Internet aplicaciones comerciales que permitan realizar una evaluación de riesgos.

EVALCARGAS

F-PSICO

RISKOFDERM

PVCHECH

7. A partir de los principios expresados en este epigrafe:

a) Plantea un posible ataque contra uno de estos principios.

b) Indica una posible solución para cada uno de los ataques planeados.

1.12 ESQUEMAS

Los esquemas criptográficos visuales son protocolos criptográficos que codifican un mensaje formado por una imagen definida por píxeles envez de codificar un mensaje de texto definido por letras y números.

UNIDAD II

ALGORITMOS DE CLAVES SIMETRICAS

¿Qué significa cifrar?

Aplicar un algoritmo de cifrado determinado junto con una clave, a una determinada información que se quiere transmitir confidencialmente. Dentro del cifrado digital encontramos dos tipos de criptografía: **simétrica y asimétrica.** En este artículo hablaremos sobre la **Criptografía de clave simétrica.**

El cifrado mediante clave simétrica significa que dos o más usuarios, tienen una única clave secreta, esta clave será la que cifrará y descifrará la información transmitida a través del canal inseguro. Es decir, la clave secreta la deben tener los dos usuarios, y con dicha clave, el usuario A cifrará la información, la mandará a través del canal inseguro, y a continuación el usuario B descifrará esa información con la MISMA clave que ha usado el usuario A.

Para que un algoritmo de clave simétrica sea fiable debe cumplir:

- Una vez que el mensaje es cifrado, no se puede obtener la clave de cifrado/descifrado ni tampoco el texto en claro.
- Si conocemos el texto en claro y el cifrado, se debe tardar más y gastar más dinero en obtener la clave, que el posible valor derivado de la información sustraída (texto en claro).

Debemos tener en cuenta que los algoritmos criptográficos son públicos, por lo que su fortaleza debe depender de su complejidad interna, y de la longitud de la clave empleada para evitar los ataques de fuerza bruta.

La seguridad en clave simétrica reside en la propia clave secreta, y por tanto el principal problema es la distribución de esta clave a los distintos usuarios para cifrar y descifrar la información. La misión del emisor y receptor es mantener la clave en secreto. Si cae en

manos equivocadas ya no podríamos considerar que la comunicación es segura y deberíamos generar una nueva clave.

Otro problema reside en que las claves secretas a guardar son proporcionales al número de canales seguros que deseamos mantener. Esto no es un problema en sí, pero debemos administrar bien las llaves para no equivocarnos. Este problema no se va a presentar en los algoritmos asimétricos porque cada usuario tiene una pareja de claves, una pública y la otra privada, independientemente del número de canales seguros que queramos establecer. Únicamente debe proteger la clave privada.

La principal ventaja de los algoritmos simétricos es la velocidad de los algoritmos, y son muy usados para el cifrado de grandes cantidades de datos. VeraCrypt, por ejemplo, usa algoritmos simétricos para cifrar toda la información.

Si alguien se pregunta que cómo podemos transmitir por un medio inseguro la clave simétrica, la respuesta es que podemos crear unas claves asimétricas y así transmitir la información. Es lo que se hace en el **cliente-servidor OpenVPN**.

Os voy a presentar algunos algoritmos de clave simétrica, algunos ya no son seguros, pero vamos a ver por qué.

2.1. DES – EL ESTÁNDAR DE ENCRIPTACIÓN DE DATOS

El 15 de mayo de 1973, el **NBS** (National Bureau of Standars, en castellano: Agencia Nacional de Normalización) hoy en día denominada NIST (National Institute of Standars and Technology, en castellano: Instituto Nacional de Normalización y Tecnología), hizo un llamamiento en el Federal Register (el equivalente en España del Boletín Oficial del Estado) para la creación de un algoritmo de cifrado que cumpliera con los siguientes requisitos:

- ofrecer un alto nivel de seguridad relacionado con una pequeña clave utilizada para cifrado y descifrado
- ser comprensible
- no depender de la confidencialidad del algoritmo
- ser adaptable y económico
- ser eficaz y exportable

A finales de 1974, IBM propuso "Lucifer", que gracias a la NSA (National Standard Agency, en castellano: Agencia Nacional de Seguridad) fue modificado el 23 de noviembre de 1976, convirtiéndose en **DES** (Data Encryption Standard, en castellano: Estándar de Cifrado de Datos). El DES fue aprobado por el NBS en 1978. El DES fue estandarizado por el ANSI (American National Standard Institute, en castellano: Instituto Nacional Americano de Normalización) bajo el nombre de ANSI X3.92, más conocido como DEA (Data Encrytion Algorithm, en castellano: Algoritmo de Cifrado de Datos).

Principio de funcionamiento del DES

Se trata de un sistema de cifrado simétrico por bloques de 64 bits, de los que 8 bits (un byte) se utilizan como control de paridad (para la verificación de la integridad de la clave). Cada uno de los bits de la clave de paridad (1 cada 8 bits) se utiliza para controlar uno de los bytes de la clave por paridad impar, es decir, que cada uno de los bits de paridad se ajusta para que tenga un número impar de "1" dentro del byte al que pertenece. Por lo tanto, la clave tiene una longitud "útil" de 56 bits, es decir, realmente sólo se utilizan 56 bits en el algoritmo.

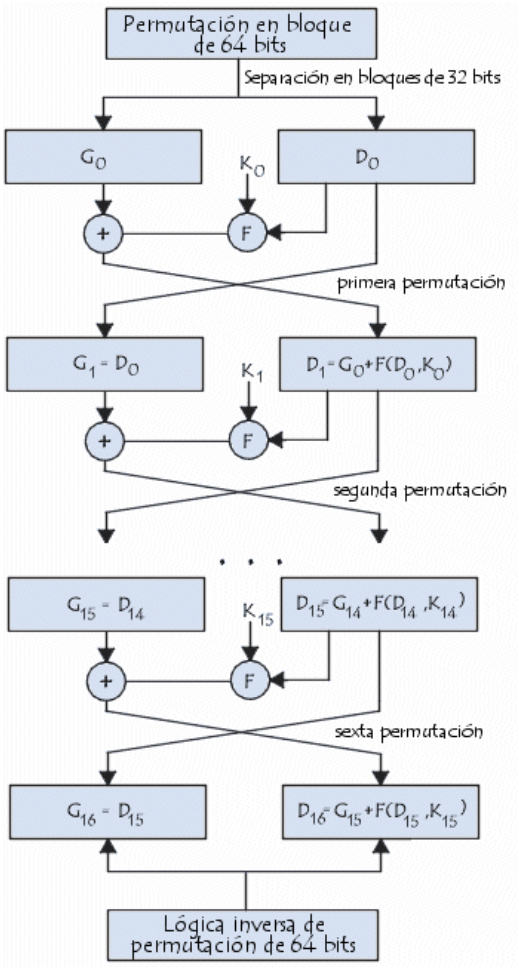
El algoritmo se encarga de realizar combinaciones, sustituciones y permutaciones entre el texto a cifrar y la clave, asegurándose al mismo tiempo de que las operaciones puedan realizarse en ambas direcciones (para el descifrado). La combinación entre sustituciones y permutaciones se llama **cifrado del producto**.

La clave es codificada en 64 bits y se compone de 16 bloques de 4 bits, generalmente anotadas de k_1 a k_{16}. Dado que "solamente" 56 bits sirven para el cifrado, ¡puede haber hasta 2^{56} (o $7.2*10^{16}$) claves diferentes!

El algoritmo DES

Las partes principales del algoritmo son las siguientes:

- fraccionamiento del texto en bloques de 64 bits (8 bytes),
- permutación inicial de los bloques,
- partición de los bloques en dos partes: izquierda y derecha, denominadas I y Despectivamente,
- fases de permutación y de sustitución repetidas 16 veces (denominadas **rondas**),
- reconexión de las partes izquierda y derecha, seguida de la permutación inicial inversa.



Fraccionamiento del texto

Permutación inicial

En primer lugar, cada bit de un bloque está sujeto a una permutación inicial, que puede representarse mediante la siguiente matriz de permutación inicial (anotada como PI):

IP								
	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Esta tabla de permutación muestra, al leerla de izquierda a derecha y de arriba a abajo, que el 58° bit de un bloque de 64 bits está en la primera posición, el 50° está en la segunda posición y así sucesivamente.

División en bloques de 32 bits

Una vez que la permutación inicial se completó, el bloque de 64 bits se divide en dos bloques de 32 bits denominados **I** y **D** respectivamente (para izquierda y derecha, siendo la anotación en anglo-sajón L y R por Left y Right). El estado inicial de estos dos bloques se denomina $\mathbf{L}_0$ y $\mathbf{R}_0$:

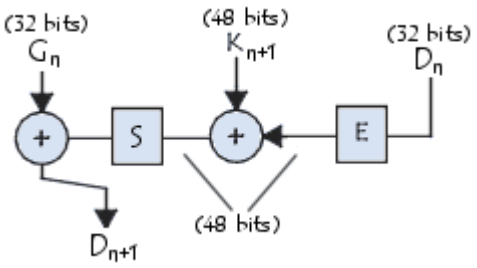
L_0								
	58	50	42	34	26	18	10	2

	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
R_0	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Es interesante observar que $\mathbf{L}_0$ contiene todos los bits que se encuentran en posición par en el mensaje inicial, mientas que $\mathbf{R}_0$ contiene los bits en posición impar.

Rondas

Los bloques $\mathbf{L}_n$ y $\mathbf{R}_n$ están sujetos a un conjunto de transformaciones iterativas denominadas rondas, que se muestran en este esquema y que detallamos a continuación:



Función de expansión

Los 32 bits del bloque $\mathbf{R}_0$ se expanden a 48 bits gracias a una tabla (matriz) llamada tabla de expansión (que se anota como **E**), en la que los 48 bits se mezclan y 16 de ellos se duplican:

E						
	32	1	2	3	4	5
	4	5	6	7	8	9
	8	9	10	11	12	13
	12	13	14	15	16	17
	16	17	18	19	20	21
	20	21	22	23	24	25
	24	25	26	27	28	29
	28	29	30	31	32	1

Así, el último bit de $\mathbf{R}_0$ (es decir, el 7° bit del bloque de origen) se convierte en el primero, el primero en el segundo, etc.

Además, los bits 1,4,5,8,9,12,13,16,17,20,21,24,25,28 y 29 de $\mathbf{R}_0$ (respectivamente los bits 57, 33, 25, l, 59, 35, 27, 3, 6l, 37, 29, 5, 63, 39, 31 y 7 del bloque de origen) son duplicados y diseminados en la matriz.

OR exclusiva con la clave

La tabla resultante de 48 bits se denomina $\mathbf{D'}_0$ o $\mathbf{E[D_0]}$. El algoritmo DES aplica después OR exclusivas entre la primera clave $\mathbf{K}_1$ y $\mathbf{E[D_0]}$. El resultado de este OR exclusivo es una tabla de 48 bits que, por comodidad, llamaremos $\mathbf{D}_0$ (¡no es la $\mathbf{D}_0$ inicial!).

Función de sustitución

Después, $\mathbf{D}_0$ se divide en 8 bloques de 6 bits, denominado $\mathbf{D}_{0i}$. Cada uno de estos bloques se procesa a través de **funciones de selección** (a veces llamadas cajas de sustitución o funciones de compresión), denominadas generalmente $\mathbf{S}_i$. Los primeros y últimos bits de cada $\mathbf{D}_{0i}$ determinan (en valor binario) la línea de la función de selección; los otros bits (2, 3, 4 y 5 respectivamente) determinan la columna. Como la selección de la línea se basa en dos bits, existen 4 posibilidades (0,1,2,3). Como la selección de la columna se basa en 4 bits, existen 16 posibilidades (0 a 15). Gracias a esta información, la función de selección "selecciona" un valor cifrado de 4 bits.

Esta es la primera función de sustitución, representada en una tabla de 4 por 16:

$\mathbf{S}_1$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Sea $\mathbf{R}_{01}$ igual a 101110. El primer y último bit dan 10, es decir, 2 en valor binario. Los bits 2,3,4 y 5 dan 0111, o 7 en valor binario. Por lo tanto, el resultado de la función de selección es el valor ubicado en la línea n° 2, de la columna n° 7. Es el valor 11 o 111en binario.

Cada uno de los 8 bloques de 6 bits pasa a través de la función de selección correspondiente, dando un resultado de 8 valores con 4 bits cada uno. A continuación, están las otras funciones de selección:

S_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4

3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Por lo tanto, cada bloque de 6 bits se sustituye por un bloque de 4 bits. Estos bits se combinan para formar un bloque de 32 bits.

Permutación

Finalmente, el bloque de 32 bits se somete a una permutación **P**. A continuación, mostramos la tabla:

P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

OR exclusivo

El conjunto de estos resultados salidos de **P** está sujeto a un OR exclusivo con $\mathbf{I}_0$inicial (como se muestra en el primer esquema) para devolver D1, en tanto que la $\mathbf{D}_0$inicial devuelve $\mathbf{I}_1$.

Iteración

El conjunto de los pasos anteriores (rondas) se reitera 16 veces.

Permutación inicial inversa

Al final de las iteraciones, los dos bloques $\mathbf{L}_{16}$ y $\mathbf{R}_{16}$ se vuelven a conectar y se someten a una permutación inicial inversa:

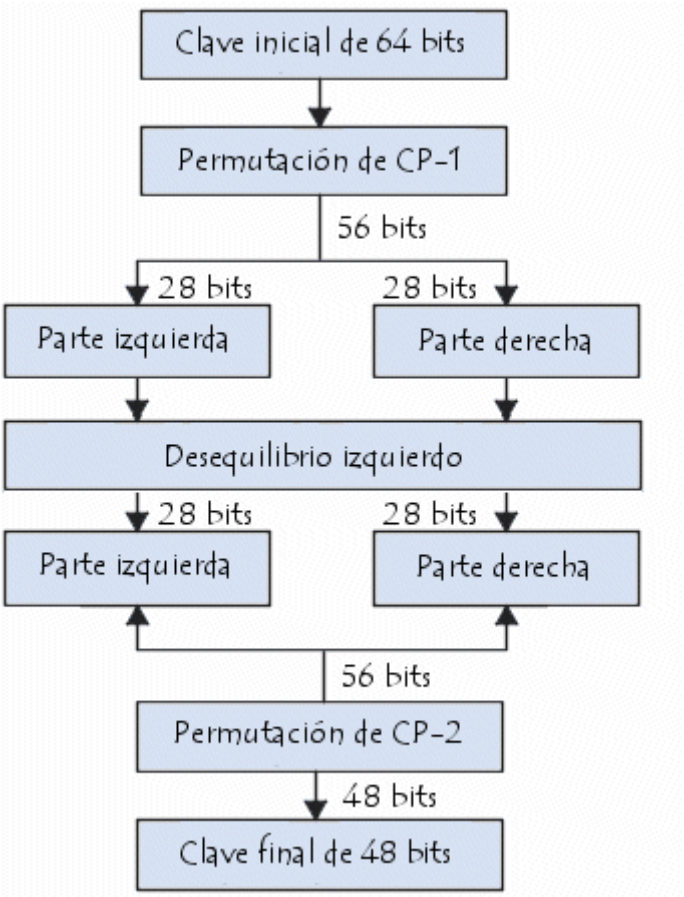
IP-1								
	40	8	48	16	56	24	64	32
	39	7	47	15	55	23	63	31
	38	6	46	14	54	22	62	30
	37	5	45	13	53	21	61	29
	36	4	44	12	52	20	60	28
	35	3	43	11	51	19	59	27
	34	2	42	10	50	18	58	26
	33	1	41	9	49	17	57	25

¡El resultado que surge es un texto cifrado de 64 bits!

Generación de claves

Dado que el algoritmo DES mencionado anteriormente es público, toda la seguridad se basa en la complejidad de las claves de cifrado.

El algoritmo que sigue a continuación muestra cómo obtener a partir una clave de 64 bits (compuesta por cualquier de los 64 caracteres alfanuméricos), 8 claves diferentes de 48 bits, cada una de ellas utilizadas en el algoritmo DES:



En primera instancia, se eliminan los bits de paridad de la clave para obtener una clave que posea una longitud de 56 bits.

El primer paso es una permutación denominada **PC-1**, cuya tabla se presentará a continuación:

PC-1														
	57	49	41	33	25	17	9	1	58	50	42	34	26	18
	10	2	59	51	43	35	27	19	11	3	60	52	44	36
	63	55	47	39	31	23	15	7	62	54	46	38	30	22
	14	6	61	53	45	37	29	21	13	5	28	20	12	4

Esta matriz puede escribirse en forma de dos matrices $\mathbf{L}_i$ y $\mathbf{R}_i$ (para la izquierda y la derecha respectivamente), cada una ellas de 28 bits:

$\mathbf{L}_i$	57	49	41	33	25	17	9
	1	58	50	42	34	26	18
	10	2	59	51	43	35	27
	19	11	3	60	52	44	36
$\mathbf{R}_i$	63	55	47	39	31	23	15
	7	62	54	46	38	30	22
	14	6	61	53	45	37	29
	21	13	5	28	20	12	4

El resultado de esta primera permutación se denomina $\mathbf{I}_0$ y $\mathbf{D}_0$.

Luego, estos dos bloques se rotan hacia la izquierda, de manera que los bits que estaban en la segunda posición pasan a la primera, aquellos que estaban en tercera posición pasan a la segunda, etc.

Los bits que estaban en la primera posición se mueven hacia la última posición.

Los dos bloques de 28 bits se agrupan en un bloque de 56 bits. Este pasa por una permutación, denominada **PC-2**, dando como resultado un bloque de 48 bits que representa la clave $\mathbf{K}_i$.

pc-2	14	17	11	24	1	5	3	28	15	6	21	10

	23	19	12	4	26	8	16	7	27	20	13	2
	41	52	31	37	47	55	30	40	51	45	33	48
	44	49	39	56	34	53	46	42	50	36	29	32

Realizando iteraciones del algoritmo es posible obtener las 16 claves $\mathbf{K}_1$ a $\mathbf{K}_{16}$utilizadas en un algoritmo DES.

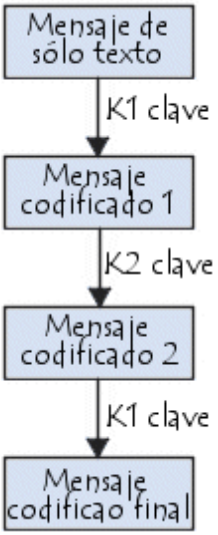
LS	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

TDES (en castellano: Triple Estándar de Cifrado de Datos), una alternativa para DES

En 1990, Eli Biham y Adi Shamir desarrollaron el criptoanálisis diferencial, que buscaba pares de textos planos y pares de textos cifrados. Este método funciona con un máximo de 15 rondas, mientras que en el algoritmo presentado anteriormente admite 16 rondas.

Por otro lado, aunque una clave de 56 bits ofrece una enorme cantidad de posibilidades, muchos procesadores pueden calcular más de 10^6 claves por segundo. Con lo que, cuando se utilizan al mismo tiempo una gran cantidad de máquinas, es posible que un gran organismo (un Estado, por ejemplo) encuentre la clave correcta.

Una solución a corto plazo requiere que se encadenen tres cifrados DES mediante dos claves de 56 bits (esto equivale a una clave de 112 bits). Este proceso se llama **Triple DES**, denominado TDES (algunas veces 3DES o 3-DES).



El **TDES** permite aumentar de manera significativa la seguridad del DES, pero posee la desventaja de requerir más recursos para el cifrado y descifrado.

Por lo general, se reconocen diversos tipos de cifrado triple DES:

- DES-EEE3: Cifrado triple DES con 3 claves diferentes,
- DES-EDE3: una clave diferente para cada una de las operaciones de triple DES (cifrado, descifrado, cifrado),
- DES-EEE2 y DES-EDE2: una clave diferente para la segunda operación (descifrado).
- En 1997, elNIST lanzó una nueva convocatoria para que desarrollar el **AES** (Advanced Encryption Standard, en castellano: Estándar de Cifrado Avanzado), un algoritmo de cifrado cuyo objetivo era reemplazar al DES.

El sistema de cifrado DES se actualizaba cada 5 años. En el año 2000, durante su última revisión y después de un proceso de evaluación que duró 3 años, el NISTseleccionó como nuevo estándar un algoritmo diseñado conjuntamente por dos candidatos belgas, el Sr. Vincent Rijmen y el Sr. Joan Daemen. El nuevo algoritmo, llamado por sus inventores **RIJNDAEL** reemplazará, de ahora en adelante, al DES.

2.2. AES – EL ESTÁNDAR DE ENCRIPTACIÓN AVANZADA

AES significa Advanced Encryption Standard. Aunque sus raíces se remontan a 1997, actualmente sigue siendo el único algoritmo en la lista del National Institute of Standards and Technology (NIST) para proteger datos clasificados.

AES se adoptó a partir de una colección más grande publicada originalmente como Rijndael – una palabra compuesta de los dos nombres de sus creadores belgas. Se ha analizado ampliamente y ahora se utiliza en todo el mundo en algunas de las aplicaciones de seguridad más exigentes.

AES es lo que se conoce como un cifrado simétrico por bloques, lo que significa que cifra y descifra los datos en bloques de 128 bits cada uno. Para ello, utiliza una clave criptográfica específica, que es efectivamente un conjunto de protocolos para manipular información. Esta clave puede ser de 128, 192 o 256 bits de tamaño.

AES-256 – la versión clave de 256 bits de AES – es el estándar de cifrado utilizado por Le VPN. Es la forma más avanzada del cifrado y consiste en 14 rondas de sustitución, transposición y mezcla para un nivel de seguridad excepcionalmente alto.

Los Beneficios del Cifrado AES-256

AES-256 es el primer cifrado públicamente accesible y abierto aprobado por la Agencia Nacional de Seguridad de Estados Unidos (NSA) para la información ultra-secreta. Su tamaño de clave mayor hace que sea esencialmente irrompible, lo que significa que, incluso si nuestros servidores fueran hackeados, tus datos serían imposibles de descifrar.

AES-256 también tiene la ventaja de ser extremadamente rápido. Cuando navegas por la web con una VPN que utiliza el cifrado AES-256 en sus servidores, no experimentarás ninguna disminución en el rendimiento en comparación con otro protocolo de seguridad.

Por razones de seguridad y conveniencia, exige una VPN con cifrado AES-256.
Este algoritmo es el más conocido entre los usuarios de routers, ya que WPA opera con AES como método de cifrado. Este cifrado puede implementar tanto en sistemas hardware como en software. El sistema criptográfico AES opera con bloques de tamaño

fijo de 128 bits y claves de longitudes variable, hay AES de 128bits, de 192 bits 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).

Seguridad de AES:

AES tiene 10 rondas para llaves de 128 bits, 12 rondas para llaves de 192 bits y 14 rondas para llaves de 256 bits. En el año 2006, los mejores ataques conocidos fueron las 7 rondas para claves de 128 bits, 8 rondas para llaves de 192 bits, y 9 rondas para claves de 256 bits. Algunos criptógrafos muestran preocupación sobre la seguridad del AES. Ellos creen que el margen entre el número de rondas especificado en el cifrador y los mejores ataques conocidos es muy pequeño.

Otra preocupación es la estructura de AES. A diferencia de la mayoría de cifradores de bloques, AES tiene una descripción matemática muy ordenada. Recordemos que AES es usado en los cifrados wireless de los routers de los hogares como método de cifrado (no clave) ya que en los routers podemos usar una clave estática o una dinámica mediante un servidor Radius. AES también es usado por OpenSSL y por supuesto en OpenVPN (ya que usa las librerías OpenSSL) e IPSec.

La forma en que se gestionan estos bloques de mensaje, se denomina «modo de cifrado».

Por ejemplo, existe el AES-CBC, AES-CFB y AES-OFB, os voy a explicar qué es exactamente esto que aparece en las librerías criptográficas como OpenSSL y LibreSSL.

- **CBC (Cipher-block chaining):** a cada bloque de texto plano se le aplica la operación XOR con el bloque cifrado anterior antes de ser cifrado. De esta forma, cada bloque de texto cifrado epende de todo el texto en claro procesado hasta este punto. Como no se dispone de un texto cifrado con el que combinar el primer bloque, se usa un vector de inicialización IV (número aleatorio que puede ser públicamente conocido). La desventaja es que el cifrado es de forma secuencial y por tanto no puede ser paralelizado.
- **OFB (Output feedback):** se generan bloques de flujo de claves, que son operados con XOR y el texto en claro para obtener el texto cifrado. Al igual que con otras unidades de flujo de cifrado, al intercambiar un bit en el texto cifrado produce texto cifrado con un bit intercambiado en el texto plano en la misma ubicación. También se usa un vector de inicialización que debe ser único.
- **CFB (Cipher feedback):** se hace igual que en OFB, pero para producir el keystream cifra el último bloque de cifrado, en lugar del último bloque del keystream como hace OFB. Un bit erróneo en el texto cifrado genera 1+64/m bloques de texto claro incorrectos (siendo m la longitud del flujo en el que se divide el bloque). El cifrado no puede ser paralelizado, sin embargo, el descifrado sí.

Operación básica

El AES puede ser descrito como un bloque cifrado iterativo y simétrico. El AES utiliza una estructura de bucle para realizar repetidamente reordenamientos de datos, o permutaciones. El bucle reemplaza una unidad de datos con otra para datos de entrada. La rutina de cifrado utiliza la misma clave para cifrar y descifrar los datos, y aplica esa clave a los bloques de datos de longitud fija.

Programa clave

La rutina de cifrado de AES almacena la clave de cifrado principal en una matriz. Una matriz es un grupo de objetos con los mismos atributos que pueden ser abordados de forma individual. La matriz consta de cuatro filas, conteniendo cada uno cuatro, seis u ocho bytes, dependiendo del tamaño de la clave. Tras bambalinas, la rutina de cifrado

utiliza esta matriz para generar una tabla, conocida como un programa clave, que contiene varias claves. Estas claves se denominan claves redondas para distinguirlas de la llave maestra original.

Matriz de estado

El AES utiliza una clave de cifrado que puede ser 128, 192 o 256 bits de largo, y se aplica en unidades de datos, llamados bloques, cada uno de los cuales es de 128 bits de largo. El algoritmo AES comienza copiando cada bloque de 16 bits en una matriz bidimensional llamada el Estado, para crear una matriz de bytes de 4x4. El algoritmo realiza una operación exclusiva "O" que devuelve "verdadero" si uno u otro de sus operandos es verdadero. Esto se conoce como "AddRoundKey", y está entre las primeras cuatro filas del programa clave y la matriz de Estado.

Operaciones matemáticas

Tras la operación inicial exclusivo "O", el algoritmo de cifrado AES entra en su bucle principal, en el que realiza repetidamente cuatro operaciones matemáticas diferentes en la matriz de Estado: "SubBytes", "ShiftRows", "MixColumns" y "AddRoundKey". Estas operaciones emplean una combinación de suma, multiplicación, rotación y sustitución para cifrar cada byte en la matriz de Estado. El bucle principal se ejecuta 10, 12 o 14 veces dependiendo del tamaño de la clave de cifrado. Una vez que se completa la ejecución, el algoritmo copia la matriz de estado a su salida en forma de texto cifrado.

2.3. MODOS DE CIFRADO

Los algoritmos de cifrado por bloque pueden ser ejecutados de diferentes modos. Mostramos ahora los modos más extendidos. Supondremos que el alfabeto de nuestro bloque a cifrar es Σ y que la longitud del bloque es. Suponemos que el algoritmo de cifrado es, que el de descifrado es, que cada bloque de texto plano lo llamamos, y cada bloque de texto cifrado .

Modo de cifra ECB. electronic codebook mode.

En este modo, el texto plano se descompone en bloques de longitud. Si es necesario, al texto plano se le añade un suplemento para conseguir que su longitud sea divisible por. En este modo, cada bloque de longitud es cifrado de forma independiente al resto de bloques: el texto cifrado es una secuencia de los bloques cifrados. Y el descifrado se realiza aplicando el algoritmo inverso a cada bloque del criptograma, también de forma independiente al resto de criptogramas.

Tenemos, pues, que:

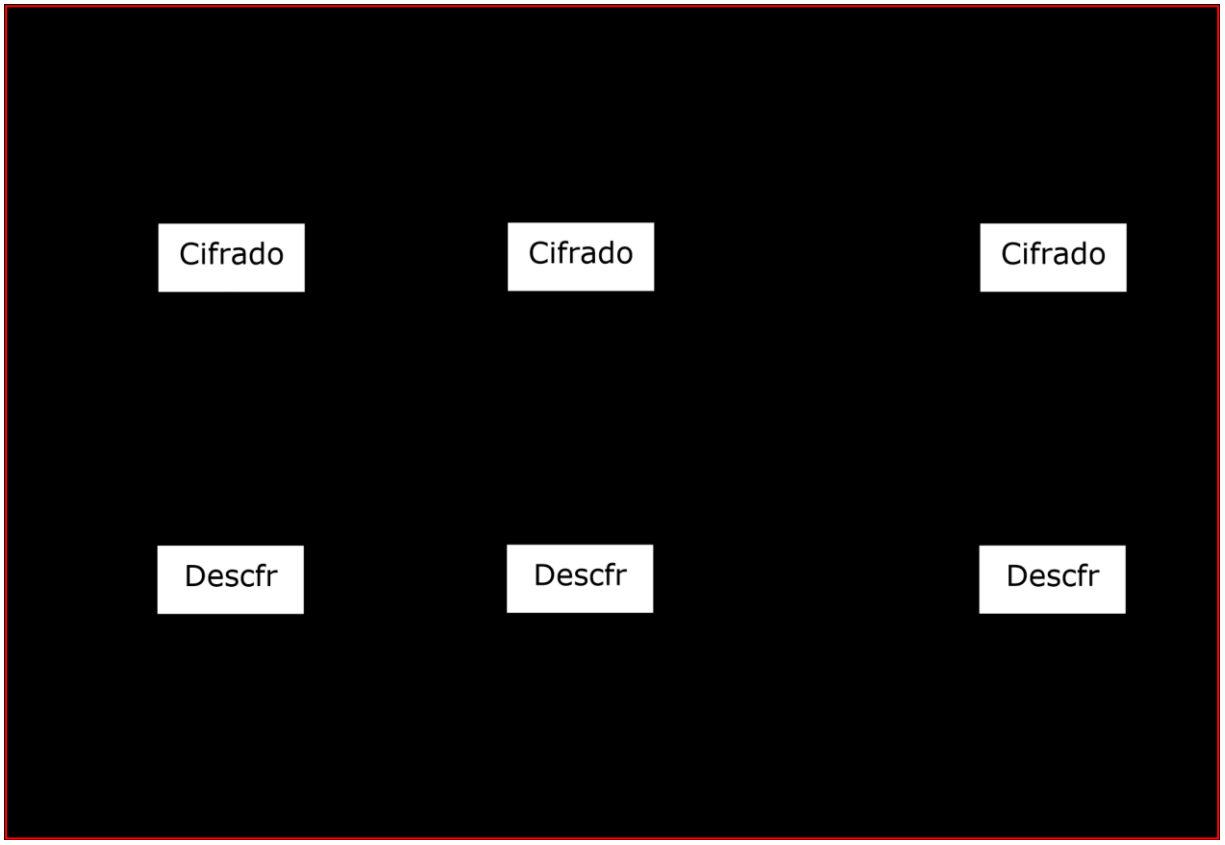
$$c_j = E_K[m_j], \text{ para } j = 1, 2, \ldots, l.$$

Y para descifrar tenemos que:

$$m_j = D_K[c_j].$$

Este modo se emplea para el **envío de valores sencillos**. Pero es un modo que tiene ciertas debilidades a tener en cuenta:

Cuando se usa este modo, a iguales bloques de texto plano se obtienen iguales bloques de texto cifrado. Es así posible reconocer algunos patrones del texto plano en el texto cifrado. Eso facilita un **ataque estadístico**.

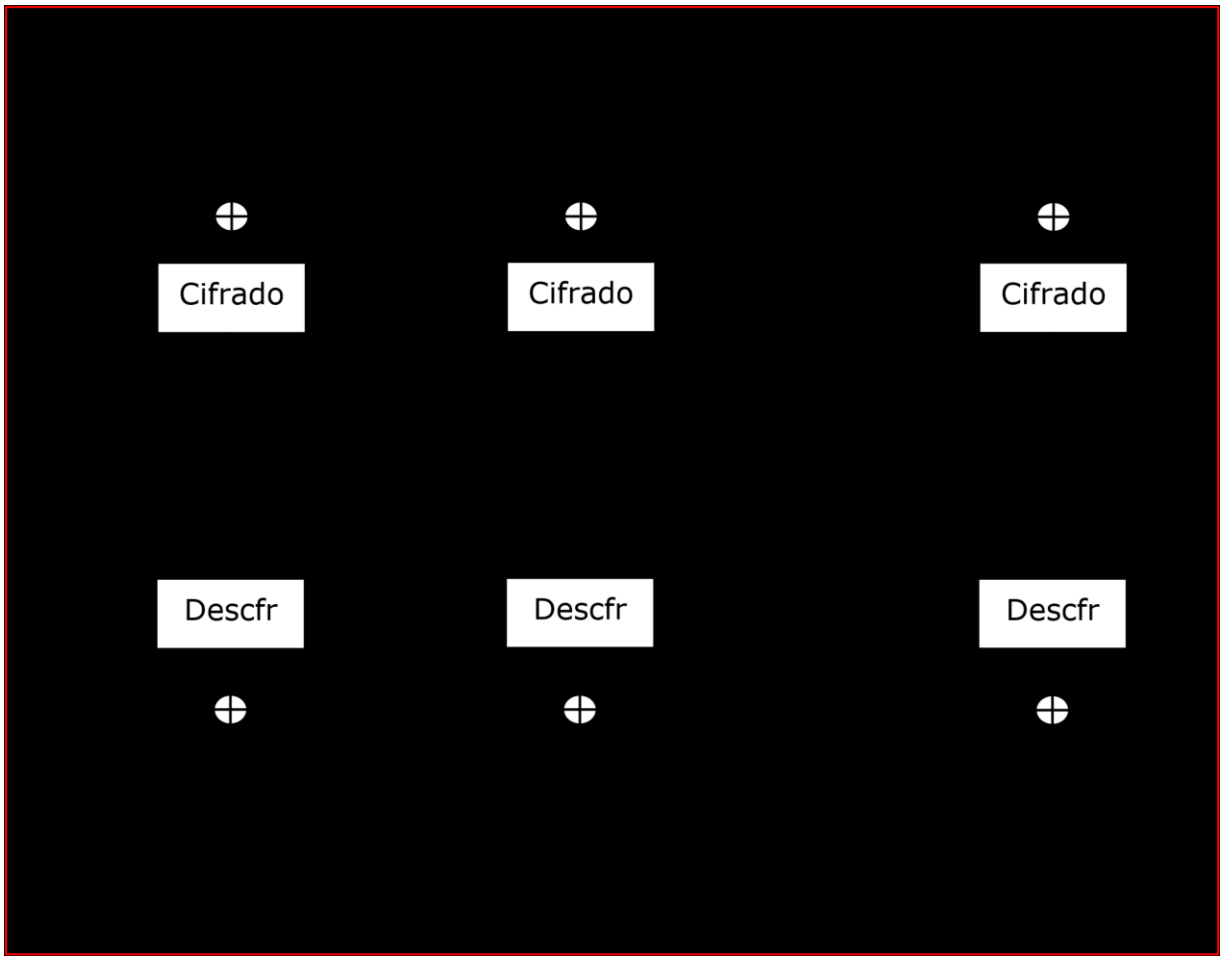


Otra vulnerabilidad de este modo de cifra es que un atacante puede **sustituir** algunos bloques del texto cifrado con otros bloques cifrados que hayan sido cifrados con la misma clave. Esta manipulación es difícil de detectar en el receptor. Por eso, ECB no se usa para el cifrado de textos planos largos.

Un modo de incrementar la seguridad de este modo de cifra es que cada bloque de texto a cifrar esté formado por un determinado número de caracteres del texto plano y otros hasta lo ocupen caracteres aleatorios. Pero eso exige la generación y el uso de muchos caracteres aleatorios y aumenta el número de bloques a cifrar, reduciendo así la eficiencia del procedimiento.

Modo de cifra CBC. cipherblock chaining mode.

En este modo, la entrada al algoritmo de cifrado es el resultado de la operación XOR entre el actual bloque de texto plano a cifrar y el bloque de texto cifrado precedente. Se emplea la misma clave en cada bloque.



Para descifrar, cada bloque cifrado es procesado por el algoritmo de descifrado, y el resultado es sometido a la operación XOR con el bloque cifrado precedente, para obtener así el bloque de texto plano.

Tenemos entonces que:

$$c_j = E_K[c_{j-1} \oplus m_j] \quad \text{para} \quad j = 2, 3, \ldots, l.$$

Y para descifrar tenemos que:

$$D_K[c_j] = D_K\left[E_K[c_{j-1} \oplus m_j]\right] = c_{j-1} \oplus m_j,$$

por lo cual, finalmente tenemos que:

$$\boldsymbol{c_{j-1} \oplus D_K[c_j] = c_{j-1} \oplus c_{j-1} \oplus m_j = m_j.}$$

El primer bloque (J=1) no tiene bloque cifrado previo. Para la generación del primer bloque de texto cifrado se introduce un **vector de inicialización** (lo llamamos IV) que es el que se va a operar con el operador XOR ("xorear"), con el primer bloque de texto plano. Para el descifrado, el IV será "xoreado" con la salida del algoritmo de descifrado, para obtener así el primer bloque de texto plano:

$$\boldsymbol{c_0 = IV,\ c_1 = E_K[c_0 \oplus m_1]\ y\ m_1 = c_0 \oplus D_K[c_1].}$$

El vector IV debe ser conocido por ambas partes: tanto por el emisor que cifra como por el receptor que descifra. Para obtener máxima seguridad, el vector IV puede protegerse como si de una clave se tratara. Esto puede hacerse enviado el emisor al receptor el valor de IV cifrado con el modo ECB. Es conveniente actuar así, porque existen ataques que se basan en el conocimiento del vector IV.

El modo CBC evita los problemas del modo ECB. En este modo, el cifrado de cada bloque no sólo depende de la clave, sino también del bloque previo. Es decir, estamos ante un modo de cifrado dependiente del contexto. Así, bloques iguales, en diferentes contextos, quedan cifrados de forma diferente. El receptor puede darse cuenta de que le han cambiado el texto cifrado porque no obtiene nada en la manipulación del descifrado.

En caso de que se produzca un error en la transmisión de un bloque cifrado (por ejemplo, el bloque cj) entonces pierde sentido el bloque descifrado a partir de este cj y perdemos mj. Como mj+1 depende también del valor de cj, entonces también perderemos ese bloque de texto plano. El bloque mj+2, al depender únicamente de cj+1 y cj+2, y tener éstos correctamente transmitidos y recibidos, sí se puede obtener correctamente. Y así también con los bloques sucesivos.

Así pues, este modo es apropiado para cifrar mensajes de longitud bastante mayor que (tamaño del bloque en el sistema de cifrado que estemos usando).

Modo de cifra CFB. cipher feedback mode.

CBC es un modo válido para cifrar mensajes largos. Pero en aplicaciones de tiempo real (por ejemplo, que el receptor quiera descifrar el mensaje a medida que lo vaya recibiendo) se encuentran, en la práctica, problemas de eficiencia. Eso es necesario, por ejemplo, en comunicaciones telefónicas: el emisor cifra el bloque actual y lo envía, y el receptor lo descifra en cuanto lo recibe. Es decir, las funciones de cifrado y descifrado se utilizan secuencialmente y no simultáneamente. Además, cuanto más computacionalmente complejo sea el proceso de cifrado y descifrado, más tiempo pasa entre el cifrado y el descifrado.

En el caso del modo CFB, este procedimiento se hace de forma diferente. Ahora la función de cifrado no se usa directamente para cifrar bloques de texto plano, sino para **generar una secuencia de bloques de clave**. El texto plano se cifra sumándole módulo 2 el bloque de clave (o lo que es lo mismo, realizando la operación XOR entre el bloque de texto plano y el bloque de clave generada). Y para descifrar el bloque se vuelve a "xorear" el bloque cifrado con el correspondiente bloque de clave. Los bloques de clave pueden ser generados simultáneamente por el emisor y por el receptor; únicamente la operación XOR sí se realiza simultáneamente.

De nuevo, necesitamos un vector de inicialización . $IV \in \{0,1\}^n$. También necesitamos un entero positivo r, tal $1 \leq r \leq n$ que (un valor habitual es r = 8: si n = 64, como es el caso de DES, entonces cada bloque de texto plano se trocearía en 8 sub-bloques). El texto plano queda descompuesto en bloques de longitud

(supongamos que tenemos bloques). **Para cifrar los mensajes planos,** m1… mu, hacemos:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = E_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $c_j = m_j \oplus t_j$.
    d. $I_{j+1} = 2^r I_j + c_j \ mod \ 2^n$: I_{j+1} se genera eliminando los primeros r bits de I_j y sumando c_j.

El texto cifrado queda formado por la secuencia c1, c2,…,cn.

Para el descifrado, el procedimiento a seguir es similar. Ahora el receptor hace los siguientes pasos:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = D_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $m_j = c_j \oplus t_j$.
    d. $I_{j+1} = 2^r I_j + c_j \ mod \ 2^n$.

Tanto el emisor como el receptor pueden ponerse al cálculo de la cadena tj+1tan pronto como ambos conocen el sub-bloque de texto cifrado cj. Así, el bloque de clave t1 puede calcularse simultáneamente en el emisor en el receptor. El emisor genera el sub-bloque

$c_1 = m_1 \oplus t_1$

de texto cifrado y lo envía al receptor. El cálculo de m1 es rápido ya que no es más que una simple operación XOR. Ahora tanto el emisor como el receptor pueden calcular la cadena de clave t2, etc.

Con este modo logramos que la transmisión sea más rápida, pero por otro lado hay que aplicar con mucha mayor frecuencia el algoritmo de cifrado y de descifrado. **El valor de r es un valor de conveniencia entre las velocidades de computación y de transmisión**.

En este modo de transmisión un error en un sub-bloque estropea la labor de descifrado mientras que ese tramo forma parte del vector Ij.

Veámoslo: Supongamos que r= 8 y n=64. Supongamos que el emisor transmite los sub-bloques de texto cifrado c1, c2, … ck…, y supongamos que se corrompe durante la transmisión, de forma que los bloques recibidos son, c1, c2…

Al descifrar, tomaremos c1 y produciremos un subbloque de texto plano erróneo: una versión de m1 con bits erróneos en las posiciones donde c1 tenga bits erróneos.

Ahora, después de descifrar este sub-bloque primero, el receptor construirá un valor equivocado de I2 (lo llamaremos I2) Si I1 era (*, *, *, *, *, *, *, *) entonces será (*, *, *, *, *, *, *, c1). Cuando el receptor reciba el subgrupo no corrupto c2 y lo descifre, obtendrá una versión errónea de m2. Cuando forme I3, obtendrá de la forma (*, *, *, *, *, *, c1, c2). El receptor repetirá este proceso, y seguirá obteniendo versiones erróneas de m1, m2…mn. Cuando calcule entonces el valor de I9, el bloque erróneo se habrá trasladado a la primera posición de la cadena: I9= (c1, c2… c8). Al siguiente paso, el error quedará perdido, y el valor de será I10 ya el correcto.

Modo de cifra OFB. output feedback mode.

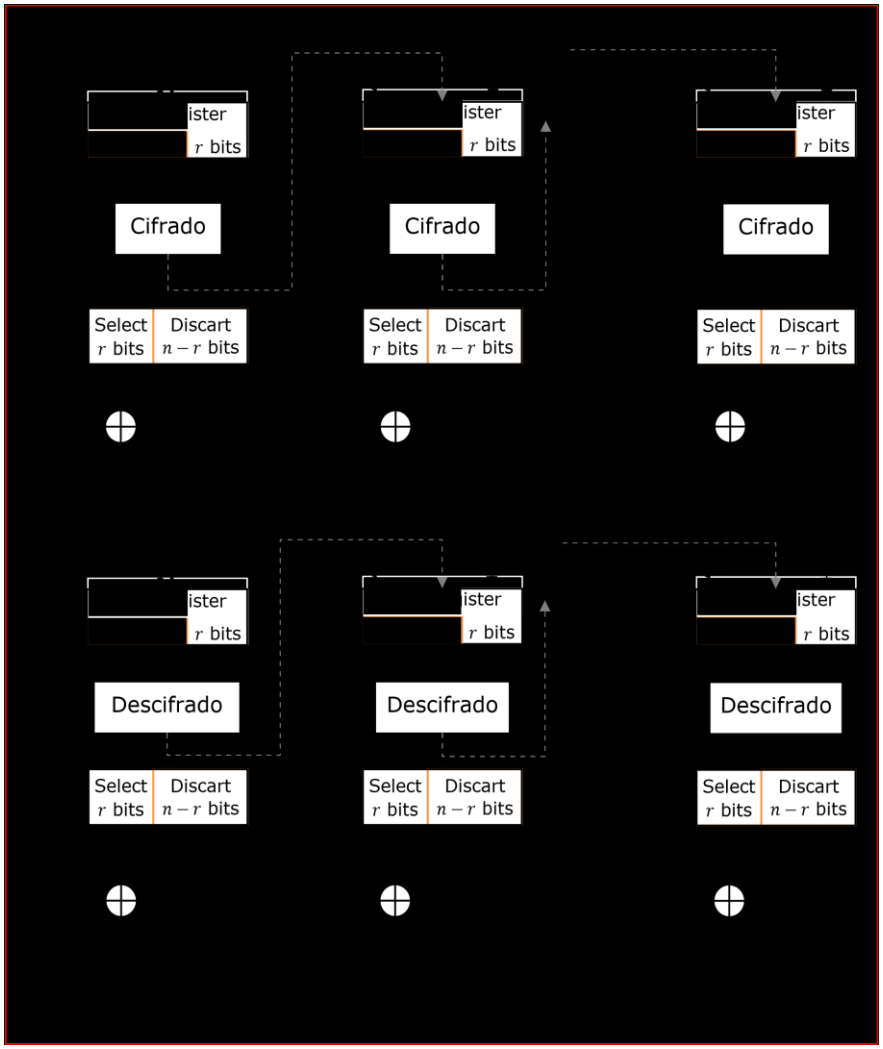
Es un modo muy parecido al CFB. Al igual que en CFB, el modo OFB usa un bloque cifrado de longitud r, otra longitud de $1 \leq r \leq n$ bloque, con y un vector de inicialización IV.

Si el emisor cifra su mensaje con la clave e, entonces, al igual que con CFB, deberá descomponer cada bloque en sub-bloques de longitud r.

Y entonces, se deben realizar las siguientes operaciones:

1. $I_1 = IV$.
2. Para $1 \leq j \leq u$ hacer:
    a. $O_j = E_k(I_j)$.
    b. Construimos la cadena t_j, que consiste en los r bits más significativos de O_j.
    c. $c_j = m_j \oplus t_j$.
    d. $I_{j+1} = O_j$. Otra posibilidad: $I_{j+1} = 2^r I_j + t_j \ mod \ 2^n$

De nuevo, la tarea de descifrado es similar: simplemente hay que cambiar el paso 2. c por $m_j = c_j \oplus t_j$.



Si un bit del texto cifrado se transmite de forma incorrecta, entonces el texto plano estará erróneo exactamente en la misma posición. Un error en un bit no tiene influencia.

La clave de bloque tj sólo depende del vector de inicialización IV, y no de la clave k.

Dichos bloques pueden ser calculados simultáneamente por el receptor y por el emisor: eso supone una mejora frente al modo CFB. Sin embargo, el cifrado de un bloque del texto plano en el modo OFB no depende de los bloques de texto plano previos, sino sólo de su posición. Por eso, la manipulación del texto cifrado resulta aquí más sencilla que en el modo CFB.

Una ventaja del modo OFB es que no se transmiten los errores de transmisión en un bit: si ocurre un error en un bit de c1, entonces ese error sólo afecta al valor de m. Las siguientes sub-secuencias o sub-bloques de texto plano no se ven afectadas por ese error.

Con CFB, como ya vimos, un error se propaga en los siguientes sub-bloques de texto plano descifrado. La desventaja del modo OFB es que vuelve a ser vulnerable a ataques por modificación de la cadena del mensaje.

2.4. OTROS CIFRADOS

Cifrado por transposición o permutación

Cada letra (o carácter) se intercambia por otra del mensaje, reordenando de algún modo las letras, pero no *disfrazándolas*. Para este tipo de cifrado se usan multitud de métodos, como colocar las letras en una matriz de una manera y *sacarlas* de otra manera diferente.

Cifrado Vernam

Según el principio de Kerkhoff todos los algoritmos de cifrados y descifrados deben ser públicos y conocidos por todos, lo único secreto es la clave del algoritmo, esta clave se convierte en la piedra angular del algoritmo.

Basándose en este principio, el cifrado perfecto **(el cifrado Vernam)** debe ser público con su clave en secreto y ésta debe tener la misma longitud del mensaje, ser generada aleatoriamente y solamente puede ser usada una sola vez.

Para cifrar el mensaje se realiza una operación XOR *(or exclusivo)* entre el mensaje y la clave.

Como se puede observar este método sería perfecto de no ser porque cada clave generada aleatoriamente debería ser generada también aleatoriamente e idéntica a la del emisor, por el receptor del mensaje, algo que en principio es muy difícil.

2.5. CRIPTOANÁLISIS

Criptoanálisis como el estudio de los métodos para obtener sentido a un mensaje cifrado, también es conocido como la ciencia opuesta a la Criptografía. Estos métodos se traducen típicamente en conseguir la clave secreta con la cuál fue cifrado el mensaje.

Criptoanálisis

Es la ciencia opuesta a la criptografía quizás no es muy afortunado hablar de ciencias opuestas, sino más bien de ciencias complementarias, ya que, si ésta trata principalmente de crear y analizar criptosistemas seguros, la primera intenta romper esos sistemas, demostrando su vulnerabilidad: dicho de otra forma, trata de descifrar los criptogramas. El término descifrar siempre va acompañado de discusiones de carácter técnico, aunque asumiremos que descifrar es conseguir el texto en claro a partir de un criptograma, sin entrar en polémicas de reversibilidad y solidez de criptosistemas.

Análisis de las debilidades

En el análisis para establecer las posibles debilidades de un sistema de cifrado, se han de asumir las denominadas condiciones:

- El criptoanalista tiene acceso completo al algoritmo de encriptación
- El criptoanalista tiene una cantidad considerable de texto cifrado
- El criptoanalista conoce el texto en claro de parte de ese texto cifrado. También se asume generalmente el Principio de Kerckhoffs, que establece que la seguridad del cifrado ha de residir exclusivamente en el secreto de la clave, y no en el mecanismo de cifrado.

Objetivos de ataques

Aunque para validar la robustez de un criptosistema normalmente se suponen todas las condiciones del peor caso, existen ataques más específicos, en los que no se cumplen todas estas condiciones. Cuando el método de ataque consiste simplemente en probar

todas y cada una de las posibles claves del espacio de claves hasta encontrar la correcta, nos encontramos ante un ataque de fuerza bruta o ataque exhaustivo.

Si el atacante conoce el algoritmo de cifrado y sólo tiene acceso al criptograma, se plantea un ataque sólo al criptograma; un caso más favorable para el criptoanalista se produce cuando el ataque cumple todas las condiciones del peor caso; en este caso, el criptoanálisis se denomina de texto en claro conocido.

Si además el atacante puede cifrar una cantidad indeterminada de texto en claro al ataque se le denomina de texto en claro escogido; este es el caso habitual de los ataques contra el sistema de verificación de usuarios utilizado por Unix, donde un intruso consigue la tabla de contraseñas generalmente /etc/passwd y se limita a realizar cifrados de textos en claro de su elección y a comparar los resultados con las claves cifradas a este ataque también se le llama de diccionario, debido a que el atacante suele utilizar un fichero `diccionario' con los textos en claro que va a utilizar. El caso más favorable para un analista se produce cuando puede obtener el texto en claro correspondiente a criptogramas de su elección; en este caso el ataque se denomina de texto cifrado escogido.

Algoritmo de cifrado

El algoritmo de cifrado, para ser considerado seguro, ha de soportar todos estos ataques y otros no citados; sin embargo, en la criptografía, como en cualquier aspecto de la seguridad, informática o no, no debemos olvidar un factor muy importante: las personas. El sistema más robusto caerá fácilmente si se tortura al emisor o al receptor hasta que desvelen el contenido del mensaje, o si se le ofrece a uno de ellos una gran cantidad de dinero; este tipo de ataques (sobornos, amenazas, extorsión, tortura...) se consideran siempre los más efectivos.

Utilidades

Criptoanálisis también se utiliza para referirse a cualquier intento de sortear la seguridad de otros tipos de algoritmos y protocolos criptográficos en general, y no solamente el cifrado. Sin embargo, el criptoanálisis suele excluir ataques que no tengan como objetivo primario los puntos débiles de la criptografía utilizada; por ejemplo, ataques a la seguridad que se basen en el soborno, la coerción física, el robo, el keylogging y demás, aunque estos tipos de ataques son un riesgo creciente para la seguridad informática, y se están haciendo gradualmente más efectivos que el criptoanálisis tradicional.

Criptoanálisis y Ataques a Criptosistemas

El criptoanálisis es el arte de descifrar comunicaciones encriptadas sin conocer las llaves correctas. Existen muchas técnicas criptoanalíticas. Algunas de las más importantes se describen a continuación.

Ataques a textos cifrados Ciphertext-only attack

Esta es la situación en la cual el atacante no conoce nada sobre el contenido del mensaje, y debe trabajar solo desde el texto cifrado. En la práctica es muy probable hacer tantas conjeturas acerca del texto plano, como cantidad de tipos de mensajes tengan un encabezado similar.

Incluso las cartas y los documentos ordinarios comienzan de una manera muy previsible. Por ejemplo, muchos ataques clásicos utilizan análisis frecuencial del texto cifrado, sin embargo, no funciona bien contra los cifrados modernos.

Los criptosistemas modernos no son débiles contra ataques de texto cifrado, aunque algunas veces son considerados con el agregado de que el mensaje contiene "tendencia" estática.

Ataques de texto plano conocidos

El atacante conoce o puede adivinar el texto de alguna parte del texto cifrado. La tarea es desencriptar el resto del bloque cifrado utilizando esta información. Esto puede ser hecho determinando la clave utilizada para encriptar la información, o a través de algún atajo Uno de los mejores ataques modernos de texto plano conocido es el criptoanálisis lineal contra cifradores de bloques.

Ataques de texto plano seleccionado

El atacante puede tener cualquier texto encriptado con una llave desconocida. La tarea es determinar la llave utilizada para encriptar. Un buen ejemplo de este ataque es el criptoanálisis diferencial que puede ser aplicado a cifradores de bloques y, en algunos casos, a funciones Hash
Algunos criptosistemas, particularmente el RSA, son vulnerables a estos ataques. Cuando tales algoritmos son utilizados, se debe tener cuidado en el diseño de la aplicación (o protocolo) de forma tal que un atacante no pueda obtener el texto encriptado.

Ataque del hombre en el medio

Este ataque es relevante para las comunicaciones criptográficas y los protocolos de intercambio de llaves. La idea es que cuando dos partes, A y B, están intercambiando llaves por comunicaciones seguras por ejemplo utilizando Diffie-Hellman, un adversario intruso se posiciona entre A y B en la línea de comunicación. El intruso intercepta las señales que A y B se envían, y ejecuta un intercambio de llaves entre A y B. A y B

terminaran utilizando llaves diferentes, cada una de las cuales es conocida por el intruso. El intruso puede luego desencrespar cualquier comunicación de A con la llave que comparte con A, y luego reenviarla a B encrestándola nuevamente con la llave que comparte con B. Ambos A y B pensarán que se están comunicando en forma segura, pero de hecho el intruso está escuchando todo.

Como Prevenir

La forma habitual de prevenir este ataque es utilizar un sistema de clave pública capaz de proveer firmas digitales.

- Por configuración, las partes deben conocer de antemano la clave pública de cada una de ellas.
- Después de que han sido generadas, las partes se envían firmas digitales.
- El hombre de por medio falla en el ataque a causa de que no es capaz de falsificar las firmas sin conocer las llaves privadas utilizadas para generar las firmas.
- Este medio es suficiente si existe también una manera segura de distribuir claves públicas. *Una forma es la jerarquía de certificados como X.509. Es utilizado por ejemplo en IP Sec.

La correlación entre la clave secreta y la salida del criptosistema es la fuente principal de información para el criptoanálisis. En el caso más simple, la información sobre la llave secreta es filtrada por el criptosistema. Casos más complicados requieren estudios sobre la correlación básicamente, cualquier relación que no sería esperada entre la información observada o tomada de los criptosistemas y la información de la llave adivinada.

Teorías

Por ejemplo, en ataques lineales contra bloques cifrados el criptoanálisis estudia el texto plano conocido y el observado. Adivinando algunos bits del criptosistema el analista determina, por correlación entre el texto plano y el cifrado, si el "adivinó bien". Esto se puede repetir y tiene muchas variantes.

El criptoanálisis diferencial introducido por Eli Biham y Adi Shamir en los '80 fue el primer ataque que utilizó completamente esta idea contra los bloques cifrados. Más tarde Eli Biham y Adi Shamir introducen el criptoanálisis lineal que fue aún más efectivo contra el DES. Más recientemente, se han desarrollado nuevos ataques que utilizan ideas similares. La idea correlacionar es fundamental para la criptografía y muchas investigaciones han tratado de construir criptosistemas que sean seguros contra tales ataques.

Por ejemplo, Knudsen y Nyberg han estudiado esta seguridad contra el criptoanálisis diferencial.

Ataques contra el hardware o utilizando el hardware base

Así como en los últimos años más y más dispositivos pequeños de criptografía han sido ampliamente utilizados, una nueva categoría de ataques se ha hecho relevante la cual apunta directamente a las implementaciones de hardware de los criptosistemas. Los ataques utilizan datos muy buenos obtenidos del dispositivo criptográfico, supongamos, información de la encriptación y de la llave de estas medidas. Las ideas básicas están estrechamente relacionadas con aquellos en otros ataques correlacionados. Por ejemplo, el atacante adivina algunos bits de la clave y trata de verificar la exactitud de lo adivinado estudiando la correlación contra lo que el adivinó.

Se han propuesto varios ataques como la utilización cuidadosa del cronometraje del dispositivo, medidas del consumo de energía, y patrones de radiación. Estas mediciones pueden ser utilizadas para obtener la llave secreta u otro tipo de información almacenada en el dispositivo
Estos ataques son independientes de los algoritmos criptográficos utilizados y pueden ser aplicados a cualquier dispositivo que no esté explícitamente protegido.

Las Fallas en los Criptosistemas

Logran conducir al criptoanálisis y aún al descubrimiento en de la llave secreta. El interés en dispositivos criptográficos conduce al descubrimiento de que algunos algoritmos se comportan muy mal con la introducción de una pequeña falla en el cálculo interno, Por ejemplo, la implementación usual de una operación de llave privada RSA es susceptible a los ataques de fallas. Se ha sido demostrado que causando un bit de error en un punto adecuado puede revelar la factorización del módulo revela la llave privada.

Se han aplicado ideas similares a una gran variedad de algoritmos y dispositivos. Es así necesario que los dispositivos criptográficos sean diseñados para ser altamente resistentes a fallas y contra introducciones maliciosas de fallas por criptoanálisis.

Informática o cálculo Cuántico

Escritos de Peter Shor sobre factores polinómico del tiempo y algoritmos logarítmicos discretos con informática cuántica han causado el creciente interés en la informática cuántica. La informática cuántica es un campo reciente de investigación que utiliza mecanismos cuánticos para construir computadoras que son, en teoría, más potentes que las modernas computadoras seriales. El poder es derivado del paralelismo inherente de los mecanismos cuánticos. Así que, en lugar de hacer una tarea a la vez, como lo hacen los mecanismos seriales, las computadoras cuánticas pueden ejecutarlas todas al mismo tiempo. De esa manera se espera que con las computadoras cuánticas podamos resolver problemas que no son resueltos por los seriales
Los resultados de Shor implican o sugieren que si las computadoras cuánticas pueden ser implementadas eficientemente entonces muchas de las llaves públicas criptográficas serán historia. Sin embargo, son mucho menos efectivas contra llaves criptográficas secretas
El estado de arte actual de las computadoras cuánticas no aparenta ser alarmante, ya que solo se han implementado máquinas muy pequeñas. La teoría de la informática cuántica brinda más promesas en cuanto al rendimiento que las computadoras seriales, sin embargo, que se realice en la práctica es una cuestión pendiente.

Mecanismos Cuánticos

Son también un recurso para una nueva forma de ocultamiento de datos y comunicaciones seguras con el potencial de brindar una seguridad inquebrantable, este es el campo de la criptografía cuántica. A diferencia de la informática cuántica, se han logrado muchas implementaciones experimentales exitosas de la criptografía cuántica. Aun así, la criptografía cuántica está de alguna manera lejos de ser implementada en aplicaciones comerciales.

Criptografía DNA

Leonard Adleman (uno de los inventores del RSA) trajo a colación la idea de utilizar DNA como computadoras. Las moléculas de DNA pueden ser vistas como una gran computadora capaz de realizar ejecuciones en paralelo. Esta naturaleza concurrente podría brindar a las computadoras DNA incrementos exponenciales de velocidad contra las computadoras
Desafortunadamente hay problemas con las computadoras DNA, el crecimiento exponencial de la velocidad implica también la necesidad de crecimiento del volumen de material requerido. De esa manera las computadoras DNA tienen limitaciones de rendimiento en la práctica. Además, no es muy fácil construir una.

Hay muchas otras técnicas y ataques criptográficos. Sin embargo, estas son las más importantes para el diseño de aplicaciones. Cualquiera que contemple el diseño de un nuevo criptosistema.

2.6. CRIPTOGRAFIA SIMETRICA O CRIPTOGRAFIA DE UNA CLAVE

La criptografía simétrica es la técnica criptográfica más antigua que existe, pero sigue ofreciendo un alto nivel de seguridad. Se basa en la utilización de una única clave secreta que se encargará de cifrar y descifrar la información, ya sea información en tránsito con protocolos como TLS, o información en un dispositivo de almacenamiento extraíble. La

criptografía simétrica fue el primer método empleado para el cifrado de la información, se basa en que se utilizará la misma contraseña tanto para el cifrado como el descifrado, por tanto, es fundamental que todos los usuarios que quieran cifrar o descifrar el mensaje, tengan esta clave secreta, de lo contrario, no podrán hacerlo. Gracias a la criptografía simétrica, podremos realizar comunicaciones o almacenar archivos de forma segura.

1.7. CHACHA20

El algoritmo ChaCha20 es un algoritmo de cifrado simétrico que **soporta claves de 128 y 256 bits** y de alta velocidad, a diferencia de AES que es un cifrado por bloques, ChaCha20 es un cifrado de flujo. Tiene características similares a **su predecesor Salsa20** pero con una función primitiva de 12 o 20 rondas distintas. Su código fue publicado, estandarizado por la IETF en la RFC 7539 y en implementaciones de software, es mucho más eficiente y rápido que AES, por lo que rápidamente se ha hecho un hueco dentro de los algoritmos más usados en la actualidad.

Para saber por qué se ha hecho tan famoso, vamos a meter a Google de por medio para que se pueda entender todo mucho más rápido. Las conexiones HTPPS están enfocadas a ofrecer la máxima seguridad en las webs que visitamos todos los días, fue el siguiente paso del protocolo HTTP el cual no tenía protección alguna. El cifrado, sin embargo, varía de un navegador a otro. Hasta hace algunos años, Chrome para Android ha estado utilizando AES-GCM como algoritmo de cifrado simétrico, sin embargo, Google lleva trabajando desde hace muchos años en cifrados más actuales, seguros y rápidos.

2.8. CIFRADO TWOFISH

Lo primero que vamos a ver acerca de este cifrado, es de qué se compone. Estamos ante un cifrado de clave simétrica, que dispone de un tamaño de bloque de 128 bits, lo que puede verse a priori, como muy seguro ante ataques de fuerza bruta, al requerir grandes capacidades de procesamiento para poder descifrarlo. La longitud de su clave puede variar entre los 128, 192 o 256 bits. Es de código abierto, y su uso es totalmente gratuito.

2.9. CRIPTOGRAFIA DE CLAVE PÚBLICA

La criptografía de clave asimétrica también es conocida como clave pública, emplea dos llaves diferentes en cada uno de los extremos de la comunicación para cifrarla y descifrarla. Cada usuario de la comunicación tendrá una clave pública y otra privada. La clave privada tendrá que ser protegida y guardada por el propio usuario, será secreta y no la deberá conocer absolutamente nadie ni tampoco debe ser enviada a nadie. La clave pública será accesible por todos los usuarios del sistema que quieran comunicarse.

La fortaleza del sistema por el cual es seguro este tipo de algoritmo asimétrico, es que está basado en funciones matemáticas las cuales son fáciles de resolver en un sentido, pero que su resolución en sentido contrario es extremadamente complicada, a menos que se conozca la clave. Las claves públicas y privadas se generan simultáneamente y están ligadas la una a la otra. La relación entre ambas debe ser muy compleja, para que resulte muy difícil que obtengamos una clave a partir de la otra, en este caso, que obtengamos la clave privada puesto que la pública la conoce toda persona conectada al sistema.

1.10. DESAFIO – RESPUESTA

Para aumentar la seguridad, este método comprueba que el emisor es realmente quien dice ser, para ello se envía un texto al emisor y éste lo cifrará con su clave privada (lo que está haciendo realmente es firmarlo), el emisor nos enviará el texto cifrado (firmado) y nosotros descifraremos la clave (comprobaremos la firma) aprovechando que tenemos la clave pública del emisor, y por último, compararemos que el mensaje obtenido sea el mismo que enviamos anteriormente.

Si algún usuario se hace pasar por el emisor real, no tendría la clave privada por lo que el «desafío» no hubiera resultado satisfactorio y no se establecería la comunicación de los datos.

1.11. DIFFIE – HELLMAN

No es un algoritmo asimétrico propiamente dicho, es un protocolo de establecimiento de claves, se usa para generar una clave privada a ambos extremos de un canal de comunicación inseguro**.** Se emplea para obtener la clave privada con la que posteriormente se cifrará la información junto con un algoritmo de cifrado simétrico. El punto fuerte del Diffie-Hellman es que, su seguridad radica en la dificultad de calcular el logaritmo discreto de números grandes (Diffie-Hellmann también permite el uso de curvas elípticas).

UNIDAD III

ALGORITMOS DE CLAVE PÚBLICA Y FIRMAS DIGITALES

El protocolo que sigue este mecanismo es una combinación de hardware y software, políticas y procedimientos de seguridad que permiten la ejecución con garantías de operaciones criptográficas como el cifrado, la firma digital o el no repudio de transacciones electrónicas.

Un ejemplo muy simple de cómo funciona este proceso sería que al enviar una información importante pero el mensajero resulta inseguro para nosotros, escribimos el mensaje en un papel, se introduce en una caja de metal con un candado y se envía. La caja llega a su destino sin problema, pero el destinatario no puede leerla, pues no puede abrir el candado.

Lo primordial sería enviarle la llave, pero, aunque sea por otro medio u otro mensajero, puede que haya otro nivel de riesgo, al comprometer la seguridad de la llave, confiándola a extraños. Así funciona el ciframiento convencional. Cuando decidimos brindar información confidencial, pero queremos que solo el receptor pueda leerla, ahí es cuando entra en juego este método.

A través de un juego entre distintos caracteres informáticos, la **llave pública** solamente puede cifrar.

La **llave privada** puede descifrar o hacer las dos cosas, aunque esto último no es tan importante. Se recibe la llave pública del destinatario y con ella cifra la información que se le enviará. Una vez cifrada, no se puede ver la información. Al enviarla, en un correo, por ejemplo, el destinatario la recibe y la descifra con su llave privada.

Es por ello, que las llaves permiten a los usuarios autenticarse frente a otros usuarios y usar la información de los certificados de identidad (por ejemplo, las claves públicas de otros usuarios) para cifrar y descifrar mensajes, firmar digitalmente información, garantizar el no repudio de un envío, y otros usos. Dependerá en parte importante, cómo se guarden las claves privada, puesto que existen dispositivos especiales denominados tokens de seguridad para facilitar la seguridad de la clave privada, así como evitar que ésta pueda ser exportada. Estos dispositivos pueden incorporar medidas biométricas, como la verificación de huella dactilar, que permiten aumentar la confiabilidad, dentro de las limitaciones tecnológicas, en que sólo la persona dueña del certificado pueda utilizarlo.

3.1. EL ALGORITMO RSA

El método de encriptado de datos conocido como algoritmo RSA, por los nombres de sus inventores (Rivest, Shamir y Adleman) es uno de los más usados hoy día para la transmisión segura de datos a través de canales inseguros. Este documento es una introducción a las bases matemáticas de dicho algoritmo de encriptado escrita para llegar desde unos conocimientos mínimos (el concepto de anillo de los enteros y la existencia y unicidad de la descomposición en factores primos de un entero) hasta la comprensión del algoritmo en sí, tratando de no omitir casi ninguna demostración (aunque algunas se realizarán en la clase, o se dejan como ejercicio, y una de ellas excede el alcance del curso). Está casi ´íntegramente basado en los libros del Prof. Manuel Lucena (Univ. de Jaén) y de los profesores J.M. Basart, J. Rifá y M. Villanueva (Universidad Autónoma de Barcelona). Véase la bibliografía al final.

Anillo de los enteros MD

Un concepto del que se parte es la existencia de un conjunto llamado de los números enteros, en el que está definida una relación de orden total (ser mayor que) y unas operaciones aritméticas (suma y producto, con las definiciones usuales) que le confieren estructura de anillo, es decir:

$Z = \{. . . , -4, -3, -2, -1, 0, 1, 2, 3, . . .\}$

- La operación + en Z es conmutativa, asociativa, tiene elemento neutro (el 0) y todo elemento tiene su simétrico, llamado opuesto. Por tanto, (Z, +) es grupo conmutativo.
- La operación · en Z es conmutativa, asociativa y tiene elemento neutro (el 1). Sin embargo, no todo elemento tiene simétrico, que aquí se llamaría inverso (de hecho, solamente el 1 y el −1 lo tienen, y son ellos mismos).
- La operación · es distributiva respecto a +.

Por tanto, (Z, +, ·) es un anillo conmutativo con elemento unidad, llamado el anillo de los enteros.

Conceptos básicos de aritmética en z

División entera

Dados a, b ∈ Z, diremos que la división de a entre b tiene cociente q y resto r si es cierto que a = bq + r, siendo q un número entero sin restricciones, y r un numero ´ entero comprendido entre 0 y b − 1.

- Ejemplos: 137 dividido entre 21 da como cociente 6 y como resto 11, dado que se puede escribir 137 como 137 = 21 ·6 + 11.
- -137 dividido entre 21 da como cociente -7 y como resto 10, dado que se puede escribir -137 como −137 = 21 · (−7) +10. Nótese que escribir -137 como −137 = 21 · (−6) −11 no cumple los requisitos de la definición de división entera, dado que el supuesto resto, -11, no está entre 0 y 21.

Múltiplos y divisores

Dados dos números enteros a, b ∈ Z, con a ≤ b, se dice que sea múltiplo de b, o equivalentemente, que b es divisor de a, si existe algún entero q ∈ Z tal que a = q · b, o lo que es lo mismo, si al realizar la división entera de a entre b, según´ se ha definido en el punto anterior, el cociente es q y el resto es 0. Es obvio que 1 es divisor de cualquier número ´ entero, dado que ∀a ∈ Z, a = a · 1 con lo que el cociente es el propio a y el resto 0.

Definiciones básicas

Máximo común divisor

Dados n números enteros, {a1, . . ., an} ∈ Z se llama máximo común´ divisor, abreviado m.c.d., al mayor número entero positivo m que es divisor de todos ellos, y se escribe m = mcd (a1, . . . , an). Como 1 es divisor de cualquier número, en ausencia de otro divisor común mayor, 1 sería el m.c.d. de cualquier conjunto de enteros.

Mínimo común múltiplo

Dados n números enteros, {a1, . . ., an} ∈ Z se llama mínimo común múltiplo, abreviado m.c.m., al menor número entero positivo m que es múltiplo de todos ellos, y se escribe m = mcm (a1, . . ., an). Como el producto de un ai cualquiera por cualquier entero (p. ej., por el producto de todo el demás aj) es múltiplo de ai, entonces, en ausencia de otro múltiplo menor, al producto de todos los ai sería el m.c.m. de cualquier conjunto de enteros.

Descomposición en factores primos

Dado un número entero a, siempre se puede escribir de modo único como

$$a = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_n^{k_n} = \prod_{i=1}^{n} p_i^{k_i}$$

siendo toda la pi números primos estrictamente menores que a y ki exponentes naturales. A cada uno de los números primos pi se le llama factor primo de a.

Cálculo del mcd y del mcm por descomposición

Dados dos enteros, a y b:

- Su mcd se puede calcular multiplicando todos los factores primos que aparecen en ambas descomposiciones (los factores primos comunes), elevado cada uno de ellos al menor de los dos exponentes con que aparece.
- Su mcm se puede calcular multiplicando todos los factores primos que aparecen en cualquiera de las descomposiciones (los factores primos comunes, y los no comunes), elevado cada uno de ellos al mayor de los dos exponentes con que aparece. Los factores que sean comunes se multiplicarán una sola vez.

Ejemplos:

Hallar mcd y mcm de 643 y 412. Descomponiendo,

643 = 643 · 1 (este número es primo)

412 = 103 · 2 2 · 1 Luego mcd (643, 412) = 1 y mcm (643, 412) = 643 · 103 · 2 2 · 1 = 264916

Hallar mcd y mcm de 22253 y 4675.

22253 = 172 · 11 · 7 · 1

4675 = 17 · 11 · 5 2 · 1

Luego mcd (22253, 4675) = 17 · 11 · 1 = 187 y mcm (22253, 4675) = 172 · 11 · 7 · 5 2 · 1 = 556325

En una de las demostraciones siguientes (concretamente, el cálculo de la función multiplicativa de Euler) necesitaremos la siguiente proposición, que pasamos a enunciar:

Prop.:MC Si mcd (p, q) = 1, entonces mcm (p, q) = pq

Dem.: Por ser el mcd 1, el 1 es el único factor que aparece en ambas descomposiciones (el único factor común). Esto significa que todos los demás son no comunes, por lo que todos ellos deberán multiplicarse para calcular el mcm. Además, deberán hacerlo con el exponente con el que aparecen, que es el mayor, dado que es el único. Por tanto, las dos descomposiciones aparecen íntegras y multiplicadas en el cálculo del mcm, por tanto, el mcm es la multiplicación de ambos números.

3.2. OTROS ALGORITMOS DE CLAVE PÚBLICA

Un cifrado de clave pública (o asimétrica), es aquel cifrado que se basa en el uso de una pareja de claves, pública y privada, de las cuales una se usa para cifrar y la otra para descifrar.

Ambas claves están relacionadas por una *función trampa*, suele ser una función matemática. Las claves se calculan usando la función y la inversa de ésta, siendo la función inversa la *función trampa* al ser muy difícil o imposible de calcular.

- **Función irreversible**

  $x \in A$, $f(x)$ fácil de calcular

  $y \in f(A)$, $x = f^{-1}(y)$ difícil de calcular
- **Función trampa**

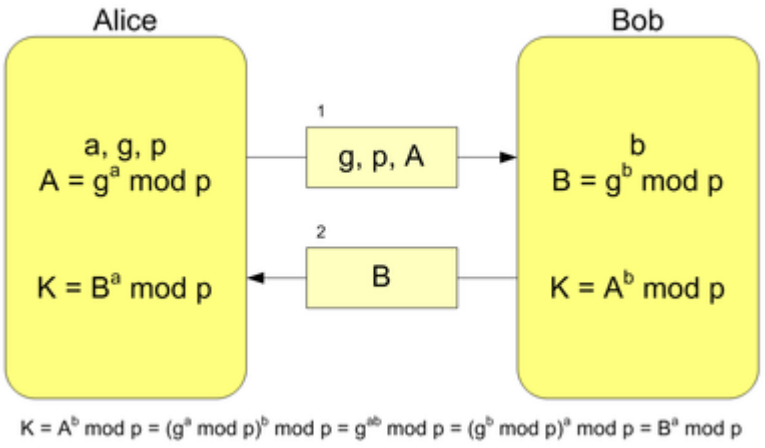
  $x = f^{-1}(y)$ Es calculable conociendo la trampa de la función. Pero sin conocer dicha trampa, $y = f(x)$ es unidireccional.

  Además la trampa sólo se puede calcular con la clave privada.

Algoritmo de Diffie-Hellman

Desarrollado por **Whitfiled Diffie y Martín Hellman** en 1976. Este algoritmo es, básicamente, un protocolo para realizar el intercambio de claves. En realidad, no es un cifrado, se creó para solucionar el problema de los cifrados de clave privada (o simétricos) en el intercambio de claves. El algoritmo consiste en los siguientes pasos:

1. Se establecen un primo «p» y un generador $g \in Z^*_p$. Estos dos valores («g» y «p») son públicos. Siendo Z^* el conjunto de los enteros menores que «p», que son primos relativos de éste y además es un grupo bajo la multiplicación módulo «p».
2. A escoge $x \in Z_{p-1}$ al azar, calcula $X = g^x \pmod p$, y envía X a B.
3. B escoge $y \in Z_{p-1}$ al azar, calcula $Y = g^y \pmod p$, y encía Y a A.
4. A calcula $K = (g^y \bmod p)^x \bmod p$
5. B calcula $K = (g^x \bmod p)^y \bmod p$
6. Siendo la clave «K».



Si alguien interceptara nuestra *conversación*, por ejemplo un usuario malintencionado «E» que poseyera p, g, X e Y, podría calcular el secreto compartido si tuviera también uno de los valores privados (x o y) o lograse invertir la función, pero como se menciona anteriormente estos algoritmos usan *funciones trampa*, ya que para calcular x dado X tenemos que el problema del logaritmo discreto en Z^*_p es **un problema que se cree intratable computacionalmente**.

Este algoritmo se usa(ba) para compartir una clave privada usando unos valores públicos, como veremos en próximos posts los cifrados de clave pública se basan básicamente en métodos semejantes.

3.3. FIRMA DE CLAVES SIMÉTRICAS

Una de las principales ventajas de la criptografía de clave pública es que ofrece un método para el desarrollo de firmas digitales. La firma digital permite al receptor de un mensaje verificar la autenticidad del origen de la información, así como verificar que dicha información no ha sido modificada desde su generación. De este modo, la firma digital ofrece el soporte para la autenticación e integridad de los datos, así como para el no repudio en origen, ya que el originador de un mensaje firmado digitalmente no puede argumentar que no lo es.

Una firma digital está destinada al mismo propósito que una manuscrita. Sin embargo, una firma manuscrita es sencilla de falsificar mientras que la digital es imposible mientras no se descubra la clave privada del firmante.

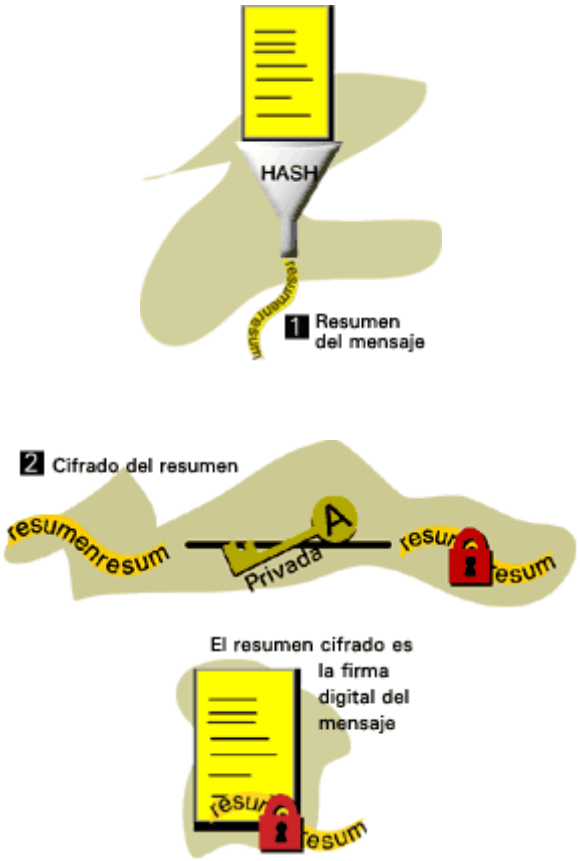
La firma digital se basa en la propiedad ya comentada sobre que un mensaje cifrado utilizando la clave privada de un usuario sólo puede ser descifrado utilizando la clave pública asociada. De tal manera, se tiene la seguridad de que el mensaje que ha podido descifrarse utilizando la clave pública sólo pudo cifrarse utilizando la privada. La firma digital, por tanto, es un cifrado del mensaje que se está firmando, pero utilizando la clave privada en lugar de la pública.

Sin embargo, ya se ha comentado el principal inconveniente de los algoritmos de clave pública: su lentitud que, además, crece con el tamaño del mensaje a cifrar. Para evitar éste problema, la firma digital hace uso de funciones hash. Una función hash es una operación que se realiza sobre un conjunto de datos de cualquier tamaño de tal forma que se obtiene como resultado otro conjunto de datos, en ocasiones denominado resumen de los datos originales, de tamaño fijo e independiente el tamaño original que, además, tiene la propiedad de estar asociado unívocamente a los datos iniciales, es decir, es prácticamente imposible encontrar dos mensajes distintos que tengan un resumen hash idéntico.

Proceso:

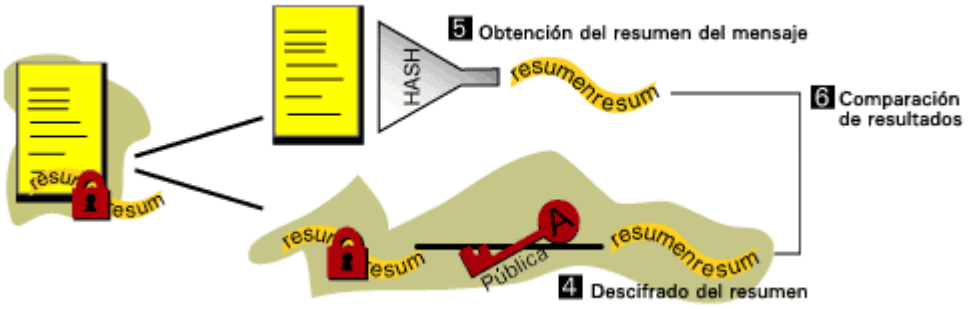
Ana y Bernardo tienen sus pares de claves respectivas.

Ana escribe un mensaje a Bernardo. Es necesario que Bernardo pueda verificar que realmente es Ana quien ha enviado el mensaje. Por lo tanto, Ana debe enviarlo firmado:



1. Resume el mensaje mediante una función hash.
2. Cifra el resultado de la función hash con su clave privada. De esta forma obtiene su firma digital.
3. Envía a Bernardo el mensaje original junto con la firma.

Bernardo recibe el mensaje junto a la firma digital. Deberá comprobar la validez de ésta para dar por bueno el mensaje y reconocer al autor del mismo (integridad y autenticación).

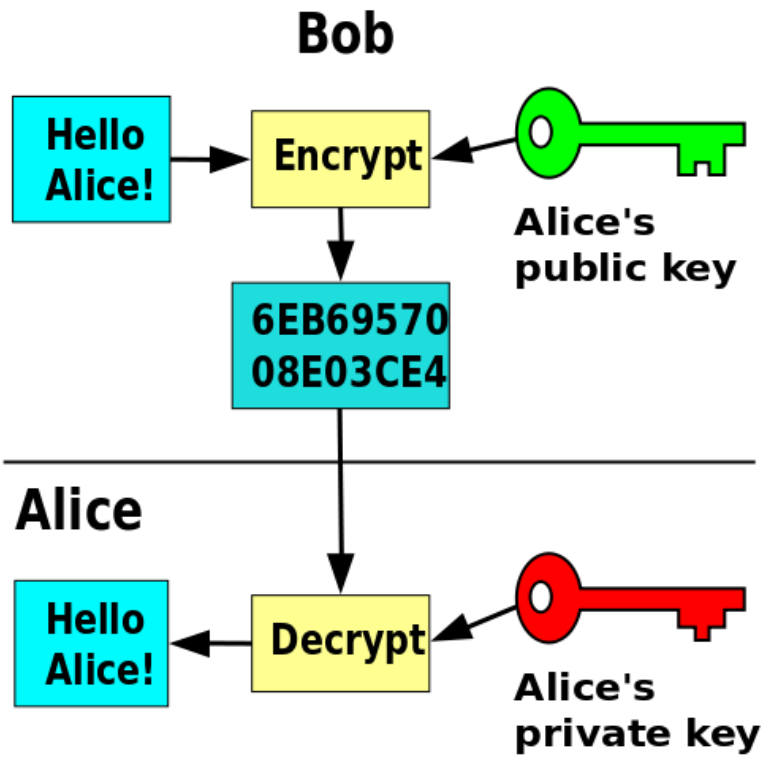


4. Descifra el resumen del mensaje mediante la clave pública de Ana.
5. Aplica al mensaje la función hash para obtener el resumen.
6. Compara el resumen recibido con el obtenido a partir de la función hash. Si son iguales, Bernardo puede estar seguro de que quien ha enviado el mensaje es Ana y que éste no ha sido modificado.

3.4. FIRMAS DE CLAVES PÚBLICAS

Este tipo de criptografía se basa en una función de un sólo sentido. Esto significa que el cálculo de la función en un sentido es relativamente sencillo, sin embargo, para deshacer ese cálculo, en el otro sentido, el coste computacional requerido es muy alto.

Existen dos claves: una pública y una privada. Como ejemplo, si nos basamos en las claves de Alice, cualquiera puede usar la clave pública de Alice, por ejemplo, Bob, para enviarle un mensaje cifrado, pero únicamente Alice puede ver ese mensaje gracias a la clave privada que sólo ella posee.



Además, un mensaje puede ser firmado (lo que se conoce como firma digital) por Alice. Este mensaje estará cifrado con su clave privada, ya que únicamente Alice la tiene. Por esta última premisa tiene validez como firma: sólo Alice la tiene. Mientras que cualquiera con acceso a la clave pública de Alice puede verificar el mensaje firmado (cifrado) por Alice.

Dicho esto, podemos concluir que el uso de la firma privada está ligado al concepto de no repudio, es decir, como sólo Bob puede tener su clave privada, si algo está firmado con su clave se tiene la certeza de que la ha firmado Alice.

Por esto último es muy **importante** tener claro que la clave que se debe compartir es la clave pública, no la clave privada.

A continuación, veremos brevemente cómo funciona el sistema criptográfico más utilizado: RSA.

RSA

RSA debe su nombre a sus presuntos inventores: Rivest, Shamir y Adleman. RSA se apoya en el problema de la factorización de números muy grandes al que, por el momento, no se ha encontrado una solución eficiente.

Para generar dos claves RSA pública y privada, es necesario elegir dos números primos *p* y *q*para después multiplicarlos. De esta forma tenemos que $N = p \cdot q$.

Lo siguiente es elegir e, primo relativo al producto $(p - 1)(q - 1)$ y *d*, tal que $ed = 1 \bmod (p - 1)(q - 1)$. En este punto tenemos *N*, que es el producto de *p* y *q*. Llamamos a *N* el módulo, el número e que hemos elegido será el *exponente de cifrado* y el número *d* es el *exponente de descifrado*. El par de claves RSA se corresponde con:

Clave pública: *(N, e)*

Clave privada (secreta): *d*

Para cifrar un mensaje *M*, dando como resultado el mensaje cifrado *C*, se calcula: $C = M^e \bmod N$

Para descifrar *C*, dando como resultado nuestro mensaje *M*, se calcula: $M = C^d \bmod N$

La forma que tiene un atacante de romper este sistema criptográfico es factorizar *N*, y usar e, que es público, para determinar *d*. Señalar que esta forma puede no ser la única.

Vamos a ver ahora un ejemplo con números muy pequeños para ver cómo funciona.

- Seleccionamos **dos primos**, $p = 11$ y $q = 3$
- Calculamos el **módulo *N***: $N = p \cdot q = 33$.
- Calculamos también $(p - 1)(q - 1) = 20$ para elegir el *exponente de cifrado* **e**, que es un primo relativo con *20*. Por ejemplo, *3*. ($e = 3$)

- Ahora calculamos el *exponente de descifrado* **d** tal que $e \cdot d = 1 \mod 20$. Despejando determinamos que es 7. ($d = 7$)

Llegados a este punto, tenemos una **clave pública** $(N, e) = (33,3)$ y una **clave privada** $(d=7)$. Vamos a cifrar un mensaje, que puede ser $M = 15$.

$C = M^e \mod N = 15^3 = 3375 = 9 \mod 33$

Por tanto, tenemos que $C = 9$, que es el mensaje que enviaríamos. Para descifrarlo, aplicaríamos la otra fórmula:

$M = C^d \mod N = 9^7 = 4782969$

Dividimos entre nuestro **módulo**, 33, y el resto determinaría el mensaje: $4782969 = 144938 \cdot 33 + 15 = 15 \mod 33$

Señalar que este ejemplo no sirve en el mundo real ya que sería computacionalmente sencillo de factorizar. En el mundo real los primos que se eligen al principio son inmensos: normalmente usando un módulo N de 2048 bits o mayor. ¡Ponte tú a factorizar eso! E incluso después, para mensajes muy pequeños, es posible llegar a determinar el mensaje. Esto se consigue mediante la llamada *búsqueda exhaustiva* (o de fuerza bruta), que consiste en cifrar todos los valores de nuestro mensaje hasta que uno coincida con el mensaje cifrado que tratamos de averiguar. Para evitar esto se añade un relleno a ese tipo de mensajes muy pequeños.

Para acabar este apartado mencionar que existen más tipos de sistemas criptográficos asimétricos. Por nombrar algunos: Diffie-Hellman, Curvas Elípticas (ECC), etc.

Veremos finalmente qué usos tiene y problemáticas surgen en la criptografía asimétrica.

Usos y problemáticas de las claves públicas.

Los sistemas criptográficos de clave pública se pueden utilizar para los mismos propósitos que las claves simétricas, que no cubrimos en este post, sin embargo, son más lentos, y destacan en los aspectos en los que las claves simétricas no se pueden utilizar. Como hemos introducido al principio, principalmente tienen dos usos: *confidencialidad* y *autenticación*.

Confidencialidad porque como hemos dicho antes, si se quiere enviar un mensaje a Bob, se cifra con la clave pública de Bob y sólo Bob puede descifrar el mensaje, aunque este mensaje cifrado se almacene o transmita por un lugar inseguro.

Este tipo de sistemas criptográficos se utilizan como *firmas digitales*, que proporcionan *integridad*, es decir, se sabría si el mensaje ha sido modificado, y no-repudio, es decir, sólo el que posea la clave privada puede ser el autor de la firma. Aquí entran en juego los robos de claves, pero en estos casos las firmas pueden ser **revocadas** por una autoridad certificadora en la que se confíe o haber caducado. Utilizamos estos conceptos a diario, por ejemplo, mediante los certificados que proporciona el FNMT, la Fábrica Nacional de Moneda y Timbre.

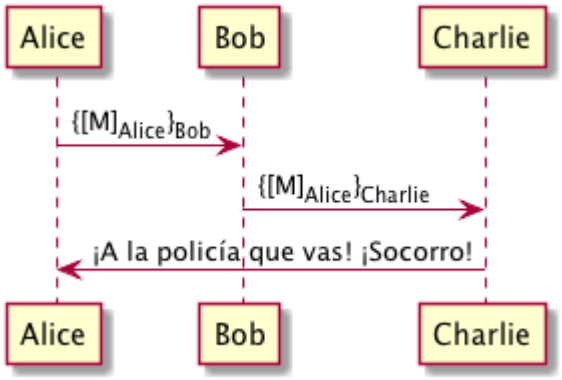
Firmar – cifrar

A estas alturas deberíamos saber la diferencia entre *firmar* y *cifrar*: firmar **sólo** se puede hacer con la clave privada, mientras que cifrar se suele referir a cifrar con la clave pública. Antes de empezar con ejemplos, vamos a aclarar la terminología a usar:

- Para **cifrar** M a Bob, es decir, cifrar un mensaje M con la clave pública de Bob, usaremos $\{M\}_{Bob}$.
- Para **firmar** M como Bob, es decir, cifrar un mensaje M con la clave privada de Bob, usaremos $[M]_{Bob}$.

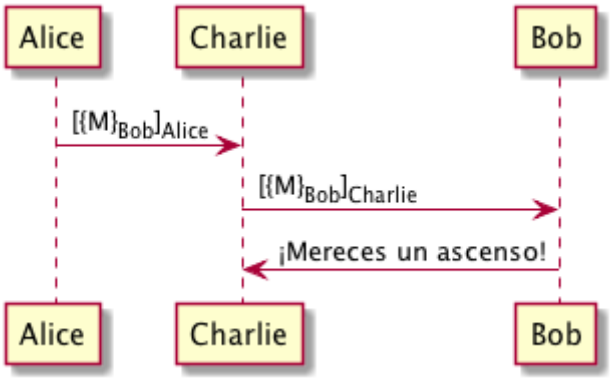
Mezclando los conceptos que acabamos de comentar, se consigue *confidencialidad* y *no repudio*. ¿Pero de qué forma hay que mezclarlos? ¿Firmar y cifrar, o cifrar y firmar? Pongamos ejemplos.

Alice envía un mensaje amenazante a Bob, y quiere asegurarse de que sabe quién se lo envía, así que firma el mensaje primero y luego lo cifra con la clave pública de Bob para que sólo él pueda abrirlo.



Bob, al percatarse del mensaje, decide jugársela a Alice aprovechando la forma en que Alice ha cifrado el mensaje. Bob descifra con su clave privada el mensaje obteniendo $[M]_{Alice}$ y lo vuelve a cifrar con la clave de Charlie para luego enviárselo. Charlie lo descifra y descubre la amenaza de Alice, por lo que entra en pánico.

Vamos a poner otro ejemplo con la situación contraria. Alice tiene una idea genial y se la quiere transmitir a su jefe Bob, así que la cifra con la clave pública de su jefe y luego la firma con su clave privada.



Charlie, su compañero de trabajo, ha interceptado el mensaje, lo ha descifrado con la clave pública de Alice y lo ha firmado con su clave privada, apropiándose de la idea de Alice y enviándosela a Bob, el jefe. El jefe felicita a Charlie y Alice llora.

¿Y ahora? ¿Ambas situaciones son malas? Hay que hacer algunas **aclaraciones**. En el primer ejemplo Charlie cree que Alice le ha enviado el mensaje. Realmente lo único que sabe es que Alice ha firmado ese mensaje en algún momento. Las claves públicas son públicas, puede haberlo enviado cualquiera.

En el segundo ejemplo, Bob asume que Charlie ha cifrado el mensaje con la clave pública de Bob. Es cierto que Charlie lo firmó, pero no implica que el mensaje lo cifrara él o incluso que supiera lo que contenía el mensaje. Se caer en este tipo de errores por no tener claro cómo funciona este tipo de criptografía, puede ser una fuente de confusión si no anda con ojo.

Aun así, se puede concluir que el uso más apropiado es el del segundo ejemplo, aunque en el mundo real se aplican otras medidas adicionales.

Autoridades certificadoras (CAs)

Retomando el tema de las revocaciones de firmas digitales, para un uso seguro de las claves públicas en el mundo real se requiere una infraestructura por detrás que asegure la gestión de los pares de claves, la autenticidad de las mismas y su posible revocación, ya sea por caducidad o por otros motivos.

Esta labor la realizan las autoridades certificadoras (en nuestro caso por ejemplo la FNMT). Un certificado emitido por este tipo de autoridades se compone de la clave pública de la persona que lo ha solicitado, con algún dato más, todo firmado con la clave privada de esa autoridad certificadora. De esta forma cualquiera puede verificar el certificado de esa persona verificando con la clave pública de la entidad certificadora.

Certificado emitido = [clave pública de la persona, algún dato más]$_{\text{Autoridad Certificadora}}$, es decir, un conjunto de datos firmados por la entidad certificadora.

Este sistema evita que alguien malintencionado use la clave pública de otra persona, ya que le sería imposible firmarlo con la clave privada de la entidad certificadora, y por tanto fallaría en la verificación que hemos comentado. Vemos un ejemplo: Como la clave de Alice es pública, Bob «emite» un certificado, que firma con su clave privada (Bob hace de entidad certificadora).

Falsificado = [clave pública de Alice, algún dato más]$_{\text{Bob}}$

Emitido por autoridad = [clave pública de Alice, algún dato más]$_{\text{Autoridad Certificadora}}$

Para verificar un certificado, comprobamos la firma con la que está cifrado. Si confiamos en la entidad certificadora que lo firma, perfecto. Si por el contrario no confiamos (como es el caso de Bob, en quien no confiamos), podemos tomar ese certificado como no válido.

Las autoridades certificadoras proporcionan una firma propia en cada clave pública emitida (los certificados) que, valida esas claves, además de probar que se ha provisto de la respectiva clave privada a la persona a la que se ha emitido el certificado.

Entre los datos que se pueden incluir en los certificados emitidos, además de la clave, se suele proporcionar una fecha de expiración o revocación, para prevenir la vigencia del certificado en el tiempo debido a la naturaleza cambiante de los datos que pueda incluir el mismo, como por ejemplo un teléfono. Sin embargo, puede darse el caso de que se necesite revocar de inmediato un certificado, ya sea por robo, una emisión por error, etc. Para esto la autoridad certificadora proporciona listas de revocación, que son consultadas como parte de la verificación de los certificados.

3.5. COMPENDIOS DE MENSAJES

Dado $m \in Z$, $m > 1$, se dice que $a, b \in Z$ son congruentes módulo m si y sólo si $m|(a-b)$. Se denota esta relación como $a \equiv b \pmod{m}$. m es el **módulo** de la congruencia.

Es importante darse cuenta de que si m divide a $a-b$, esto supone que ambos a y b tienen el mismo resto al ser divididos por el módulo m.

Ejemplos: $23 \equiv 2 \bmod 7$ (porque $23 = 3.7 + 2$), y $-6 \equiv 1 \bmod 7$ (porque $-6 = -7.1 + 1$)

La relación de congruencia como equivalencia. El conjunto de residuos.

La relación de congruencia módulo m es una relación de equivalencia para todo $m \in Z$. Es decir, cumple las propiedades reflexiva, simétrica y transitiva. Como en toda relación de equivalencia, podemos definir el conjunto cociente de las clases de equivalencia originadas por la relación de congruencia. En este caso la relación *clasifica* a cualquier entero a según el resto obtenido al dividirlo por el módulo m.

Llamaremos Z_m al conjunto cociente de Z respecto de la relación de congruencia módulo m. A la clase de equivalencia de un elemento $a \in Z$ se la denota por $[a]_m$ o simplemente **[a]**.
Para todo $a \in Z$ se tiene que $\mathbf{[a] = [r]}$ en Z_m, donde r es el resto de dividir a entre m.Por lo tanto, el conjunto Z_m es finito y tiene $m$elementos: $Z_m = \{ [0]_m, [1]_m, \dots , [m-1]_m\}$, donde la clase $[i]_m$ representa al conjunto de todos los enteros que son congruentes con $i$mod m. A este conjunto cociente se le conoce como el **conjunto de restos o residuos** (módulo m)

Ejemplo: siguiendo con el ejemplo anterior, está claro que en Z_7, el número entero 9, el 16 y el 23 pertenecen todos a la clase [2], y que el entero -6, el 1 y el 8 pertenecen a la clase [1]

Compatibilidad de la relación de congruencia con la suma y el producto

Sean $m \in N$ y $a, b, c, d \in Z$ tales que $a \equiv b \pmod{m}$ y $c \equiv d \pmod{m}$. Entonces se cumple que:

i. $a + c \equiv b + d \pmod{m}$

ii. $a \cdot c \equiv b \cdot d \pmod{m}$

Consecuentemente, el resto de la suma es congruente con la suma de restos, y el resto del producto es congruente con el producto de restos. Además podremos sumar y multiplicar clases de equivalencia (*residuos*) porque es indiferente el representante que se elija de cada clase a la hora de operar: el resultante de la operación siempre será un representante de la misma clase resultado.

Vamos ahora a definir la **aritmética módulo *m*** o aritmética en Z_m:

Aritmética en Z_m

Definición

En Z_m podemos definir dos operaciones binarias internas:

$+ , . : Z_m \times Z_m \Rightarrow Z_m$

que llamamos *suma* y *producto*, y están definidas de la siguiente manera, para cualesquiera $a, b \in Z$:

I. $[a] + [b] = [a+b]$
II. $[a] \cdot [b] = [a.b]$

Propiedades

i. Son operaciones cerradas, conmutativas y asociativas
ii. Cumplen la propiedad distributiva
iii. Tienen elemento neutro. [0] es el elemento neutro para $(Z_m , +)$ y [1] es el elemento neutro para $(Z_m , .)$
iv. En el caso de $(Z_m , +)$ existe el elemento opuesto: $-[a] = [-a]$
v. Propiedad cancelativa para $(Z_m , .)$: si $[a].[c] = [b].[c]$ en Z_m, entonces $[a] = [b]$ en $Z_{(m/mcd\ (m,c))}$
    - Un caso especial es cuando *mcd (m,c)=1* , ya que entonces se cumple la propiedad cancelativa para el producto en Z_m: si $[a][c] = [b][c]$ en $Z_m \Rightarrow [a] = [b]$ en Z_m

- Si *m* es primo, (Z_m, .) tendrá la propiedad cancelativa del producto para todo *c*

Elementos invertibles o unidades de Z_m

Se dice que [a] **es invertible en** Z_m si existe un [b] en Z_m tal que [a][b]=[1]. Ese elemento [b] será el inverso de [a] en Z_m, y se denota como $[a]^{-1}$.

Proposición:

- [a] es invertible en Z_m ,si y sólo si
- existe [b] $\in Z_m$ tal que [a][b] = [1] en Z_m ,si y sólo si
- existen *b*, *k* $\in$ Z tales que *ab* + *km* = *1* ,si y sólo si
- mcd(*a*,*m*)= 1

Si [a] es invertible puede por tanto calcularse su inverso $[a]^{-1}$ mediante el algoritmo de Euclides. Además, se puede asegurar que si existe el inverso de un elemento en módulo *m*, es único.

Por ejemplo, en Z_{12} sólo 1, 5, 7 y 11 son primos relativos al módulo 12, por lo tanto sólo [1], [5], [7] y [11] son los enteros que tienen inverso en aritmética módulo 12. Si queremos, por ejemplo, hallar el inverso del [5], tenemos que mediante Euclides:

12= 5.2 + 2

5= 2.2 + 1

2= ***1***.2

Luego recorriendo el camino inverso:

1= **5** - 2.2 = **5** - 2(**12** - **5**.2)= **5** - (**2**.12 - **5**.4)= 5.**5** -2.**12** ⇒ [5] es el inverso módulo 12 de [5].

3.6. EL ATAQUE DE CUMPLEAÑOS

De vez en cuando se hace mención en algunos artículos técnicos a que un determinado algoritmo de hash como SHA-1 o MD5 se ha "roto". En rigor, estos algoritmos no se "rompen" sino que se cuestiona su robustez en determinados supuestos, gracias al avance de las matemáticas.

En particular, en muchos de estos artículos se hace referencia a la "paradoja del cumpleaños", un clásico de la estadística que se enuncia, más o menos, de la siguiente forma: Suponiendo que la función de probabilidad de nacimientos es uniforme (es decir, que es igualmente probable nacer en un día que en otro) la probabilidad de que una persona nazca un día concreto es de 1/365, 25. Sin embargo, si nos planteamos cuantas personas tenemos que meter en una habitación para tener una probabilidad mayor del 50% de que dos de ella compartan cumpleaños, la sorprendente respuesta es 23. Lo cierto es que por las influencias del clima y de las costumbres, la distribución de probabilidad de nacimiento no es uniforme y algunos meses (singularmente los que distan 9 meses de los de verano) computan más nacimientos que otros. De esta forma la probabilidad de que 2 personas en una sala llena de gente cumplan años el mismo día es bastante alta.

Trasladado a la probabilidad de que dos documentos diferentes en una colección de ellos computen el mismo hash, el valor, con la mayor parte de los algoritmos, es significativamente mayor que 1/(2^numbits).

Si los documentos con los que trabajamos son de tipo estructurado (es decir, solo se pueden modificar campos concretos para que no se afecte su estructura) esta debilidad de los algoritmos no es significativa. Sin embargo, si trabajamos sobre documentos con áreas modificables (por ejemplo, ficheros gráficos o discos duros), la debilidad comienza a ser significativa, ya que un atacante podría modificar de forma ventajosa para el la parte del contenido modificable generando resultados que podrían resultar ser colisiones respecto al valor de hash del documento original.

3.7. FUNCIONES DE DISPERSION

SHA y SHA-1

El SHA (*Secure Hash Algorithm*) es un algoritmo de resumen seguro desarrollado por el NIST. El SHA-1 es una versión corregida del algoritmo publicada en 1994. El algoritmo es un estándar ANSI.

El algoritmo toma un mensaje de menos de 2^{64} bits y genera un resumen de 160 bits. Es más lento que el MD5, pero la mayor longitud de clave lo hace más resistente a ataques de colisión por fuerza bruta y de inversión.

MD2, MD4 y MD5

Los tres son algoritmos de resumen de mensajes (el MD viene de *Message Digest*) desarrollados por Rivest.

Los tres toman un mensaje de longitud arbitraria y generan un resumen de 128 bits. El MD2 está optimizado para máquinas de 8 bits, mientras que el MD4 y MD5 son para arquitecturas de 32 bits. El código para los tres algoritmos se puede encontrar en los RFCs 1319, 1320 y 1321.

El MD2 funciona rellenando el mensaje para que tenga una longitud en bytes múltiplo de 16. Sobre ese mensaje se calcula un checksum de 16 bytes que se añade al mensaje y la función de dispersión se aplica al mensaje resultante. El único problema que se le conoce es que si se omite el checksum se pueden obtener colisiones.

El MD4 fue desarrollado en 1990 por Rivest. El mensaje se rellena para que su longitud en bits más 448 sea divisible por 512. Una representación de la longitud del mensaje de 64 bits se concatena entonces con el mensaje. El mensaje se procesa iterativamente en bloques de 512 bits y cada bloque es procesado en tres rotaciones distintas. El algoritmo ha sido criptoanalizado y se han encontrado debilidades, de hecho es posible encontrar colisiones en menos de un minuto en máquinas modernas, por lo que el algoritmo se considera a todos los efectos roto.

El MD5 fue desarrollado en 1991 por Rivest. Es básicamente el MD4 con mejoras en la seguridad, aunque es más lento que este. El tamaño del resumen y la necesidad del relleno son

iguales que en el MD4. Consta de cuatro rotaciones que tienen un diseño ligeramente diferente a las del MD4. El algoritmo ha sido criptoanalizado con técnicas similares a las del MD4 y se han encontrado pseudo-colisiones en la función de compresión, pero no en el algoritmo completo. Adicionalmente, se ha estimado que es posible construir una máquina capaz de atacar el algoritmo por fuerza bruta y encontrar una colisión en 24 días, aunque el coste de la máquina era de 10 millones de dolares en 1994.

3.8. FIRMAS DIGITALES

DSA y DSS

El DSA (*Digital Signature Algorithm* o *Algoritmo Estándar de Firmado*) es el algoritmo de firmado digital incluido en el DSS (*Digital Signature Standard* o *Estándar de Firmas Digitales*) del NIST Norteamericano. Se publicó en 1994.

El DSA está basado en el problema de los logaritmos discretos y sólo puede emplearse para las firmas digitales (a diferencia del RSA, que también puede emplearse para encriptar). La elección de este algoritmo como estándar de firmado generó multitud de críticas: se pierde flexibilidad respecto al RSA (que además, ya era un estándar *de hecho*), la verificación de firmas es lenta, el proceso de elección fue poco claro y la versión original empleaba claves que lo hacían poco seguro.

El algoritmo es más rápido para generar la firma que para validarla, al revés de lo que sucede con el RSA. Emplea claves de 1024 bits (originalmente eran 512 bits, pero se aumento por falta de seguridad). No se conocen ataques eficientes contra este algoritmo, sólo existen problemas con un conjunto de números primos, pero son fácilmente evitables si se siguen los sistemas adecuados de generación de claves.

3.9. CERTIFICADOS DIGITALES

Los certificados digitales son el equivalente digital del DNI, en lo que a la autentificación de individuos se refiere, ya que permiten que un individuo demuestre que es quien dice ser, es decir, que está en posesión de la clave secreta asociada a su certificado.

Para los usuarios proporcionan un mecanismo para verificar la autenticidad de programas y documentos obtenidos a través de la red, el envío de correo encriptado y/o firmado digitalmente, el control de acceso a recursos, etc.

En este apartado explicaremos qué son los certificados digitales, cuales son los formatos estándar, como podemos controlar sus periodos de validez o anularlos si se ven comprometidos, quien los genera y las infraestructuras necesarias para soportarlos.

3.10. LISTAS DE ANULACIÓN DE CERTIFICADOS

Los certificados tienen un periodo de validez que va de unos meses a unos pocos años. Durante el tiempo que el certificado es válido la entidad certificadora que lo generó mantiene información sobre el estado de ese certificado.

La información más importante que guarda es el *estado de anulación*, que indica que el periodo de validez del certificado ha terminado antes de tiempo y el sistema que lo emplee no debe confiar en él. Las razones de anulación de un certificado son varias: la clave privada del sujeto se ha visto comprometida, la clave privada de la CA se ha visto comprometida o se ha producido un cambio en la afiliación del sujeto (por ejemplo cuando un empleado abandona una empresa).

Las *listas de anulación de certificados* (*Certification Revocation Lists* o CRL) son un mecanismo mediante el cual la CA publica y distribuye información a cerca de los certificados anulados a las aplicaciones que los emplean. Una CRL es una estructura de datos firmada por la CA que contiene su fecha y hora de publicación, el nombre de la entidad certificadora y los números de serie de los certificados anulados que aun no han expirado. Cuando una aplicación trabaja con certificados debe obtener la última CRL de la entidad que firma el certificado que está empleando y comprobar que su número de serie no está incluido en él.

Existen varios métodos para la actualización de CRLs:

1. **Muestreo de CRLs**. Las aplicaciones acceden a la CA o a almacenes de archivos y copian el último CRL a intervalos regulares. La pega de este esquema es que durante el periodo entre actualizaciones del CRL podemos aceptar un certificado ya anulado, por lo que el periodo debe ser corto.

2. **Anuncio de CRLs**. La entidad certificadora anuncia que ha habido un cambio en el CRL a las aplicaciones. El problema de este enfoque es el anuncio puede ser muy costoso y no sabemos que aplicaciones deben ser informadas.

3. **Verificación en línea**. Una aplicación hace una consulta en línea a la CA para determinar el estado de revocación de un certificado. Es el mejor método para las aplicaciones, pero es muy costoso para la CA

3.11. INFRAESTRUCTURA DE CLAVE

La difusión de las técnicas de clave pública requiere una *infraestructura* que defina un conjunto de estándares, autoridades de certificación, estructuras entre múltiples CAs, métodos para descubrir y validar rutas de certificación, protocolos operacionales, protocolos de gestión, herramientas que pueden operar entre sí y un marco legislativo.

Los protocolos operacionales se dirigen al problema del envío de certificados y CRLs a los sistemas que emplean certificados. Los protocolos de gestión tratan de los requisitos para la interacción de dos componentes de la infraestructura: registro, inicialización, certificación, anulación y recuperación de claves.

Una estructura entre múltiples CAs proporciona una o más rutas de certificación entre un subscriptor y una aplicación. Una *ruta de certificación* (o cadena de certificación) es una secuencia de uno o más puntos conectados entre el subscriptor y una CA raíz. Una *CA raíz* es una autoridad en la que confía la aplicación, ya que tiene almacenada de forma segura su clave pública.

Un sistema que emplea certificados necesita obtener una ruta de certificación entre un subscriptor y un CA raíz antes de evaluar el nivel de confianza en el certificado del subscriptor. El problema de determinar una ruta de certificación entre dos subscriptores arbitrarios en una estructura de interconexiones entre diferentes CAs se denomina *descubrimiento de rutas de certificación*. El problema de verificar la asociación entre el nombre del subscriptor y su clave pública en una ruta de certificación se denomina *validación de la ruta de certificación*

3.12. PROTOCOLO SSL

En este apartadodiscutiremos mecanismos para establecer canales seguros para aplicaciones de red a nivel de la capa de transporte. Trataremos los protocolos SSL y TSL, que son los más utilizados en la actualidad para proporcionar versiones seguras de protocolos de red como el http (https).

3.13. APLICACIONES E IMPLEMENTACIONES

El protocolo SSL/TLS tiene multitud de aplicaciones en uso actualmente. La mayoría de ellas son versiones seguras de programas que emplean protocolos que no lo son. Hay versiones seguras de servidores y clientes de protocolos como el http, nntp, ldap, imap, pop3, etc.

3.14. MODELOS DE SEGURIDAD

Desde su creación el entorno Java ha tenido presentes los problemas de seguridad y ha definido un modelo para controlar y limitar el acceso a los recursos desde los programas y aplicaciones. El modelo de seguridad ha ido evolucionando con las distintas versiones del *Entorno de Desarrollo Java* (de aquí en adelante denominado JDK, por sus siglas en inglés), pasando de un modelo muy sencillo y restrictivo, el del JDK 1.0, a uno más complejo y flexible desde la aparición del JDK 1.2.

3.15. DOMINIOS PROTEGIDOS

El concepto de *dominio protegido* es fundamental para la seguridad de los sistemas. El alcance de un dominio está definido por el conjunto de objetos que estan directamente accesibles para un *principal*, donde *principal* es una entidad en el sistema informático a la que se le han asignado *permisos*. El *cajón de arena* del JDK 1.0 es un ejemplo de dominio de protección con límites fijos.

El concepto de *dominio protegido* proporciona un mecanismo adecuado para agrupar y aislar unidades de protección. Por ejemplo, se pueden separar dos dominios de forma que la interacción entre ambos únicamente sea posible a través de código del sistema o de un protocolo explícito para la comunicación entre ambos.

Los dominos protegidos se dividen generalmente en dos categorias:

1. **Dominos del sistema**, que controlan el acceso a los recursos del sistema (sistema de archivos, acceso a la red, E/S).

2. **Dominios de aplicación**, que controlan el acceso a los recursos de una aplicación.

UNIDAD IV

PROTOCOLOS DE AUTENTICACIÓN

4.1. AUTENTICACIÓN BASADA EN UNA CLAVE SECRETA COMPARTIDA

La autenticación es un aspecto fundamental de la seguridad de un sistema. Confirmar la identidad de cualquier usuario que intenta iniciar la sesión en un dominio o tener acceso a los recursos de la red.

En la familia de servidores Windows Server 2003, la autenticación permite el inicio de sesión único en todos los recursos de red. Con un inicio de sesión

único, un usuario puede iniciar la sesión en el dominio una vez, mediante una contraseña única o una tarjeta inteligente, y autenticarse en cualquier equipo de l dominio.

A lo largo de la historia el ser humano ha desarrollado unos sistemas de seguridad que le permiten comprobar en una comunicación la identidad del interlocutor, a asegurarse de que sólo obtendrá la información el destinatario seleccionado , que además ésta no podrá ser modificada e incluso que ninguna de las dos partes podrá negar el hecho ni cuándo se produjo (ej. fechado de d ocumentos).

En la mayor parte de los casos el sistema de seguridad se basa en la identificación física de la persona, información que se contrasta con el documento de identidad. Actualmente cada vez mayor número de actividades se está trasladando al mundo electrónico a través de Internet.

Protocolos de autenticación.

Un protocolo de autenticación, es un tipo criptográfico que tiene el propósito de autentificar entidades que desean comunicarse de forma segura. Los protocolos de autenticación se negocian inmediatamente después de determinar la calidad del vínculo y antes de negociar el nivel de red. Algunos protocolos de autenticación son:

PAP (Protocolo de autenticación de contraseña):

Es un protocolo de autenticación simple en el que el nombre de usuario y la contraseña se envían al servidor de acceso remoto como texto simple (sin cifrar). No se recomienda utilizar PAP, ya que las contraseñas pueden leerse fácilmente en los paquetes del Protocolo punto a punto intercambiados

durante el proceso de autenticación. PAP suele utilizarse únicamente al conectar a servidores de acceso remoto antiguos basados en UNIX que no admiten métodos de autenticación más seguros.

CHAP (Protocolo de autenticación por desafío mutuo):

Es un método de autenticación muy utilizado en el que se envía una representación de la contraseña del usuario, no la propia contraseña. Con CHAP, el servidor de acceso remoto envía un desafío al cliente de acceso remoto. El cliente de acceso remoto utiliza un algoritmo hash (también denominado función hash) para calcular un resultado hash de MessageDigest-5 (MD5) basado en el desafío y un resultado hash calculado con la contraseña del usuario. El cliente de acceso remoto envía el resultado hashMD5 al servidor de acceso remoto.

El servidor de acceso remoto, que también tiene acceso al resultado hash de la contraseña del usuario, realiza el mismo cálculo con el algoritmo hash y compara el resultado con el que envió el cliente. Si los resultados coinciden, las credenciales del cliente de acceso remoto se consideran auténticas. El algoritmo hash proporciona cifrado unidireccional, lo que significa que es sencillo calcular el resultado hash para un bloque de datos, pero resulta matemáticamente imposible determinar el bloque de datos original a partir del resultado hash.

SPAP (Protocolo de autenticación de contraseña de shiva):

Es un protocolo de autenticación simple de contraseña cifrada compatible conservadores de acceso remoto de Shiva. Con SPAP, el cliente de acceso remoto envía una contraseña cifrada al servidor de acceso remoto. SPAP utiliza un algoritmo de cifrado bidireccional. El servidor de acceso remoto

descifra la contraseña y utiliza el formato sin cifrar para autenticar al cliente de acceso remoto.

MS-CHAP y MS-CHAP v2

Protocolo de autenticación por desafío mutuo de Micr osoft: Microsoft creó MS-CHAP para autenticar estaciones de trabajo Windows remotas, integrando la funcionalidad a la que los usuarios de redes LAN están habituados con los algoritmos de has utilizados en las redes Windows. Utiliza un mecanismo de desafío y respuesta para autenticar conexiones sin enviar contraseñas.

El autenticador (el servidor de acceso remoto o el servidor IAS) envía al cliente de acceso remoto un desafío formado por un identificador de sesión y una cadena de desafío arbitraria.

El cliente de acceso remoto envía una respuesta que contiene el nombre de usuario y un cifrado no reversible de la cadena de desafío, el identificador de sesión y la contraseña.

El autenticador comprueba la respuesta y, si es válida, se autentican las credenciales del usuario. La familia Windows Server 2003 admite MS-CHAP v2, que proporciona autenticación mutua, la generación de claves de cifrado de datos iniciales más seguras para Cifrado punto a punto de Microsoft (MPPE) y distintas claves de cifrado para los datos enviados y los datos recibidos. Para reducir al mínimo el riesgo de que una contraseña se vea comprometida durante su cambio, no se admiten métodos más antiguos que el cambio de contraseña de MS-CHAP.

EAP (Protocolo de autenticación extensible):

Es una extensión del Protocolo punto a punto (PPP) que admite métodos de a utenticación arbitrarios que utilizan intercambios de credenciales e información de longitudes arbitrarias. EAP se ha desarrollado como respuesta a la creciente demanda de métodos de autenticación que utilizan dispositivos de seguridad, como las tarjetas inteligentes, tarjetas de identificación y calculadoras de cifrado. Mediante EAP, se pueden admitir esquemas de autenticación adicionales, conocidos como tipos EAP. Entre estos esquemas se incluyen las tarjetas de identificación, contraseñas de un solo uso, autenticación por clave pública mediante tarjetas inteligentes y certificados. EAP, junto con los tipos de EAP seguros, es un componente tecnológico crítico para las conexiones de red privada virtual (VPN) seguras.

La familia Windows Server 2003 admite dos tipos de EAP: y EAP-MD5 CHAP (equivalente al protocolo de autenticación CHAP) y EAP-TLS (utilizado para autenticación basada en certificados de usuario). EAP-TLS es un método de autenticación mutua, lo que significa que tanto el cliente como el servidor deben demostrar sus identidades uno a otro. Durante el proceso de autenticación, el cliente de acceso remoto envía su certificado de usuario y el servidor de acceso remoto envía su certificado de equipo. Si el certificado no se envía o no es válido, se termina la conexión.

Kerberos:

Es un protocolo de autenticación de redes de ordenador que permite a dos computadores en una red insegura demostrar su identidad mutuamente de manera segura. Sus diseñadores se concentraron primeramente en un modelo de cliente-servidor, y brinda autenticación mutua: tanto cliente como servidor verifican la identidad uno del otro. Los mensajes de autenticación están protegidos para evitar eavesdropping y ataques de Replay. Kerberos se basa en criptografía de clave simétrica y requiere un tercero de confianza. Además,

existen extensiones del protocolo para poder utilizar criptografía de clave asimétrica.

Claves secretas compartidas

Este protocolo se basa en un principio encontrado en muchos protocolos de a utenticación. Una parte envía un nuero aleatorio a la otra, quien a continuación lo transforma en una forma especial y después regresa al resultado. Tales protocolos se conocen como de desafío-respuesta. Es posible especificar una clave secreta compartida previamente. Su uso es sencillo y no r equiere que el cliente ejecute el protocolo Kerberos V5 ni que tenga un certificado de claves públicas. Ambas partes deben configurar IP Se c manualmente para utilizar esta clave compartida previamente.

Importante:

El uso de autenticación por claves compartidas previamente no se recomienda porque es un método de autenticación relativamente débil. La autenticación por claves compartidas previamente crea una clave maestra que es menos segura (y que podría ofrecer una forma de cifrado más débil) que los certificados o el protocolo Kerberos V5. Asimismo, las claves compartidas previamente se almacenan en texto no cifrado. La autenticación por claves compartidas previamente se utiliza por motivos de interoperabilidad y por compatibilidad con los estándares de IPSec. Se recomienda que sólo utilice claves previamente compartidas en pruebas y que, en su lugar, emplee certificados o Kerberos V5 en un entorno de producción.

4.2. ESTABLECIMIENTO DE UNA CLAVE COMPARTIDA: EL INTERCAMBIO DE CLAVES DE DIFFIE-HELLMAN

En cuestiones de seguridad informática, el **algoritmo Diffie-Hellman** fue uno de los métodos pioneros de intercambio de claves en un esquema de alta seguridad. El presente artículo trata de la implementación de un ejemplo sencillo de este algoritmo sobre la programación con Socket, por lo que se debe de hacer lo siguiente:

1. Definir dos números primos muy grandes los cuales tendrán las funciones de generador (g) y primo de referencia (p). Por lo general, siempre se designa, en el esquema de intercambio a un equipo/entidad, con funciones de servidor, que defina estos números y los de a conocer a sus pares del mismo esquema.

Esta primera notificación indica el inicio del ciclo de seguridad de intercambio de claves.

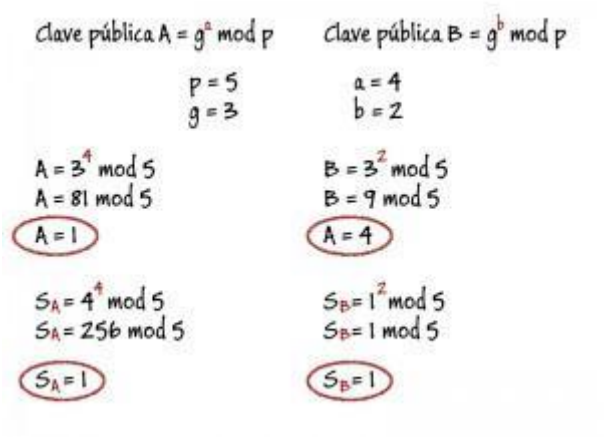
2. Cada equipo/entidad debe de definir una clave privada, la cual puede de ser un número no primo; sin embargo, debe de ser un número muy alejado del cero en la recta numérica, es decir, un número muy grande.
3. Cada equipo/entidad debe de generar una clave pública, a partir del número generador (g), del número primo de referencia (p) y de la clave privada, según la fórmula siguiente:

$$\text{Clave pública A} = g^a \bmod p \qquad \text{Clave pública B} = g^b \bmod p$$

4. Cada equipo/entidad debe de intercambiar la clave pública generada en el inciso anterior. Al momento de la recepción de la clave pública de los pares, se debe de generar una clave de verificación, como lo muestra el siguiente procedimiento:

$$\text{Verificación A} = B^a \bmod p$$

5. Las claves de verificación deben de ser iguales entre sí, tal a como se muestra en la ilustración siguiente:



Bien, ahora que ya conocemos el funcionamiento del algoritmo Diffie-Hellman, veamos cómo implementarlo en una aplicación que pueda centrar utilidad en un entorno operativo aplicando Sockets, pero antes, veamos algo respecto a este tema.

Programación con socket:

No existe una definición estándar para este término que englobe todos los contextos, pero, en programación, la definición más común para *socket* es la siguiente: Un socket es una interface de programación de aplicaciones en un entorno de red.

Existen dos tipos programación de sockets: sincrónica y asincrónica. La programación sincrónica se enfoca en tiempo, y la asincrónica en la demanda del servicio.

Este tipo de programación se utiliza cuando se desea que dos aplicaciones (programas) se comuniquen entre sí, ya sea de manera definida cada cierto tiempo (sincrónica) o cuando un equipo/entidad lo demande (asincrónica). Estas aplicaciones, normalmente, se ubican en equipos diferentes con acceso mutuo de red, sin embargo, también pueden configurarse en el mismo equipo si el requerimiento así lo permite. La programación de socket **asincrónica** es más común y menos costosa en recursos de servidor, ya que no requiere de una conexión permanente asignada, sino que opera bajo un esquema en el

cual se abre una conexión por demanda, y se cierra al finalizar el intercambio. La programación de socket **sincrónica** sí requiere este alto costo, una conexión asignada por el servidor de manera permanente.

En este tipo de aplicaciones, se aplica un esquema variante de cliente/servidor, dado que en algún tiempo/demanda, una entidad presta los servicios y notifica a las demás (entidades clientes); pero esta condición puede variar cuando le toque ser notificada por otra entidad.

En el momento de la notificación, una aplicación cliente requiere saber que ha recibido un paquete del servidor, por lo que tiene dos opciones: programa un evento que verifique el contenido en el socket cada cierto tiempo, o espera una notificación o *trigger* del sistema por cada paquete entrante en el socket. En el caso de Windows, que es un sistema operativo que maneja los eventos, esto le da al sistema de notificación la ventaja, por tanto, lo único que se requiere es tener la herramienta necesaria para capturar ese evento, interpretarlo y obtener el mensaje.

4.3. AUTENTICACIÓN QUE UTILIZA UN CENTRO DE DISTRIBUCIÓN DE CLAVES

Kerberos surge porque algunas aplicaciones cliente/servidor asumen que el cliente proveerá su identificación correctamente, y otras confían en que el cliente restringirá sus actividades a aquellas que están autorizadas sin ningún otro refuerzo del servidor.

Algunos sitios utilizan firewalls para solucionar los problemas de seguridad en redes. Pero los firewalls asumen que las personas que desean hacer daño están del lado de afuera, no siendo esto, una verdad en todos los casos.

Kerberos es un personaje de la mitología griega (un perro tricéfalo) que custodiaba las puertas del infierno, es decir, representa seguridad. Se podría decir que, como servicio de autenticación, ahora cuida las puertas de la red, impidiendo que entren personas indeseadas.

Kerberos es un servicio de autenticación desarrollado en MIT (Massachusetts Institute of Technology) y diseñado por Miller y Neuman en el contexto del Proyecto Athena – proyecto sobre autentificación de usuarios - en 1987.

Está basado en el protocolo de distribución de claves presentado por Needham y Schroeder en 1978, y lo podemos encontrar en la RFC 1510. Microsoft ha decidido implementar su propia versión de Kerberos como protocolo de autentificación por omisión para su nuevo sistema operativo Windows.

En una red con usuarios que solicitan servicios desde muchas terminales, hay tres enfoques básicos que se pueden utilizar para dar control de acceso:

- No hacer nada. Confiar en que la máquina en la que el usuario está "logueado" evite accesos no autorizados.
- Requerir que el host pruebe su identidad, pero confiar en su palabra sobre quién es el usuario.
- Requerir que el usuario pruebe su identidad para cada servicio solicitado.

En los ambientes en los que cada usuario debe probar su identidad para obtener todos y cada uno del servicio deberemos utilizar el último de los enfoques, y aquí es donde entra Kerberos.

En una red de usuarios en la que se requieren servicios de distintas computadoras, si la red tiene miles de usuarios y decenas de servidores, no es deseable que cada servidor guarde las contraseñas de todos los usuarios. En ese caso, habría tantos puntos de ataque como servidores. Además, por ejemplo, si un usuario quisiera cambiar su contraseña, debería contactar a todos los servidores y notificarles del cambio. Es para evitar estos problemas que surge la idea de tener un servicio de autenticación.

Kerberos es el resultado de satisfacer estos requerimientos. La seguridad de Kerberos descansa en la seguridad de varios servidores de autenticación, pero no en el sistema que se "loguea" o en los servidores finales que se utilicen.

¿Qué es y qué hace Kerberos?

Kerberos es un protocolo que ofrece un servicio de autentificación en arquitecturas cliente / servidor. Cada usuario tendrá una clave y cada servidor tendrá una clave, y Kerberos tiene una base de datos que las contendrá a todas.

En el caso de ser de un usuario, su clave será derivada de su contraseña y estará encriptada, mientras que, en el caso del servidor, la clave se generará aleatoriamente. Los servicios de red que requieren autenticación y los usuarios que requieran estos servicios, se deben registrar con Kerberos. Las claves privadas se negocian cuando se registran.

Como Kerberos sabe todas las claves privadas, puede crear mensajes que convenzan a un servidor de que un usuario es realmente quien dice ser y viceversa. La otra función de Kerberos es generar las llamadas claves de sesión, que serán compartidas entre un cliente y un servidor, y nadie más. La clave de sesión podrá ser usada para encriptar mensajes que serán intercambiados entre ambas partes. El almacenamiento de la base de datos y la generación de claves, se lleva a cabo en un servidor que se denomina Servidor de Autenticación (AS por las siglas en inglés de Authentication Server).

Por Ejemplo, en Windows todas las claves de sesión se generan en el KDC (Kerberos Key Distribution Center, Centro de distribución de claves Kerberos). Además, incluye un proveedor de autentificación Kerberos cliente, además del soporte Kerberos para otros tipos de cliente, como Win9x. Si desea que su cliente Win9x utilice Kerberos para llevar a cabo la autentificación, deberá instalar el cliente para los servicios de Directorio. Si, en cambio, necesita disponer del soporte Kerberos para Windows NT 4.0 Workstation, no tendrá más remedio que migrar a Windows Professional.

Kerberos provee tres niveles distintos de protección. El programador de la aplicación determinará cual es apropiado, de acuerdo a los requerimientos de la aplicación.

- Autenticación: Prueba que el usuario es quien dice ser. Puede ser que la autenticidad se establezca al inicio de la conexión de red y luego se asuma que los siguientes mensajes de una dirección de red determinada se originan desde la parte autenticada.
- Integridad de datos: Asegura que los datos no se modifican en tránsito. Se requiere autenticación de cada mensaje, sin importar el contenido del mismo. Éstos se denominan mensajes seguros.
- Privacidad de datos: Asegura que los datos no son leídos en tránsito. En este caso no sólo se autentica cada mensaje, sino que también se encripta. Éstos son mensajes privados.

Hay dos tipos de credenciales que se utilizan en el modelo de autenticación de Kerberos: Tickets y Autenticadores. Aunque ambos se basan en encriptado de clave privada, se encriptan con claves diferentes. El ticket se usa para pasarle al servidor final la identidad de la persona para la que fue emitido. El autenticador es una prueba de que el ticket fue creado para el usuario y no fue robado; contiene información que al ser comparada contra la que está en el ticket prueba que el usuario que lo presenta es el mismo al que le fue emitido.

El núcleo del protocolo Kerberos es un sistema de tickets único que facilita una autentificación. Un ticket Kerberos proporciona un medio de transporte seguro, a través de la red, a una clave de sesión Kerberos, clave que constituye la entidad de autentificación básica.

La autentificación Kerberos se basa en el cifrado de claves simétricas. Supongamos, por ejemplo, que Alicia y Roberto comparten una clave de sesión y que desean usarla para autentificarse entre ellos. Cuando Alicia desea autentificarse ante Roberto utiliza su clave de sesión para cifrar su nombre y la fecha y hora actuales y, a continuación, remite el resultado a Roberto (en terminología Kerberos, el paquete cifrado resultante se conoce con el nombre de autentificador). Roberto utilizará la clave de sesión, que sólo Alicia y él conocen, para descifrar el paquete. Una vez descifrado, si el resultado es el nombre de Alicia y una fecha y hora aceptables, Roberto sabe que sólo Alicia pudo haberlo enviado. Kerberos utiliza los tickets para asegurarse de que el intercambio de claves de sesión se realiza de forma segura.

Funcionamiento de Kerberos

El funcionamiento de Kerberos es como ya se ha dicho antes mediante tickets o billetes y claves de sesión. Todas las claves de sesión se generan en el KDC (Kerberos Key Distribution Center, Centro de distribución de claves Kerberos). El KDC también crea los billetes asociados y los envía al cliente y al servidor de recursos a través del cliente. El KDC envía todos los billetes a través del cliente, de modo que éste puede guardarlos en caché y reutilizarlos.

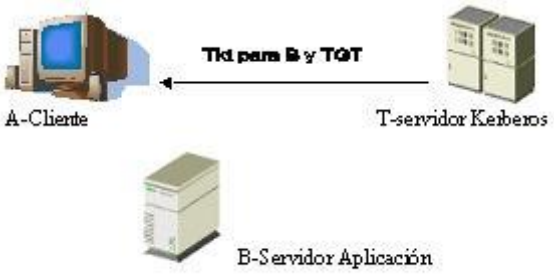
La notación que se va a emplear el E para la función de cifrado simétrico, L para el tiempo de vida del ticket Tkt, N_A es un valor irrepetible y T_A será el sello de tiempo de A. La secuencia de funcionamiento es la siguiente:

a) Solicitud de credenciales. A genera un valor irrepetible N_A y se lo envía al servidor Kerberos junto con los identificadores de A y B

A -> T = (A,B,Na)

b) Generación de tickets y clave de sesión. T genera una clave de sesión k y define un tiempo de vida L para el ticket de acceso. A partir de esta información construye el ticket de acceso válido para A y que habrá que presentar a B, Tkt = $E_{KAT}(k, N_A, L, B)$

TA: Tkt = $E_{KBT}(k, A, L)$ + TGT = $E_{KAT}(k, N_A, L, B)$

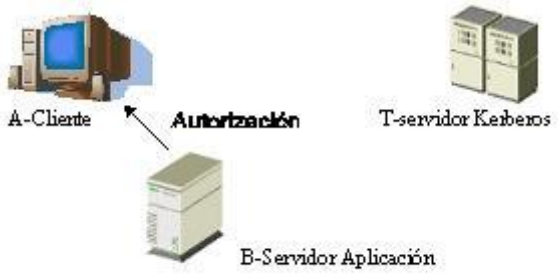


c) Verificación del ticket y generación del autenticador. A descifra el TGT verificando que N_A es el mismo que envió y guarda el valor de L. Además, genera el autenticador para B para lo cual cifra con la clave de sesión, su propio identificador y el sello de tiempo T_A que establece en que instante se genera éste. A reenvía a B el ticket Tkt junto con el autenticador.

AB: Tkt = $E_{KBT}(k, A, L)$ + Tok = $E_K(A, T_A)$

d) Concesión de la autorización. B descifra el ticket Tkt empleando la clave compartida KBT. De esta forma obtiene la clave de sesión k que le permite descifrar el autenticador Tok. De este modo, B verifica que los identificadores de A en el ticket Tkt y el autenticador Tok son iguales, el sello de Tiempo T_A de autenticador Tok es válido y la hora local de B está dentro de los márgenes del tiempo de vida L del ticket Tkt. En el caso de que todo sea correcto B autoriza el acceso a A.

BA: AUT



Servicios ofrecidos

Autenticación mutua

Finalmente, el cliente puede querer que el servidor pruebe su identidad también. Entonces, el servidor le suma uno al sello de tiempo que el cliente envió en el autenticador, encripta el resultado con la clave de sesión y envía el resultado de vuelta al cliente.

Esto demuestra que el servidor pudo leer el sello de tiempo del autenticador y para ello debió obtener la clave de sesión que se encontraba en el ticket encriptado con su clave privada. Entonces el cliente se cerciora de que el servidor es auténtico. Además, el cliente y el servidor comparten un secreto que nadie más sabe, y pueden asumir que un mensaje relativamente reciente encriptado con esa clave fue originado por el otro.

Para el usuario la presencia de Kerberos es casi imperceptible. Los tickets se obtienen como parte del proceso de "login" y son destruidos automáticamente cuando el usuario termina la sesión. El usuario notará la presencia de Kerberos si su sesión dura más de 8 horas. En ese caso, si intenta acceder a algún servicio que utilice autenticación por medio de Kerberos el intento fallará porque el ticket habrá expirado.

Confianza transitiva

La confianza transitiva se puede explicar mediante el ejemplo siguiente: sean dos empresas A y B confían en C, entonces A confía implícitamente en B. La confianza transitiva permite reducir el número de relaciones de confianza necesarias para llevar a cabo la autentificación.

La confianza transitiva no es más que un concepto lógico: no existe ningún secreto entre los controladores de dominios que comparte una confianza transitiva. Microsoft afirma que este proceso de referencias no afecta al tráfico en la red, ya que el consumo

generado por él es comparable al generado por la autentificación típica de un NT 4.0. A pesar de ello, la evidencia indica que Kerberos no resulta adecuado para entornos distribuidos muy grandes, como Internet. Kerberos es un protocolo de autentificación orientado a Intranets.

Autentificación delegada

Si se delega la autentificación, el usuario A puede otorgar derechos a una máquina intermediaria B para que ésta lleve a cabo la autentificación ante un servidor de aplicaciones C, como si de la máquina A se tratara. Esta configuración significa que las decisiones que el servidor de aplicaciones C tome, relativas al control de acceso, se basarán en la identidad del usuario y no en la información de las cuentas de usuario mantenidas por la máquina B. La delegación también es conocida por el nombre de redireccionamiento de la autentificación. En Kerberos, este proceso de delegación consiste en que un usuario A envía un ticket a una máquina intermediaria B y ésta utiliza el ticket del usuario A para autentificarse ante el servidor de aplicaciones C.

Administración de Kerberos

A la hora de administrar Kerberos, habrá que tener en cuenta dos tipos de administración:

a) Administración General.

Para el administrador del sistema, la presencia de Kerberos implica cierto mantenimiento. El administrador debe iniciar la base de datos de Kerberos al instalarlo. Además, en caso de tener la base de datos replicada (con slaves), también será necesario administrarlas para que se mantengan consistentes. Esto implica tener que sincronizar las bases de datos replicadas con la base de datos principal cada cierto tiempo.

Se debe tener especial cuidado con la administración de las máquinas que tienen la base de datos Kerberos, pues el sistema podría volverse inseguro si alguien obtiene el control de alguna de estas máquinas. El administrador también debe asegurarse de que la fecha y hora de todas máquinas del sistema estén medianamente sincronizadas. Cuando una aplicación Kerberos es agregada al sistema, el administrador debe realizar varias operaciones para dejarla funcional. El servidor kerberizado debe ser registrado en la base de datos, y se le debe asignar una clave privada. Luego algunos datos deben ser transferidos al nuevo servidor.

Además se aconseja tener la base de datos principal resguardada, en caso de que el disco en el que reside falle.

Administración entre sistemas Kerberos.

Cada sistema Kerberos contiene como parte de su nombre el *realm* (dominio) asociado. Esto permite que distintos sistemas Kerberos se comuniquen entre sí para brindarse servicios. Cuando un usuario se autentica con un sistema Kerberos, lo hace a un *realm* específico. Las aplicaciones kerberizadas generalmente reconocen credenciales de un *realm* particular; pero los usuarios pueden obtener credenciales de otros *realms* distintos del *realm* local. Para ello deben estar autenticados con el sistema Kerberos local, y los dos sistemas Kerberos (el local y el remoto) deben ponerse de acuerdo para brindar las credenciales necesarias en el sistema Kerberos remoto.

Los administradores de cada par de sistemas Kerberos que desean cooperar deben ponerse de acuerdo en la clave privada a utilizar para que los sistemas se comuniquen entre ellos. Kerberos soporta la transitividad de la confianza; la idea es que si un sistema Kerberos confía en un segundo, y ese segundo confía en un tercer sistema Kerberos, entonces hay automáticamente una relación de confianza entre el primer y el tercer *realm*. La transitividad de la confianza asegura que los sistemas Kerberos sólo necesitan compartir una contraseña con los sistemas Kerberos de

los *realms* inmediatamente encima y debajo de ellos en la jerarquía de *realms* (Kerberos se encarga del resto).

Windows

En Microsoft NT 4.0, el protocolo primario para la seguridad es NTLM (Windows NT LAN Manager). Para Windows 2000, Microsoft eligió un nuevo protocolo de seguridad: Kerberos, por tener algunas ventajas como ser un estándar, más eficiente en el uso de la red y permitir autenticación mutua. Para permitir interoperabilidad con otras implementaciones, Kerberos en Windows soporta DES (Data Encription Standard), pero por defecto usa como algoritmo para encriptación RC4 (fue elegido porque es más rápido y seguro que DES). En Estados Unidos usa claves de 128-bits, mientras que en las versiones internacionales soporta claves de solamente 56-bits.

El protocolo de tickets de Kerberos fue adoptado y extendido por Microsoft, agregando un certificado de privilegios a los tickets (relacionado con el identificador único de NT, así como la lista de los grupos a los que el usuario pertenece).

Windows lo renovará automáticamente mientras el usuario permanezca "logueado". La primera vez que un usuario pide un ticket a Kerberos es cuando se "loguea" en alguna cuenta en el dominio de Windows.

El KDC rechaza cualquier autenticador cuya timestamp sea demasiado vieja (por defecto 5 minutos máximo). Esto implica que los relojes de las máquinas que usen Kerberos deben estar sincronizados, por eso Windows usa el protocolo SNTP (Simple Network Time Protocol) de la IETF para sincronización.

Limitaciones

- Kerberos no tiene efectividad frente a ataques como el de diccionario. Si el usuario escoge una contraseña pobre, un atacante que la consiga tratando de adivinarla puede hacerse pasar por él.
- Kerberos requiere un camino seguro para entrar las contraseñas. Si un usuario entra una contraseña a un programa que ha sido modificado por un atacante, o si la comunicación entre el usuario y el programa de autenticación inicial puede ser monitoreada, entonces el atacante puede llegar a obtener suficiente información para hacerse pasar por el usuario. Kerberos puede combinarse con otras técnicas para eliminar estas limitaciones.
- No hay un lugar seguro donde guardar las claves de sesión. De hecho, el lugar donde se guardan puede ser accedido por el root. Así es que un intruso que logre crackear el mecanismo de protección de la computadora local podrá robar las claves de sesión.
- El protocolo liga los tickets a las direcciones de red y esto es un problema en hosts con más de una dirección IP. En las estaciones de trabajo esto rara vez ocurre.
- Para que Kerberos sea útil debe integrarse con otras partes del sistema. No protege todos los mensajes que se envían entre dos computadoras. Sólo protege los mensajes desde el software que se ha escrito o modificado para usarlo. Aunque puede ser utilizado para intercambiar claves de encriptado cuando se establece un vínculo y niveles de seguridad de servicios de red, esto requeriría cambios en el software de los hosts involucrados.
- Tiempo de vida de un ticket. La elección del tiempo de vida de los tickets no es trivial. Si se elige un tiempo de vida para los tickets muy largo, y un usuario desprevenido olvida desloguearse de una máquina, otra persona puede tomar su

lugar. Por otro lado, si el tiempo de vida de los tickets es muy corto, el usuario va a ser molestado cada cierto tiempo para que ingrese nuevamente su contraseña.

- Manejo de proxies. Todavía no está claro cómo permitir autenticación mediante proxies, es decir que un servidor utilice servicios de otros servidores en nombre de un usuario autenticado. Una solución tentativa es la de permitir que un usuario transfiera momentáneamente sus credenciales hacia otra máquina para poder obtener servicios de ella; pero esto no siempre es deseable pues el usuario podría no confiar en la máquina remota.
- Estados Unidos no permite exportar criptografía, por lo que Kerberos no se puede distribuir en otros países tal como fue creado. Pero hay mucho software legal, que se exporta y que se basa en el uso de Kerberos. Por esta razón, para poder sacarlo del país se creó Bones, que es una versión de Kerberos sin las llamadas a las funciones criptográficas (carece de toda funcionalidad). De todos modos, Errol Young se las ingenió para ponerle a Bones las llamadas criptográficas nuevamente, creando Encrypted Bones, o E-bones.

Conclusiones y futuro de Kerberos

La consecución de la interoperabilidad en la autentificación no es una tarea sencilla y, además, complica la administración. Las compañías europeas han de tener presentes las restrictivas leyes de exportación norteamericanas a la hora de planificar la interoperabilidad de autentificación. Estas leyes prohíben la exportación de software cuyas longitudes de claves de cifrado superen los 56 bits. El Kerberos estándar, la versión del MIT Kerberos V5 -disponible para plataformas UNIX en varias sedes Web de Estados Unidos- utiliza claves de 128 bits y no puede exportarse.

Además, a diferencia de los sistemas de autenticación basados en clave pública, Kerberos no produce firmas digitales. Realmente está diseñado para autenticar peticiones de servicio de un determinado recurso en red, más que para autenticar la autoría de mensajes.

Al estar centralizadas las funciones administrativas, así como la gestión de claves, si la seguridad del servidor se compromete, se comprometerá la seguridad de la totalidad del sistema.

Así pues, se considera que Kerberos es un sistema adecuado dentro de dominios administrativos, ya que entre distintos dominios son más fiables los sistemas de seguridad basados en criptografía de clave pública.

En cuanto al futuro, con la implementación de Kerberos, Microsoft ha enriquecido el proceso de autentificación de su Windows 2000. Kerberos heredó los principios de criptografía simétrica que hacen de él un protocolo de autentificación típico en entornos LAN y orientados a intranets. Sin embargo, la inclusión de PKINIT es el primer paso en el camino hacia la autentificación basada en claves públicas y hacia un mayor uso de Kerberos.

4.4. AUTENTICACIÓN UTILIZANDO KERBEROS

Kerberos es un protocolo de autenticación de red de otra empresa que emplea un sistema de claves secretas compartidas para autenticar de forma segura un usuario en un entorno de red no seguro. El servidor de aplicaciones y el cliente intercambian claves cifradas (tickets), en lugar de un par de ID de usuario y contraseña de texto claro, para establecer las credenciales de un usuario en la red. Un servidor independiente denominado KDc (centro de distribución de claves) emite un ticket tras verificar la validez de un inicio de sesión del usuario.

Cada usuario, o principal en términos de Kerberos, posee una clave de cifrado privada que se comparte con el KDC. Colectivamente, el conjunto de identificadores individuales y sistemas registrados en un KDC se denominan un dominio.

Un ticket de servicio cifrado almacena las credenciales de un usuario. El servidor de bases de datos descifra el ticket para verificar que las credenciales están asociadas a un inicio de sesión de usuario autorizado para el acceso. Mientras exista un ticket de servicio válido en la red, la instancia de IBM® Informix autoriza el acceso del usuario que ha iniciado la sesión al DBMS. El protocolo Kerberos tiene las siguientes características de seguridad:

- Los tickets de servicio existen en la red durante un periodo limitado.
- Sólo el cliente y el servidor puede descifrar estos tickets, por lo que los datos están protegidos si los tickets se interceptan desde la red.
- La entrada del nombre de usuario y contraseña está limitada a la sesión de inicio de sesión inicial, lo que reduce el riesgo de una posible interceptación de credenciales de texto claro.

La administración de los ID de usuario se simplifica porque el KDC aloja un repositorio central para identificadores individuales. Sin embargo, la desventaja de esta descentralización es que crea un solo punto de ataque a los piratas informáticos. Debe sopesar las ventajas de Kerberos con esta posible amenaza para el propio entorno.

4.5. AUTENTICACIÓN UTILIZANDO CRIPTOGRAFÍA

La solución a los problemas de identificación, autenticación y privacidad en sistemas basados en computadoras está en el campo del cifrado. Debido a la naturaleza lógica (no física) del medio, los métodos tradicionales de marcar físicamente el material con un sello o firma (para varios propósitos legales y comerciales) son inútiles. En cambio, alguna marca tiene que ser cifrada dentro de la información misma para poder identificar la fuente, autenticar los contenidos y ofrecer privacidad ante posibles intrusos.

La protección de la privacidad mediante el uso de un algoritmo simétrico, como el indicado en la norma DES (el estándar de cifrado de datos patrocinado por el gobierno), es fácil de implementar en redes pequeñas, pero requiere el intercambio de claves de

cifrado secretas entre cada una de las partes. A medida que las redes aumentan de tamaño, el intercambio seguro de claves secretas se hace cada vez más costoso y difícil de controlar. Por lo tanto, esta solución por sí misma deja de ser práctica para redes medianamente grandes.

El DES tiene otra desventaja: requiere que se comparta una clave secreta. Cada persona debe confiar en que la otra persona guardará la clave secreta de su par y que nunca la revelará a terceros. Ya que el usuario debe tener una clave distinta con cada persona con la que se comunica, deben confiarle una de sus claves secretas a cada una de ellas. Esto significa que en la práctica, las comunicaciones seguras solo pueden ocurrir entre personas con relaciones ya establecidas, ya sean personales o profesionales.

El DES no contempla temas fundamentales, como la autenticación y el no rechazo. Las claves secretas compartidas hacen que sea imposible probar lo que la otra parte pueda haber hecho. Cualquiera de las partes puede modificar los datos clandestinamente sabiendo que un tercero no podrá probar quién fue el culpable. La misma clave que permite comunicarse de manera segura puede ser usada para hacer falsificaciones a nombre del otro usuario.

Una solución mejor: el cifrado de claves públicas

Los problemas de autenticación y protección de la privacidad en redes de gran tamaño fueron contemplados teóricamente en 1976 por Whitfield Diffie y Martin Hellman, cuando publicaron sus conceptos para un método de intercambiar mensajes secretos sin la necesidad de intercambiar claves secretas. La idea se hizo realidad en 1977 con la invención del sistema de cifrado de clave pública RSA, de la mano de Ronald Rivest, Adi Shamir y Len Adleman, entonces profesores en el Instituto Tecnológico de Massachusetts.

En lugar de usar la misma clave para cifrar y descifrar los datos, el sistema RSA usa un par asociado de claves de cifrado y descifrado. Cada clave lleva a cabo una transformación

unidireccional de los datos. Cada clave es la función inversa de la otra; lo que una hace, solo la otra lo puede deshacer.

El dueño de la clave pública RSA la comparte públicamente, mientras que mantiene secreta su clave privada RSA. Para enviar un mensaje privado, un autor cifra el mensaje con la clave pública del destinatario. Una vez cifrado, el mensaje solo podrá ser descifrado con la clave privada del destinatario.

A la inversa, el usuario también puede cifrar datos usando su clave privada. Es decir que las claves RSA funcionan en ambos sentidos. Esto es la base de la "firma digital", ya que, si el usuario puede descifrar un mensaje con la clave pública de alguien, ese otro usuario tiene que haber usado su clave privada para cifrar el mensaje en primer lugar. Ya que solo el dueño puede utilizar su propia clave privada, el mensaje cifrado se convierte en una suerte de firma electrónica, un documento que nadie más puede crear.

Autenticación y no rechazo: el certificado digital de VeriSign

Una firma digital se crea aplicándole un algoritmo de hash a un mensaje de texto. Esto genera una síntesis del mensaje. Esta síntesis se cifra con la clave privada del individuo que envía el mensaje, convirtiéndolo en una firma digital. La firma digital solo puede ser descifrada con la clave pública del mismo individuo. El destinatario del mensaje descifra la firma digital y luego recalcula la síntesis del mensaje (digest). El valor de esta síntesis del mensaje recién calculada se compara con el valor de la síntesis del mensaje encontrado en la firma. Si coinciden, el mensaje no fue alterado. Ya que la clave pública del remitente se usó para verificar la firma, el texto tiene que haber sido firmado con la clave privada conocida únicamente por el remitente. Todo este proceso de autenticación será incorporado en aquellas aplicaciones que tengan en cuenta la seguridad.

¿Qué es un certificado digital?

Los usuarios de la tecnología RSA suelen adjuntar sus claves públicas únicas a los documentos salientes a fin de que los destinatarios no deban buscar dicha clave pública en el repositorio de claves públicas. Pero, ¿cómo puede el destinatario estar seguro de que dicha clave pública, o aún la que está publicada en el repositorio, pertenece realmente a esa persona? ¿No podría un intruso entrar a la red informática como un usuario legítimo y literalmente sentarse y observar como otros envían documentos confidenciales y secretos a una cuenta falsa creada por el intruso?

La solución es el certificado digital, una suerte de "pasaporte" o "credencial" digital. El certificado digital es la clave pública del usuario, "firmada digitalmente" por alguien confiable, como un director de seguridad de la red, el servicio de asistencia de MIS o VeriSign Inc. La siguiente figura es una representación gráfica de un certificado digital.

Cada vez que alguien envía un mensaje, adjunta su certificado digital. El destinatario del mensaje primero usa su certificado digital para verificar que la clave pública del autor sea auténtica y luego usa esa clave pública para verificar el mensaje mismo. De esta manera, solo una clave pública, la de la autoridad de certificación, tiene que ser almacenada centralmente o ser muy publicitada, ya que todos podrán simplemente transmitir sus claves públicas y certificados digitales válidos junto a sus mensajes.

Usando certificados digitales, se puede establecer una cadena de autenticación que corresponda con una jerarquía organizacional, permitiendo un conveniente registro y certificación de claves públicas en un entorno distribuido.

Jerarquías de certificación

Una vez que un usuario tiene un certificado digital, ¿qué puede hacer con él? Los certificados digitales tienen una amplia gama de usos, desde correos electrónicos empresariales internos a transferencias electrónicas de fondos (EFT, por sus siglas en inglés). Para poder usar certificados digitales debe existir un gran nivel de confianza en la asignación del certificado digital y el usuario u organización vinculado a dicho certificado. Esta confianza se logra estableciendo jerarquías de certificación digital, y logrando que todos los miembros de dicha jerarquía cumplan con las mismas políticas. Los certificados digitales solo pueden ser emitidos por personas o entidades, como miembros potenciales de la jerarquía, una vez que sus identidades hayan sido autenticadas. Distintas jerarquías pueden tener distintas políticas sobre cómo autenticar la identidad y la manera en la que se emiten los certificados digitales.

Verisign opera varias jerarquías de certificación digital. La AC comercial ofrece un alto nivel de seguridad en cuanto a la relación entre el certificado digital del usuario final y el usuario final real. Los miembros de la AC comercial del RSA ofrecerán un alto nivel de seguridad, logrado mediante el cumplimiento de las políticas, en cuanto a la persona o entidad con la que se están comunicando. Este no suele ser el caso cuando dos usuarios finales, miembros de jerarquías con menor nivel de seguridad, se comunican con certificados digitales. Sin la seguridad que brinda una jerarquía de certificación administrada apropiadamente, el uso de certificados digitales tiene un valor limitado.

4.6. MÉTODOS DE AUTENTICACIÓN

Autenticación es el proceso que debe seguir un usuario para tener acceso a los recursos de un sistema o de una red de computadores. Este proceso implica identificación (decirle al sistema quién es) y autenticación (demostrar que el usuario es quien dice ser)

4.7. TLS, PEAP

SSL y TLS son protocolos que proporcionan comunicación segura en una red, como Internet.

- SSL: Protocolo de capa de conexion segura
- TLS: Seguridad de la capa de trasnporte

Una de las principales aplicaciones praticas de estos protocolos es formar https junto a http garantizando el envio y recepcion de informacion de forma segura en un navegador web.

Toda comunicacion mediante SSL o TLS consta de dos fases:

- Fase de saludo: Se negocia entre las partes el algoritmo que usara en la comuncacion.
- Fase de comunicación: En la que el cifrado de trafico basado en cifrado simetrico a partir de la clave de sesión y se van generando nuevas claves de forma dinamica

SSL presenta las versiones 1 y 2 (en las que se proporciona autenticación de servidor) y 3 (en que se añade autenticación del cliente por medio de certificados digitales del cliente y servidor)

4.8. CONFIGURACIONES

En esta sección se enumeran los valores que se pueden configurar para EAP protegido.

Importante

Si se implementa el mismo tipo de autenticación para PEAP y EAP, se crea una vulnerabilidad de seguridad. Al implementar PEAP y EAP (sin proteger), no use el mismo tipo de autenticación. Por ejemplo, si implementa PEAP-TLS, no implemente también EAP-TLS.

4.9. SERVIDORES

Este elemento especifica que el cliente comprueba que los certificados de servidor presentados al equipo cliente tienen las firmas correctas, no han expirado y las emitió una entidad de certificación raíz (CA) de confianza. La configuración predeterminada es "enabled". Si deshabilita esta casilla, los equipos cliente no podrán comprobar la identidad de los servidores durante el proceso de autenticación. Si no se produce la autenticación del servidor, los usuarios se exponen a riesgos de seguridad graves, incluida la posibilidad de que los usuarios puedan conectarse sin saberlo a una red no autorizado.

Conectar a estos servidores

Este elemento permite especificar el nombre de los servidores Servicio de autenticación remota telefónica de usuario (RADIUS) que proporcionan autenticación y autorización de red. Tenga en cuenta que debe escribir el nombre exactamente como aparece en el campo Asunto de cada certificado de servidor RADIUS o usar expresiones regulares para especificar el nombre del servidor. La sintaxis completa de la expresión regular se puede usar

para especificar el nombre del servidor, pero para diferenciar una expresión regular con la cadena literal, debe usar al menos un "" en la cadena especificada. Por ejemplo, puede especificar nps.example.com para especificar el servidor RADIUS nps1.example.com o nps2.example.com.

4-10. RAÍZ DE CONFIANZA

En este elemento se enumeran las entidades de certificación raíz de confianza. La lista se basa en las CA raíz de confianza que están instaladas en el equipo y en los almacenes de certificados de usuario. Puede especificar los certificados de CA raíz de confianza que usan los suplicantes para determinar si confían en los servidores como, por ejemplo, el servidor que ejecuta el Servidor de directivas de redes (NPS) o el servidor de aprovisionamiento. Si no se selecciona ninguna CA raíz de confianza, el cliente 802.1X comprueba que el certificado de equipo del servidor RADIUS haya sido emitido por una CA raíz de confianza instalada. Si se seleccionan una o varias CA raíz de confianza, el cliente 802.1X comprueba que el certificado de equipo del servidor RADIUS haya sido emitido por una CA raíz de confianza seleccionada. Incluso si no se selecciona ninguna entidad de certificación raíz de confianza, el cliente comprueba que el emisor del certificado de servidor RADIUS sea una CA raíz de confianza.

4.11. RECONEXIÓN RÁPIDA

Permite crear una asociación de seguridad nueva o actualizada de forma más eficaz o en un número menor de recorridos de ida y vuelta, en el caso de que se estableciera previamente una asociación de seguridad.

Para las conexiones VPN, la reconexión rápida usa la tecnología IKEv2 para ofrecer una conectividad perfecta y coherente de la VPN cuando los usuarios pierden temporalmente sus conexiones a Internet. Los usuarios que se conecten mediante banda ancha móvil inalámbrica serán los que más se beneficien de esta funcionalidad.

Un ejemplo de esta ventaja es un escenario común en el que un usuario viaja en tren, usa una tarjeta de banda ancha móvil inalámbrica para conectarse a Internet y, a continuación, establece una conexión VPN con la red corporativa.

Cuando el tren pasa por un túnel, se pierde la conexión a Internet. Una vez que el tren está fuera del túnel, la tarjeta de banda ancha móvil inalámbrica vuelve a conectarse automáticamente a Internet.

Bibliografía básica y complementaria:

- http://csrc.nist.gov/encryption/tkencryption.html - Especificaciones de los algoritmos DES, TDES y AES (Sitio Web del NIST);
- Block cipher modes of operation
- Secure Programming Cookbook, Ed O'Reilly. Viega, Messier.
- "Handbook of Applied Cryptography". A. Menezes, P. van Oorschot, and S. Vanstone. CRC Press, Inc. 1997.
- "Cryptography and Network Security. Principles and practices". William Stallings. Prentice Hall. Pearson Education. Third edition. 2003.
- "Introduction of Cryptography with coding theory". Wade Trappe and Lawrence C. Washington. Prentice Hall, 2002.
- "Introduction to Cryptography". Johannes A. Buchmann. Springer Verlag, 2004. Second Edition.
- www.elcodigok.com.ar
- www.ibiblio.org
- www.segu-info.com.ar
- www.cryptography.com
- http://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/ejmrsa.html
  http://es.wikipedia.org/wiki/RSA
  http://daniellerch.com/sources/doc/algoritmo_rsa.html
- "Fonaments de matemática discreta", J.M. Basart, J. Rif´a, M. Villanueva, primera edición, Universitat Autónoma de Barcelona, Servei de Publicacions, ISBN 84-490-08555-7.
- "Criptograf´ıa y Seguridad en Computadores", tercera edici´on, versi´on 1.14, Manuel Lucena, Universidad de Ja´en. Libro electr´onico gratuito disponible en http://www.di.ujaen.es/~mlucena/lcripto.
- Mark Stamp. Information Security: Principles and Practice. Capítulo 4.
- El magnífico personal del departamento de automática de la UAH.

- FNMT, Fábrica Nacional de Moneda y Timbre.
- Public-key cryptography, Wikipedia
- Jennifer G. Steiner, Clifford Neuman, Jeffrey I. Schiller. Kerberos: An Authentication Service for Open Network Systems ftp://athena-dist.mit.edu/pub/Kerberos/doc/usenix.txt
- Roger M. Needham, Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers Communications of the ACM 21(12) pp. 993-999 (diciembre)
- John Kohl, Clifford Neuman. RFC 1510 http://www.ietf.org/rfc/rfc1510.txt
- FAQ about Kerberos http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html
- Mark Walla Kerberos explained http://www.microsoft.com/TechNet/win2000/kerberos.asp?a=printable
- Exploring Kerberos, the protocol for distributed security in Windows 2000 http://www.microsoft.com/msj/0899/kerberos/kerberos.htm
- Kerberos y Windows 2000. ¿Es tan fiero el nuevo protocolo de auteticación como lo pintan? http://www.w2000mag.com/atrasados/1999/37_dic99/articulos/kerberos.htm